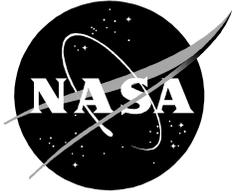


NASA/SP—2007–6105 Rev1



# Systems Engineering Handbook

*Stephen J. Kapurch*  
*Office of the Chief Engineer*  
*NASA Headquarters, Washington, D.C.*

*Neil E. Rainwater*  
*Marshall Space Flight Center, Huntsville, Alabama*

**Restricted: This document is a draft. Please do not copy or distribute.**

National Aeronautics and  
Space Administration

*NASA Headquarters*  
*Washington, DC 20546*

**June 2007**

---

## NASA STI Program ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Report Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.

- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include creating custom thesauri, building customized databases, and organizing and publishing research results.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at <http://www.sti.nasa.gov>
- E-mail your question via the Internet to [help@sti.nasa.gov](mailto:help@sti.nasa.gov)
- Fax your question to the NASA STI Help Desk at (301) 621-0134
- Phone the NASA STI Help Desk at (301) 621-0390
- Write to:

NASA STI Help Desk  
NASA Center for AeroSpace Information  
7115 Standard Drive  
Hanover, MD 21076-1320

## Table of Contents

Foreword.....	18
Preface.....	19
Acknowledgments.....	20
1.0 Introduction.....	22
1.1 Purpose.....	22
1.2 Scope and Depth .....	22
2.0 Fundamentals of Systems Engineering.....	23
2.1 The Common Technical Processes and the Systems Engineering Engine .....	25
2.2 Comparison of the SE Engine with the “Vee”.....	26
2.3 Example of Using the SE Engine.....	27
2.3.1 Detailed Example.....	28
2.3.2 Example Premise .....	29
2.3.2.1 Phase A System Design Passes.....	29
2.3.2.2 Product Realization Passes .....	33
2.3.2.3 Example in Phases B through D .....	36
2.3.2.4 Use of the SE Engine in Phase E and F .....	37
2.4 Distinctions Between Product Verification and Validation.....	38
2.5 Cost Aspect of Systems Engineering.....	39
3.0 NASA Program/Project Life Cycle .....	43
3.1 Program Formulation.....	47
3.2 Program Implementation .....	47
3.3 Project Pre-Phase A—Concept Studies .....	48
3.4 Project Phase A—Concept and Technology Development .....	49
3.5 Project Phase B—Preliminary Design and Technology Completion .....	51
3.6 Project Phase C—Final Design and Fabrication.....	52
3.7 Project Phase D—System Assembly, Integration and Test, Launch.....	54
3.8 Project Phase E—Operations and Sustainment .....	55
3.9 Project Phase F—Closeout .....	56
3.10 Funding: The Budget Cycle.....	57
4.0 System Design .....	60
4.1 Stakeholder Expectations Definition .....	62
4.1.1 Process Description.....	62
4.1.1.1 Inputs.....	62
4.1.1.2 Process Activities.....	63
Identifying Stakeholders.....	63
Identifying Stakeholder Expectations.....	63
4.1.1.3 Outputs.....	65
4.1.2 Stakeholder Expectations Definition Guidance .....	65
4.1.2.1 Concept of Operations .....	65
4.2 Technical Requirements Definition .....	71
4.2.1 Process Description.....	71
4.2.1.1 Inputs.....	72
4.2.1.2 Process Activities.....	72

	4.2.1.3	Outputs .....	74
4.2.2		Technical Requirements Definition Guidance.....	74
	4.2.2.1	Types of Requirements .....	74
		Functional Requirements.....	75
		Performance Requirements .....	76
		Interface Requirements.....	77
		Environmental Requirements .....	77
		Reliability Requirements .....	78
		Safety Requirements .....	78
	4.2.2.2	Requirements Decomposition, Allocation, and Validation ....	79
	4.2.2.3	Capturing Requirements and the Requirements Database.....	81
	4.2.2.4	Technical Standards .....	82
		Importance of Standards Application .....	82
4.3		Logical Decomposition.....	84
	4.3.1	Process Description.....	84
		4.3.1.1 Inputs.....	85
		4.3.1.2 Process Activities.....	85
		4.3.1.3 Outputs.....	88
	4.3.2	Logical Decomposition Guidance.....	88
		4.3.2.1 Product Breakdown Structure .....	88
		4.3.2.2 Functional Analysis Techniques .....	89
		Functional Flow Block Diagrams.....	89
		N2 Diagrams .....	92
		Time-Line Analysis.....	93
4.4		Design Solution Definition .....	94
	4.4.1	Process Description.....	94
		4.4.1.1 Inputs.....	95
		4.4.1.2 Process Activities .....	95
		Define Alternative Design Solutions .....	95
		Create Alternative Design Concepts .....	96
		Analyze Each Alternative Design Solution .....	97
		Selecting the “Best” Design Solution Alternative .....	98
		Increase the Resolution of the Design .....	100
		Fully Describe the Design Solution.....	100
		Verify the Design Solution.....	101
		Validate the Design Solution.....	102
		Identify Enabling Products.....	102
		Baseline the Design Solution .....	103
		4.4.1.3 Outputs.....	103
	4.4.2	Design Solution Definition Guidance.....	104
		4.4.2.1 Technology Assessment.....	104
		4.4.2.2 Integrating Engineering Specialties into the Systems	
		Engineering Process.....	105
		Safety and Reliability .....	105
		Quality Assurance .....	108
		Integrated Logistics Support .....	109
		Maintainability .....	110
		Producibility .....	111
		Human Factors Engineering.....	112
5.0		Product Realization.....	115

5.1	Product Implementation.....	117
5.1.1	Process Description.....	117
5.1.1.1	Inputs.....	118
5.1.1.2	Process Activities.....	118
	Prepare to Conduct Implementation.....	118
	Purchase, Make, or Reuse the Product.....	119
	Capture Work Products.....	120
5.1.1.3	Outputs.....	121
5.1.2	Product Implementation Guidance.....	121
5.1.2.1	Buying Off-the-Shelf Products.....	121
5.2	Product Integration.....	123
5.2.1	Process Description.....	123
5.2.1.1	Inputs.....	124
5.2.1.2	Process Activities.....	125
5.2.1.3	Outputs.....	126
5.2.2	Product Integration Guidance.....	126
5.2.2.1	Relationship to Product Implementation.....	126
5.2.2.2	Product/Interface Integration Support.....	127
5.2.2.3	Product Integration of Design Solution.....	127
5.2.2.4	Interface Management.....	127
5.2.2.5	Utilization and Value.....	128
5.2.2.6	Interface Management Tasks.....	128
	Define Interfaces.....	128
	Verify Interfaces.....	129
	Integration Strategy.....	129
	System and Subsystem Receipt.....	130
	System and Subsystem Element Validation.....	130
	Verify Element Interfaces.....	130
	Integration and Verification.....	130
5.3	Product Verification.....	131
	Distinctions Between Verification and Validation.....	131
5.3.1	Process Description.....	132
5.3.1.1	Inputs.....	133
5.3.1.2	Process Activities.....	133
	Product Verification Planning.....	133
	Verification Plan and Methods.....	133
	Product Verification Preparation.....	136
	Conduct Planned Product Verification.....	137
	Analyze Product Verification Results.....	137
	Reengineering.....	138
	Verification Deficiencies.....	138
	Pass Verification But Fail Validation?.....	139
	Capture Product Verification Work Products.....	139
5.3.1.3	Outputs.....	139
5.3.2	Product Verification Guidance.....	140
5.3.2.1	Verification Program.....	140
5.3.2.2	Verification in the Life Cycle.....	141
	Quality Assurance in Verification.....	141
	Configuration Verification.....	142
	Qualification Verification.....	142

	Acceptance Verification .....	143
	Deployment Verification .....	143
	Operational and Disposal Verification .....	143
5.3.2.3	Verification Procedures .....	144
5.3.2.4	Verification Reports .....	145
5.3.2.5	End-to-End System Testing .....	146
	How to Perform End-to-End Testing .....	147
5.3.2.6	Modeling and Simulation .....	150
5.3.2.7	Hardware-in-the-Loop .....	150
5.4	Product Validation .....	152
	Distinctions Between Verification and Validation .....	152
5.4.1	Process Description .....	153
5.4.1.1	Inputs .....	153
5.4.1.2	Process Activities .....	154
	Product Validation Planning .....	155
	Validation Plan and Methods .....	155
	Product Validation Preparation .....	157
	Conduct Planned Product Validation .....	158
	Analyze Product Validation Results .....	159
	Validation Notes .....	159
	Reengineering .....	160
	Validation Deficiencies .....	160
	Capture Product Validation Work Products .....	160
5.4.1.3	Outputs .....	161
5.4.2	Product Validation Guidance .....	161
5.4.2.1	Modeling and Simulation .....	161
5.4.2.2	Software .....	162
5.5	Product Transition .....	163
5.5.1	Process Description .....	163
5.5.1.1	Inputs .....	164
5.5.1.2	Process Activities .....	166
	Prepare to Implement Transition .....	166
	Prepare the Product for Transition .....	167
	Prepare the Site to Receive Product .....	167
	Transition the Product .....	168
	Confirm Ready to Support .....	168
5.5.1.3	Outputs .....	168
5.5.2	Product Transition Guidance .....	169
5.5.2.1	Additional Product Transition Input Considerations .....	169
5.5.2.2	After Product Transition to the End User What Next? .....	170
6.0	Crosscutting Technical Management .....	171
6.1	Technical Planning .....	173
6.1.1	Process Description .....	173
6.1.1.1	Inputs .....	173
6.1.1.2	Process Activities .....	174
	Technical Planning Preparation .....	177
	Define the Technical Work .....	178
	Schedule, Organize, and Cost the Technical Effort .....	178
	Prepare the SEMP and Other Technical Plans .....	185
	Obtain Stakeholder Commitments to Technical Plans .....	187

		Issue Technical Work Directives.....	188
		Capture Technical Planning Work Products .....	190
	6.1.1.3	Outputs.....	190
6.1.2		Technical Planning Guidance .....	191
	6.1.2.1	Work Breakdown Structure .....	191
		Role of the WBS .....	194
		Techniques for Developing the WBS.....	194
		Common Errors in Developing a WBS.....	195
	6.1.2.2	Cost Definition and Modeling .....	198
		Life-Cycle Cost and Other Cost Measures .....	199
		What Costs Should Be Counted .....	199
		Costs Occurring Over Time .....	199
		Difficult-to-Measure Costs.....	199
		Controlling Life-Cycle Costs .....	200
		Cost Estimating Methods .....	201
		Integrating Cost Model Results for a Complete Life-Cycle Cost Estimate .....	201
	6.1.2.3	Lessons Learned.....	202
		Systems Engineering Role in Lessons Learned.....	203
		Utilization of LL Best Practice.....	203
		Management of LL Best Practice .....	204
6.2		Requirements Management .....	205
	6.2.1	Process Description.....	205
		6.2.1.1 Inputs.....	205
		6.2.1.2 Process Activities.....	206
		Requirements Traceability .....	206
		Requirements Validation.....	207
		Managing Requirement Changes .....	208
		Key Issues for Requirements Management.....	208
		6.2.1.3 Outputs.....	210
	6.2.2	Requirements Management Guidance .....	210
		6.2.2.1 Requirements Management Plan .....	210
		6.2.2.2 Requirement Management Tools.....	211
6.3		Interface Management .....	212
	6.3.1	Process Description.....	212
		6.3.1.1 Inputs.....	212
		6.3.1.2 Process Activities.....	213
		6.3.1.3 Outputs.....	213
	6.3.2	Interface Management Guidance .....	214
		6.3.2.1 Interface Requirements Document .....	214
		6.3.2.2 Interface Control Document or Interface Control Drawing..	215
		6.3.2.3 Interface Definition Document .....	215
		6.3.2.4 Interface Control Plan .....	215
6.4		Technical Risk Management.....	216
	6.4.1	Process Description.....	216
		6.4.1.1 Inputs.....	217
		6.4.1.2 Process Activities.....	217
		6.4.1.3 Outputs.....	218
	6.4.2	Technical Risk Management Guidance .....	218

6.4.2.1	Role of Continuous Risk Management in Technical Risk Management.....	220
6.4.2.2	The Interface Between CRM and Risk-Informed Decision Analysis.....	222
	Risk Analysis, Performing Trade Studies and Ranking (CRM Steps: “Identify,” “Analyze”).....	223
6.4.2.3	Selection and Application of Appropriate Risk Methods .....	224
	Types of Risk .....	224
	Qualitative Methods .....	225
	Quantitative Methods .....	227
	Deliberation and Recommendation of Decision Alternative (CRM Steps: “Plan,” “Deliberate, Communicate, and Document”) .....	228
	Tracking and Controlling Performance Deviations (CRM Steps: “Track,” “Control”).....	231
6.5	Configuration Management .....	234
6.5.1	Process Description.....	234
6.5.1.1	Inputs.....	235
6.5.1.2	Process Activities.....	235
	CM Planning .....	236
	Configuration Identification .....	236
	Configuration Control .....	238
	Configuration Status Accounting .....	240
	Configuration Verification and Audits.....	241
6.5.1.3	Outputs.....	241
6.5.2	Configuration Management Guidance.....	241
6.5.2.1	What Is the Impact of Not Doing CM?.....	241
6.5.2.2	When Is It Acceptable to Use Redline Drawings?.....	243
6.6	Technical Data Management .....	245
6.6.1	Process Description.....	245
6.6.1.1	Inputs.....	245
6.6.1.2	Process Activities.....	246
	Role of Data Management Plan.....	246
	Technical Data Management Key Considerations .....	246
	General Data Management Roles .....	247
	Data Identification/Definition .....	248
	Initial Data Management System Structure.....	249
	Data Management Planning .....	249
	Key Considerations: for Planning of Data Management and for a Tool Selection .....	250
	Value of Data .....	252
	Technical Data Capture Tasks.....	252
	Protection for Data Deliverables .....	254
6.6.1.3	Outputs.....	254
6.6.2	Technical Data Management Guidance .....	255
6.7	Technical Assessment.....	257
6.7.1	Process Description.....	257
6.7.1.1	Inputs.....	257
6.7.1.2	Process Activities.....	258
6.7.1.3	Outputs.....	260
6.7.2	Technical Assessment Guidance.....	260
6.7.2.1	Reviews, Audits, and Key Decision Points .....	260

		Purpose and Definition.....	260	
		Project Termination.....	261	
		General Principles for Reviews.....	262	
		Internal Review Boards.....	265	
		Required Technical Reviews.....	265	
		Results of Review.....	269	
		Other Technical Reviews.....	285	
	6.7.2.2	Status Reporting and Assessment.....	288	
		Cost and Schedule Control Measures.....	288	
		Technical Measures—MOEs, MOPs, and TPMs.....	291	
		Systems Engineering Process Metrics.....	297	
6.8	Decision Analysis.....		300	
	6.8.1	Process Description.....	300	
		6.8.1.1	Inputs..... 302	
		6.8.1.2	Process Activities..... 303	
			Establish Guidelines to Determine Which Technical Issues Are Subject to a Formal Analysis/Evaluation Process.....303	
			Define the Criteria for Evaluating Alternative Solutions.....303	
			Identify Alternative Solutions to Address Decision Issues.....304	
			Select Evaluation Methods and Tools.....305	
			Evaluate Alternative Solutions with the Established Criteria and Selected Methods.....305	
			Select Recommended Solutions from the Alternatives Based on the Evaluation Criteria.....306	
			Report the Analysis and Evaluation Results and Findings with Recommendations, Impacts, and Corrective Actions.....307	
			Capture Work Products from Decision Analysis Activities.....307	
		6.8.1.3	Outputs..... 308	
	6.8.2	Decision Analysis Guidance.....	308	
		6.8.2.1	Systems Analysis, Simulation, and Performance..... 308	
			Functional Analysis Techniques.....311	
		6.8.2.2	Trade Studies..... 311	
			Controlling the Trade Study Process.....316	
			Trade Study Reports.....316	
		6.8.2.3	Cost-Benefit Analysis..... 316	
		6.8.2.4	Influence Diagrams..... 318	
		6.8.2.5	Decision Trees..... 319	
		6.8.2.6	Multi-Criteria Decision Analysis..... 320	
			The Analytic Hierarchy Process.....320	
			Flexibility and Extensibility Attributes.....321	
		6.8.2.7	Utility Analysis..... 322	
		6.8.2.8	Risk-Informed Decision Analysis Process Example..... 324	
			Introduction.....324	
			Objectives Hierarchy/TPMs.....325	
7.0	Special Topics.....		327	
	7.1	Engineering with Contracts.....	328	
		7.1.1	Introduction, Purpose, and Scope..... 328	
		7.1.2	Acquisition Strategy..... 328	
			7.1.2.1	Develop an Acquisition Strategy..... 329
			7.1.2.2	Acquisition Life Cycle..... 329
			7.1.2.3	NASA Responsibility for Systems Engineering..... 330

7.1.3	Prior to Contract Award.....	333
7.1.3.1	Acquisition Planning.....	333
7.1.3.2	Develop the Statement of Work.....	334
7.1.3.3	Task Order Contracts .....	337
7.1.3.4	Surveillance Plan .....	338
7.1.3.5	Writing Proposal Instructions and Evaluation Criteria.....	338
7.1.3.6	Selection of COTS Products .....	340
7.1.3.7	Acquisition-Unique Risks .....	340
7.1.4	During Contract Performance .....	342
7.1.4.1	Performing Technical Surveillance.....	342
7.1.4.2	Evaluating Work Products .....	343
7.1.4.3	Issues with Contract-Subcontract Arrangements.....	344
7.1.5	Contract Completion.....	346
7.1.5.1	Acceptance of Final Deliverables .....	346
7.1.5.2	Transition Management .....	347
7.1.5.3	Transition to Operations and Support.....	347
7.1.5.4	Decommissioning and Disposal.....	348
7.1.5.5	Final Evaluation of Contractor Performance .....	349
7.2	Integrated Design Facilities .....	350
7.2.1	Introduction.....	350
7.2.2	CACE Overview and Importance .....	351
7.2.3	CACE Purpose and Benefits.....	352
7.2.4	CACE Staffing.....	353
7.2.5	CACE Process.....	353
7.2.5.1	Planning and Preparation .....	353
7.2.5.2	Activity Execution Phase.....	354
7.2.5.3	Activity Wrap-Up .....	355
7.2.6	CACE Engineering Tools and Techniques .....	356
7.2.7	CACE Facility, Information Infrastructure, and Staffing .....	356
7.2.7.1	Facility .....	356
7.2.7.2	Information Infrastructure.....	357
7.2.7.3	Facility Support Staff Responsibilities .....	357
7.2.8	CACE Products.....	358
7.2.9	CACE Best Practices .....	358
7.2.9.1	People.....	358
7.2.9.2	Process and Tools .....	359
7.2.9.3	Facility .....	360
7.3	Human Factors Engineering .....	361
7.3.1	Basic Human Factors Model.....	362
7.3.2	Human Factors Analysis and Evaluation Techniques .....	363
7.4	Environmental, Nuclear Safety, Planetary Protection, and Asset Protection Policy Compliance .....	369
7.4.1	National Environmental Policy Act and Executive Order 12114 Environmental Effects Abroad of Major Federal Actions.....	369
7.4.1.1	National Environmental Policy Act .....	369

7.4.1.2	Executive Order 12114 Environmental Effects Abroad of Major Federal Actions .....	370
7.4.2	Presidential Directive/National Security Council Memorandum #25 ....	370
7.4.3	Planetary Protection .....	372
7.4.4	Space Asset Protection.....	374
7.4.4.1	Protection Policy .....	374
7.4.4.2	Goal.....	375
7.4.4.3	Scoping .....	375
7.4.4.4	Protection Planning .....	375
7.5	Use of Metric System.....	376
7.6	Selecting Engineering Design Tools.....	379
7.6.1	Program and Project Considerations.....	379
7.6.2	Policy and Processes .....	379
7.6.3	Collaboration.....	380
7.6.4	Design Standards .....	380
7.6.5	Existing IT Architecture .....	380
7.6.6	Tool Interfaces .....	381
7.6.7	Interoperability and Data Formats .....	381
7.6.8	Backward Compatibility .....	381
7.6.9	Platform.....	382
7.6.10	Tool Configuration Control .....	382
7.6.11	Security/Access Control.....	382
7.6.12	Training.....	382
7.6.13	Licenses.....	382
7.6.14	Stability of Vendor and Customer Support.....	383
Appendix A:	Acronyms .....	384
Appendix B:	Glossary.....	385
Appendix 4-2a:	How to Write a Good Requirement .....	386
Appendix 4-2b:	Requirements Verification Matrix .....	391
Appendix 4-2c:	Creating the Validation Plan (Including Validation Requirements Matrix)...	393
Appendix 4-3:	Functional, Timing, and State Analysis.....	394
Appendix 4-4:	Technology Assessment/Insertion.....	403
Appendix 5-2:	Integration Plan Outline.....	412
Appendix 5-4:	V&V Plan Sample Outline .....	415
Appendix 6-1:	SEMP Content Outline .....	417
Appendix 6-3:	Interface Requirements Document Outline .....	426
Appendix 6-5:	CM Plan Outline .....	430
Appendix 6-7:	Guidance on Technical Peer Reviews/Inspections .....	431
Appendix 7-4:	Project Protection Plan Outline .....	437
Bibliography/Suggested Reading.....		439
Index .....		446

## Boxes

The Systems Engineer’s Dilemma.....	42
Program Formulation .....	47
Program Implementation .....	48
Pre-Phase A—Concept Studies.....	49
Phase A—Concept and Technology Development.....	50
Phase B—Preliminary Design and Technology Completion.....	52
Phase C—Final Design and Fabrication .....	54
Phase D—System Assembly, Integration and Test, Launch .....	55
Phase E—Operations and Sustainment.....	56
Phase F —Closeout.....	57
System Design Keys .....	61
Example of Functional and Performance Requirements .....	76
Rationale .....	82
DoD Architecture Framework .....	87
Prototypes .....	112
Product Realization Keys.....	116
Differences between Verification and Validation Testing: .....	132
Types of Testing .....	134
Types of Verification .....	135
Differences between Verification and Validation Testing: .....	153
Types of Validation.....	156
VV&A.....	162
Crosscutting Technical Management Keys .....	172
Desirable Features in Gantt Charts .....	183
Definitions.....	207
Typical Interface Management Check List.....	215
Key Concepts in Technical Risk Management.....	220
Example Sources of Risk .....	225
Limitations of Risk Matrices .....	227
Types of Configuration Control Changes .....	240
Warning Signs/Red Flags (How Do You Know When You’re in Trouble?).....	243
Redlines Were identified as One of the Major Causes of the NOAA-N Prime Mishap September 6, 2003 .....	244
Inappropriate Uses of Technical Data.....	249
Data Collection Checklist .....	254
Data Security and ITAR.....	255
Termination Review.....	262

Analyzing the Estimate at Completion .....	291
Examples of Technical Performance Measures .....	294
Solicitations.....	334
Source Evaluation Board .....	339
Context Diagrams .....	402

## Figures

Figure 2-1	Context of SE within Project Scope.....	24
Figure 2-2	The Systems Engineering Engine .....	25
Figure 2-3	SE Engine Tracking Icon .....	29
Figure 2-4	Product Hierarchy Through One Pass of the SE Engine .....	30
Figure 2-5	Tier 2 Product Hierarchy for the External Tank .....	31
Figure 2-6	Example Orbiter Tier 2 Product Hierarchy.....	32
Figure 2-7	Tier 3 Product Hierarchy Example for Avionics System .....	32
Figure 2-8	Example Phase A Product Hierarchy after Completely Passing Through the System Design Process Side of the SE Engine .....	33
Figure 2-9	Model of Typical Activities During Operational Phase (Phase E) of a Product ..	37
Figure 2-10	New Products or Upgrades Reentering the SE Engine.....	38
Figure 2-11	The Enveloping Surface of Non-Dominated Designs .....	39
Figure 2-12	Estimates of Outcomes to be Obtained from Several Design Concepts Including Uncertainty.....	41
Figure 3-1	NASA Program Life Cycle.....	45
Figure 3-2	NASA Project Life Cycle .....	46
Figure 3-3	Typical NASA Budget Cycle.....	58
Figure 4-1	Interrelationships between the System Design Processes.....	60
Figure 4-2	Stakeholder Expectations Definition Process .....	62
Figure 4-3	Product Flow for Stakeholder Expectations .....	64
Figure 4-4	Typical ConOps Development for a Science Mission.....	66
Figure 4-5	Example of an Associated End-to-End Operational Architecture .....	67
Figure 4-6a.	Example of a Lunar Sortie Timeline Developed Early in the Life Cycle.....	68
Figure 4-6b.	Example of a Lunar Sortie Design Reference Mission.....	68
Figure 4-7	Example of a More Detailed Integrated Timeline Later in the Life Cycle for a Science Mission .....	69
Figure 4-8	Technical Requirements Definition Process.....	72
Figure 4-9	Characteristics of Functional, Operational, Reliability, Safety, and Specialty Requirements .....	75
Figure 4-10	The Flow Down of Requirements.....	80
Figure 4-11	Allocation and Flow-Down of Science Pointing Requirements .....	81

Figure 4-12	Logical Decomposition Process.....	84
Figure 4-13	An Example of a PBS .....	89
Figure 4-14	Example of a Functional Flow Block Diagram .....	91
Figure 4-15	Example of an N-squared diagram for orbital equipment .....	92
Figure 4-16	Design Solution Definition Process .....	94
Figure 4-17	The Doctrine of Successive Refinement.....	96
Figure 4-18	A Quantitative Objective Function, Dependent on Life-Cycle Cost and All Aspects of Effectiveness .....	99
Figure 5-1	Product Realization.....	115
Figure 5-2	Product Implementation Process.....	117
Figure 5-3	Product Integration Process .....	124
Figure 5-4	Product Verification Process.....	132
Figure 5-5	Bottoms-Up Realization Process .....	141
Figure 5-6	Example of End-to-End Data Flow for a Scientific Satellite Mission.....	147
Figure 5-7	Product Validation Process .....	153
Figure 5-8	Product Transition Process .....	164
Figure 6-1	Technical Planning Process .....	173
Figure 6-2	Typical Technical Planning Process Flow Tasks.....	176
Figure 6-3	Activity-on-Arrow and Precedence Diagrams for Network Schedules .....	180
Figure 6-4	Gantt chart created in Microsoft Project.....	182
Figure 6-5	Relationship Between a System, a Product Breakdown Structure, and a Work Breakdown Structure .....	193
Figure 6-6	Examples of WBS Development Errors .....	196
Figure 6-7	Requirements Management Process .....	205
Figure 6-8	Interface Management Process .....	212
Figure 6-9	Technical Risk Management Process .....	216
Figure 6-10	Scenario-Based Modeling of Hazards .....	218
Figure 6-11	Risk as a Set of Triplets .....	218
Figure 6-12	Continuous Risk Management.....	221
Figure 6-13	Risk-Informed Decision Analysis.....	222
Figure 6-14	Risk Analysis of Decision Alternatives .....	223
Figure 6-15	Risk Matrix .....	226
Figure 6-16	An Example of a Fault Tree.....	227
Figure 6-17	Deliberation.....	229
Figure 6-18	Performance Monitoring and Control of Deviations .....	232
Figure 6-19	Margin Management Method .....	233
Figure 6-20	CM Process .....	235
Figure 6-21	Five Elements of Configuration Management.....	236
Figure 6-22	Evolution of Technical Baseline.....	238

Figure 6-23	Typical Change Control Process.....	239
Figure 6-24	Technical Data Management .....	245
Figure 6-25	Technical Assessment Process.....	257
Figure 6-26	Planning and Status Reporting Feedback Loop .....	258
Figure 6-27	Cost and Schedule Variances.....	289
Figure 6-28	Relationships of MOEs, MOPs, and TPMs .....	293
Figure 6-29	Use of the Planned Profile Method for the Weight TPM with Rebaseline in Chandra Project.....	295
Figure 6-30	Use of the Margin Management Method for the Mass TPM in the Microrover Flight Experiment ( <i>Sojourner</i> ).....	296
Figure 6-31	Decision Analysis Process .....	301
Figure 6-32	Example Decision Matrix .....	306
Figure 6-33	Systems Analysis across the Life Cycle .....	310
Figure 6-34	Operations Modeling and Simulation Capabilities .....	310
Figure 6-35	Simulation Model Analysis Techniques .....	311
Figure 6-36	Trade Study Process.....	312
Figure 6-37	Trade Tree.....	314
Figure 6-38	Influence Diagrams.....	319
Figure 6-39	Decision Tree.....	320
Figure 6-40	Examples of Lower-Level Performance Measures.....	322
Figure 6-41	Utility Function for a “Volume” Performance Measure.....	323
Figure 6-42	Risk-informed Decision Analysis Process.....	325
Figure 6-43	An Example of an Objectives Hierarchy .....	326
Figure 7-1	Acquisition Life Cycle.....	330
Figure 7-2	Contract Requirements Development Process.....	335
Figure 7-3	CACE People/Process/Tools/Facility Paradigm.....	351
Figure 7-4	Human Factors Interaction Model. ....	363
Figure 1:	FFBD Flow Down.....	395
Figure 2:	FFBD—Example 1 .....	396
Figure 3:	FFBD showing additional control constructs—Example 2.....	396
Figure 4:	Enhanced Functional Flow Block Diagram—Example 3 .....	397
Figure 5:	Requirements Allocation Sheet.....	398
Figure 6:	N-Squared Diagram for Orbital Equipment .....	399
Figure 7:	Timing Diagram Example.....	400
Figure 8:	Slew Command Status State Diagram.....	401
Figure 1.	Products provided by the TA as a function of program/project phase (NPR 7120.5d)	404
Figure 2.	PBS Example.....	405
Figure 3.	Technology Assessment Process.....	407
Figure 4.	Architectural studies and technology development.....	407

Figure 5. Technology Readiness Levels .....	409
Figure 6. The TMA Assessment thought process .....	410
Figure 7. TRL Assessment Matrix .....	411

**Tables**

Table 2-1	Project Life-Cycle Phases .....	27
Table 4-1	Typical Operational Phases for a NASA Mission .....	70
Table 4-2	Benefits of Well-Written Requirements .....	73
Table 4-3	Requirements Meta-Data .....	82
Table 4-4	Key ILS Elements .....	110
Table 6-1	Technical Data Tasks .....	252
Table 6-2	Program Technical Reviews .....	263
Table 6-3	P/SRR Entrance and Success Criteria .....	266
Table 6-4	P/SDR Entrance and Success Criteria .....	267
Table 6-5	MCR Entrance and Success Criteria .....	268
Table 6-6	SRR Entrance and Success Criteria .....	269
Table 6-7	MDR Entrance and Success Criteria .....	270
Table 6-8	SDR Entrance and Success Criteria .....	271
Table 6-9	PDR Entrance and Success Criteria .....	272
Table 6-10	CDR Entrance and Success Criteria .....	274
Table 6-11	PRR Entrance and Success Criteria .....	275
Table 6-12	SIR Entrance and Success Criteria .....	276
Table 6-13	TRR Entrance and Success Criteria .....	277
Table 6-14	SAR Entrance and Success Criteria .....	279
Table 6-15	ORR Entrance and Success Criteria .....	280
Table 6-16	FRR Entrance and Success Criteria .....	281
Table 6-17	PLAR Entrance and Success Criteria .....	282
Table 6-18	CEER Entrance and Success Criteria .....	283
Table 6-19	PFAR Entrance and Success Criteria .....	283
Table 6-20	DR Entrance and Success Criteria .....	285
Table 6-21	Functional and Physical Configuration Audits .....	287
Table 6-22	Systems Engineering Process Metrics .....	298
Table 6-23	Consequence Table .....	303
Table 6-24	Typical Information to Capture in a Decision Report .....	307
Table 7-1	Applying the Technical Processes on Contract .....	331
Table 7-2	Steps in the Requirements Development Process .....	335
Table 7-3	Proposal Evaluation Criteria .....	339
Table 7-4	Risks in Acquisition .....	341
Table 7-5	Typical Work Product Documents .....	344

Table 7-6	Contract-Subcontract Issues.....	345
Table 7-7	Human and Organizational Analysis Techniques.....	364
Table 7-8	Planetary Protection Mission Categories .....	373
Table 7-9	Summarized Planetary Protection Requirements.....	373

## **Foreword**

**[to come]**

## Preface

Since the writing of SP-6105 in 1995, systems engineering at NASA, within national and international standard bodies, and as a discipline has undergone rapid evolution. Changes include implementing standards in the International Standards Organization (ISO) 9000, the use of Carnegie Mellon University's Capability Maturity Model<sup>®</sup> Integration<sup>SM</sup> (CMMI) to improve development and delivery of products, and the impacts caused by the Columbia failure. Lessons learned on systems engineering were documented in reports such as those by the NASA Integrated Action Team (NIAT), the Columbia Accident Investigation Board (CAIB), and the follow-on Diaz Report. Out of these efforts came the NASA Office of the Chief Engineer initiative to improve the overall Agency systems engineering infrastructure and capability for the efficient and effective engineering of NASA systems, to produce quality products, and achieve mission success. In addition, Agency policy and requirements for systems engineering have been approved.

In 1995, SP-6105 was initially published to bring the fundamental concepts and techniques of systems engineering to NASA personnel in a way that recognizes the nature of NASA systems and the NASA environment. This revision of SP-6105 maintains that original philosophy while updating the Agency's systems engineering body of knowledge, providing guidance for insight into current best Agency practices, and aligning the handbook with the new Agency systems engineering policy.

The update of this handbook was twofold—a top-down compatibility with higher level Agency policy and a bottom-up infusion of guidance from the NASA practitioners in the field. The approach provided the opportunity to obtain best practices from across NASA and bridge the information to the established NASA systems engineering process. The attempt is to communicate principles of good practice as well as alternative approaches rather than specify a particular way to accomplish a task. The result embodied in this handbook is a top-level implementation approach on the practice of systems engineering unique to NASA. The material for updating this handbook was drawn from many different sources, including NASA procedural requirements (NPRs), field center systems engineering handbooks and processes, as well as non-NASA systems engineering textbooks and guides.

This handbook consists of six core chapters: (1) systems engineering fundamentals discussion, (2) the NASA program/project life cycles, (3) systems engineering processes to get from a concept to a design, (4) systems engineering processes to get from a design to a final product, (5) crosscutting management processes in systems engineering, and (6) special topics relative to systems engineering. These core chapters are supplemented by appendices that provide outlines, examples, and further information to illustrate topics in the core chapters. The handbook makes extensive use of sidebars and figures to define, refine, illustrate, and extend concepts in the core chapters without diverting the reader from the main information.

The handbook provides top-level guidelines for good systems engineering practices; it is not intended in any way to be a directive.

SP-2007-6105 Rev 1 supersedes SP-6105, dated June 1995.

## **Acknowledgments**

The following individuals are acknowledged as contributing practitioners to the content of this revision:

**TBA**

## **CAS/NTIS Distribution**

Available from:

NASA Center for Aerospace Information  
7115 Standard Drive  
Hanover, MD 21076-1320  
301-621-0390

This report is also available in electronic form at

<http://>

## 1.0 Introduction

### 1.1 Purpose

This handbook is intended to provide general guidance and information on systems engineering that will be useful to the NASA community. It provides a generic description of systems engineering as *it should be* applied throughout NASA. A goal of the handbook is to increase awareness and consistency across the Agency and advance the practice of systems engineering. This handbook provides perspectives relevant to NASA and data particular to NASA.

This handbook should be used as a companion for implementing NPR 7123.1, *Systems Engineering Processes and Requirements*, as well as the Center-specific handbooks and directives developed for implementing systems engineering at NASA. It provides a companion reference book for the various systems engineering related courses being offered under NASA's auspices.

### 1.2 Scope and Depth

The coverage in this handbook is limited to general concepts and generic descriptions of processes, tools, and techniques. It provides information on systems engineering best practices, and pitfalls to avoid. There are many Center-specific handbooks and directives, as well as textbooks that can be consulted for in-depth tutorials.

This handbook describes systems engineering as it should be applied to the development of large and small NASA programs and projects. NASA has defined three different life cycles that specifically address the three major project categories, or product lines, which are: Flight Systems and Ground Support (FS&GS), Research and Technology (R&T), and Construction of Facilities (CoF) and Environmental Compliance and Restoration (ECR). The technical content of the handbook provides systems engineering best practices that should be incorporated into all NASA product lines. For simplicity this handbook uses the FS&GS product line as an example. The specifics of FS&GS can be seen in the description of the life cycle and the details of the milestone reviews. Each product line will vary in these two areas; therefore, the reader should refer to the applicable NPRs for the specific requirements for their life cycle and reviews. The engineering of NASA systems requires a systematic and disciplined set of processes that are applied recursively and iteratively for the design, development, operation, maintenance, and closeout of systems throughout the life cycle of the programs and projects.

The handbook's scope properly includes systems engineering functions regardless of whether they are performed by a manager or an engineer, in-house or by a contractor.

## 2.0 Fundamentals of Systems Engineering

Systems engineering (SE) is a methodical, disciplined approach for the design, realization, technical management, operations, and retirement of a system (product or service). It is a way of looking at the “big picture” when making technical decisions. It is a way of achieving stakeholder functional, physical, and operational performance requirements in the intended use environment over the planned life of the systems. In other words, systems engineering is a logical way of thinking.

Systems engineering is the art and science of developing an operable system capable of meeting requirements within opposed constraints. Systems engineering is a holistic, integrative discipline, wherein the contributions of structural engineers, electrical engineers, mechanism designers, power engineers, human factor engineers, and many more disciplines are evaluated and balanced, one against another, to produce a coherent whole that is not dominated by the perspective of a single discipline.

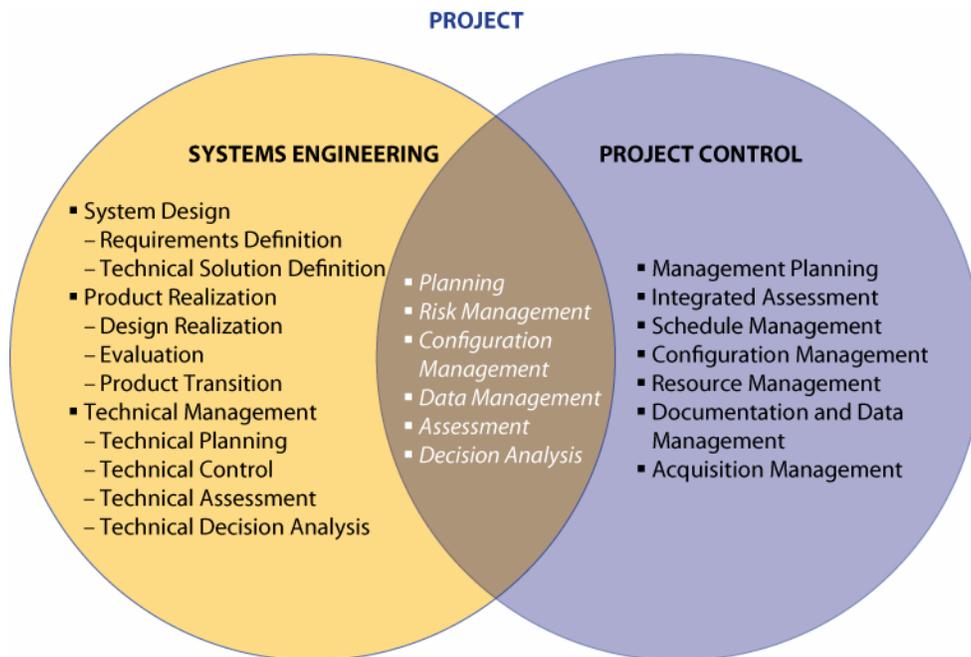
Systems engineering seeks a safe and balanced design in the face of opposing interests and multiple, sometimes conflicting constraints. The systems engineer must develop the skill and instinct for identifying and focusing efforts on assessments to optimize the overall design and not favor one system at the expense of another. The art is in knowing when and where to probe. Personnel with these skills are usually tagged as “systems engineers.” They may have other titles—lead systems engineer, technical manager, chief engineer—but for this document, we will use the term systems engineer.

The exact role and responsibility of the systems engineer may change from project to project depending on the size and complexity of the project, and from phase to phase of the life cycle. For large projects, there may be one or more systems engineers. For small projects, sometimes the project manager may perform these practices. But, whoever assumes those responsibilities, the systems engineering functions must be performed. The actual assignment of the roles and responsibilities of the named systems engineer may also therefore vary. The lead systems engineer ensures that the system technically fulfills the defined needs and requirements and that a proper systems engineering approach is being followed. The systems engineer oversees the project’s systems engineering activities as performed by the technical team and directs, communicates, monitors and coordinates tasks. The systems engineer reviews and evaluates the technical aspects of the project to ensure that the systems/subsystems engineering processes are functioning properly and evolves the system from concept to product. The entire technical team is involved in the systems engineering process.

The systems engineer will usually play the key role in leading the development of the system architecture, defining and allocating requirements, evaluating design tradeoffs, balancing technical risk between systems, defining and assessing interfaces, providing oversight of verification and validation activities as well as many other tasks. The systems engineer will usually have the prime responsibility in developing many of the project documents, including the SEMP, requirements/specification documents, V&V documents, certification packages, and other technical documentation.

In summary, the systems engineer is a generalist skilled in the art and science of balancing organizational and technical interactions in complex systems. However, since the entire team is involved in the systems engineering approach, in some ways everyone is a systems engineer. Systems engineering is about tradeoffs and compromises, about generalists rather than specialists. Systems engineering is about looking at the “big picture” and not only ensuring that they get the design right (meet requirements) but that they get the right design.

To explore this further, put SE in the context of project management. As discussed in *NPR 7120.5, NASA Space Flight Program and Project Management Requirements*, project management is the function of planning, overseeing, and directing the numerous activities required to achieve the requirements, goals, and objectives of the customer within specified cost, quality, and schedule constraints. Project management can be thought of as having two major areas of emphasis, both of equal weight and importance. These areas are systems engineering and project control. Figure 2-1 is a notional graphic depicting this concept. Note that there are areas where the two cornerstones of project management overlap. In these areas, SE provides the technical aspects or inputs; whereas, project control provides the programmatic, cost, and schedule inputs.

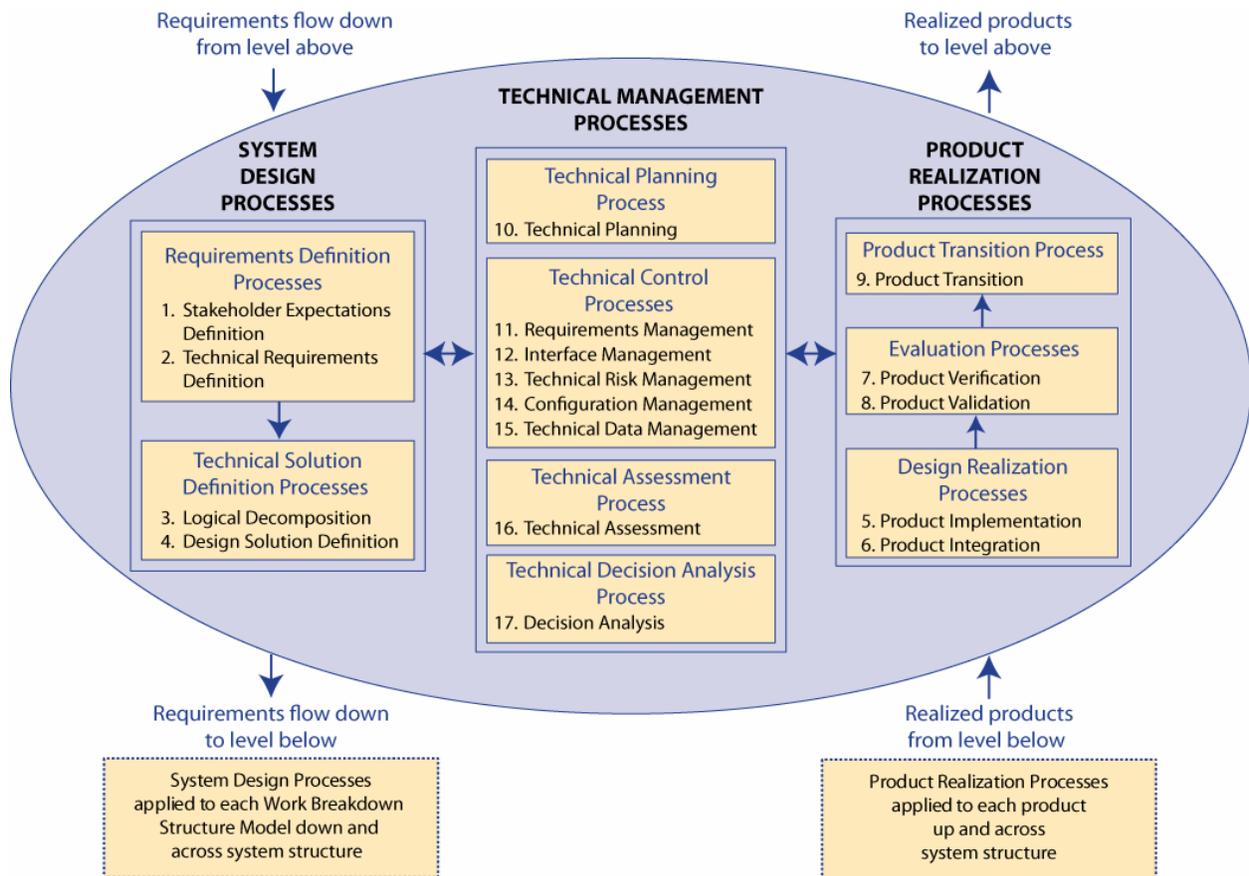


**Figure 2-1 Context of SE within Project Scope**

This document will focus on the SE side of the diagram. These practices/processes are taken from *NPR 7123.1, NASA Systems Engineering Processes and Requirements*. Each will be described in much greater detail in subsequent chapters of this document, but an overview is given below.

## 2.1 The Common Technical Processes and the Systems Engineering Engine

There are three sets of common technical processes in *NPR7123.1, NASA Systems Engineering Processes and Requirements*: system design, product realization, and technical management. The processes in each set and their interactions and flows are illustrated by the NPR systems engineering “engine” shown in Figure 2-2. The processes of the SE engine are used to develop and realize the end products. This chapter provides the application context of the 17 common technical processes required in NPR 7123.1.



**Figure 2-2 The Systems Engineering Engine**

**System Design Processes:** The four system design processes are used to define and baseline stakeholder expectations, generate and baseline technical requirements, and convert the technical requirements into a design solution that will satisfy the baselined stakeholder expectations. These processes are applied to each product of the system structure from the top of the structure to the bottom until the lowest products in any system structure branch are defined to the point where they can be built, bought, or reused. All other products in the system structure are realized by integration. Designers not only develop the design solutions to the products intended to perform the operational functions of the system, but also establish requirements for the products and services that enable each operational/mission product in the system structure.

**Product Realization Processes:** The product realization processes are applied to each operational/mission product in the system structure starting from the lowest level product and working up to higher level integrated products. These processes are used to create the design solution for each product (e.g., by the Product Implementation or Product Integration Process) and to verify, validate, and transition up to the next hierarchical level products that satisfy their design solutions and meet stakeholder expectations as a function of the applicable life-cycle phase.

**Technical Management Processes:** The technical management processes are used to establish and evolve technical plans for the project, to manage communication across interfaces, to assess progress against the plans and requirements for the system products or services, to control technical execution of the project through to completion, and to aid in the decisionmaking process.

The processes within the SE engine are used both iteratively and recursively. As defined in NPR 7123.1, “iterative” is the “application of a process to the same product or set of products to correct a discovered discrepancy or other variation from requirements.” Whereas “recursive” is defined as adding value to the system “by the repeated application of processes to design next lower layer system products or to realize next upper layer end products within the system structure. This also applies to repeating application of the same processes to the system structure in the next life-cycle phase to mature the system definition and satisfy phase exit criteria.” The example used in section 2.3 will further explain these concepts. The technical processes are applied recursively and iteratively to break down the initializing concepts of the system to a level of detail concrete enough that the technical team can implement a product from the information. Then the processes are applied recursively and iteratively to integrate the smallest product into greater and larger systems until the whole of the system has been assembled, verified, validated and transitioned.

## **2.2 Comparison of the SE Engine with the “Vee”**

The systems engineering “vee” has been used and modified since its introduction by Forsberg/Mooz in 1990. Forsberg and Mooz describe what they call “the technical aspect of the project cycle” by a vee-shaped chart, starting with user needs on the upper left and ending with a user-validated system on the upper right. On the left side of the vee, decomposition and definition activities resolve the system architecture, creating the details of the design. Integration and verification flow up and to the right as successively higher levels of subsystems are verified, culminating at the system level.

The basic concept the vee tries to convey is the flow down and recursive refinement of requirements on the left side and the gradual buildup and test of the products on the right. In this, the vee and the SE engine are conveying similar concepts. However, the SE engine goes beyond the vee concepts to also convey the full range of practices and processes that need to occur to ensure a good product has been engineered. The SE engine is therefore the tool that is used in NPR 7123.1 as well as this document to more fully describe the framework and processes that are used to properly engineer a system.

As will be seen in the next section, the SE engine conveys this 3-D aspect through the flow-down and flow-up arrows shown in the engine.

### 2.3 Example of Using the SE Engine

To help in understanding how the SE engine is applied, an example will be posed and walked through the processes. Pertinent to this discussion is the phases of the program and project life cycle, which will be discussed in greater depth in chapter 3 of this document. As described in chapter 3, NPR 7120.5 defines the life cycle used for NASA programs and projects. The life-cycle phases are described in Table 2-1.

**Table 2-1 Project Life-Cycle Phases**

Phase		Purpose	Typical Output
Formulation	Pre-Phase A Concept Studies	To produce a broad spectrum of ideas and alternatives for missions from which new programs/projects can be selected. Determine feasibility of desired system, develop mission concepts, draft system-level requirements, identify potential technology needs	Feasible system concepts in the form of simulations, analysis, study reports, models, mockups
	Phase A Concept and Technology Development	To determine the feasibility and desirability of a suggested new major system and establish an initial baseline compatibility with NASA's strategic plans. Develops final mission concept, system - level requirements and needed system structure technology developments	System concept definition in the form of simulations, analysis, engineering models, mockups
	Phase B Preliminary Design and Technology Completion	To define the project in enough detail to establish an initial baseline capable of meeting mission needs. Develops system structure end product (and their enabling product) requirements and generates a preliminary design for each system structure end product	End products in the form of mockups and prototypes
Implementation	Phase C Final Design and Fabrication	To complete the detailed design of the system (and its associated subsystems, including its operations systems) and fabricate (hardware) and code (software). Generates final designs for each system structure end product	End product detailed designs
	Phase D System Assembly, Integration and Test, Launch	To assembly and integrate the products to create the system, meanwhile developing confidence that it will be able to meet the system requirements. Launch and prepare for operations. Performs system end product implementation, assembly, integration, test and transition to use	Operations-ready system end product with supporting related enabling products
	Phase E Operations	To conduct the mission and meet the initially identified need and maintain	Desired system

Phase		Purpose	Typical Output
	and Sustainment	support for that need. Implements the mission operations plan	
	Phase F Closeout	Implement the Systems Decommissioning/ Disposal Plan developed in Phase E and performs analyses of the returned data and any returned samples	Product close out

Use of the different phases of a life cycle allows the various products of a project to be gradually developed and matured from initial concepts through the fielding of the product and to its final retirement. The SE engine shown in Figure 2-2 is used throughout all phases. In Pre-Phase A, it is used to develop the initial concepts; develop a preliminary/draft set of key high-level requirements; realize these concepts through modeling, mockups, simulation or other means; and verify and validate that these concepts and products would be able to meet the key high-level requirements. Note that this is not the formal verification and validation (i.e., certification) program that will be performed on the final product, but is a methodical run-through ensuring that the concepts that are being developed in this Pre-Phase A would be able to meet the likely requirements and expectations of the stakeholders. Concepts would be developed to the lowest level necessary to ensure that the concepts are feasible and to a level that will reduce the risk low enough to satisfy the project. Academically, this process could proceed down to the circuit board level for every system. However, that would involve a great deal of time and money. There may be a higher level or tier of product than circuit board level that would enable designers to accurately determine the feasibility of accomplishing the project (purpose of Pre-Phase A).

During Phase A, the recursive use of the SE engine is continued, this time taking the concepts and draft key requirements that were developed and validated during Pre-Phase A, and fleshing them out to become the set of baseline system requirements and operational concepts. During this phase, key areas of high risk might be simulated or prototyped to ensure that the concepts and requirements being developed are good ones and to identify verification and validation tools and techniques that will be needed in later phases.

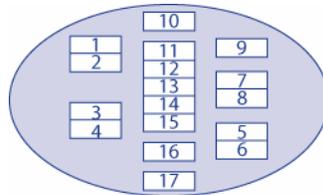
During Phase B, the SE engine is applied recursively to further mature requirements for all products in the developing product tree, develop operational concepts' preliminary designs, and perform run throughs of the verification and validation concepts to ensure the designs will likely be able to meet their requirements.

Phase C again uses the left side of the SE engine to finalize all requirement updates, finalize operational concepts, develop the final designs to the lowest level of the product tree, and begin fabrication. Phase D uses the right side of the SE engine to recursively perform the final implementation, integration, verification, and validation of the end product and, at the final pass, transition the end product to the user.

### 2.3.1 Detailed Example

Since it is already well known, the NASA space transportation system will be used as an example to look at how the SE engine would be used in Phase A. This example will be

simplified to illustrate the application of the SE processes in the engine, but will in no way be as detailed as necessary to actually build the highly complex vehicle. The SE engine is used recursively to drive out more and more detail with each pass. The icon shown in Figure 2-3 will be used to keep track of the applicable place in the SE engine. The numbers in the icon correspond to the numbered processes within the SE engine as shown in Figure 2-2. The various layers of the product hierarchy will be called “tiers.” Tiers are also called “layers,” or “levels.” But basically the higher the number of the tier or level, the lower in the product hierarchy the product is going and the more detailed the product is becoming (e.g., going from boxes, to circuit boards, to components)

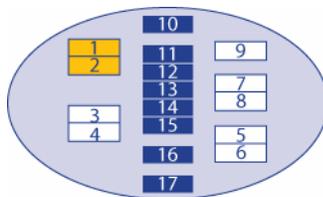


**Figure 2-3 SE Engine Tracking Icon**

### 2.3.2 Example Premise

NASA decides that there is a need for a transportation system that will act like a “truck” to carry large pieces of equipment and crew into low Earth orbit. Referring back to the project life cycle, the project first enters the Pre-Phase A. During this phase, several concept studies are performed, and it is determined that it is feasible to develop such a “space truck.” This is determined through combinations of simulations, mockups, analyses, or other like means. For simplicity, assume feasibility will be proven through concept models. The processes and framework of the SE engine will be used to design and implement these models. The project would then enter the Phase A activities to take the Pre-Phase A concepts and refine them and define the system requirements for the end product. The detailed example will begin in Phase A and show how the SE engine is used. As described in the overview, a similar process is used for the other project phases.

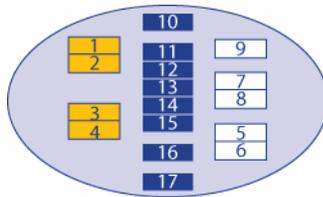
#### 2.3.2.1 Phase A System Design Passes



**1<sup>st</sup> Pass:** Taking the preliminary concepts and draft key system requirements developed during the Pre-Phase A activities, the SE engine is entered at the first process and used to determine who the product (i.e., the STS system) stakeholders are and what they want. During Pre-Phase A these needs and expectations were pretty general ideas, probably just saying the Agency needs a “space truck” that will

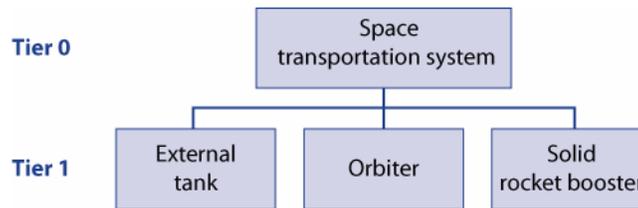
carry X tons of payload into low Earth orbit, accommodate a payload of so and so size, carry a crew of seven, etc. During this Phase A pass, these general concepts are detailed out and agreed-to. The “concept of operations” (sometimes referred to as a operational concept) generated in Pre-Phase A is also detailed out and agreed-to to ensure all stakeholders are in agreement as to what is really expected of the product—in this case the transportation system. The detailed expectations are then converted into good requirement statements. Subsequent passes and subsequent phases will refine these requirements into specifications that can actually be built.

Also note that all of the technical management processes (SE engine processes numbered 10 through 17) are also used during this and all subsequent passes and activities. These ensure that all the proper planning, control, assessment and decisions are used and maintained. Although for simplification they will not be mentioned in the rest of this example, they will *always* be in effect.



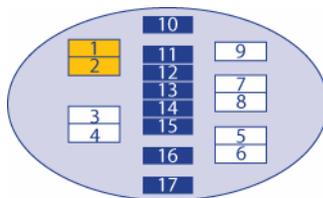
Next, using the requirements and the concept of operation (ConOps) previously developed, logical decomposition models/diagrams are built up to help bring the requirements into perspective and to show their relationship. Finally, these diagrams, requirements, and ConOps documents are used to develop one or more feasible design solutions.

Note that at this point, since this is only the first pass through the SE engine, these design solutions are not detailed enough to actually build anything. Consequently, the design solutions might be summarized –as, “To accomplish this transportation system, the best option in our trade studies is a three-part system: a reusable orbiter for the crew and cargo, a large external tank to hold the propellants, and two solid rocket boosters to give extra power for liftoff that can be recovered, refurbished, and reused.” (Of course, the actual design solution would be much more descriptive and detailed). So, for this first pass, the first tier of the product hierarchy might look like Figure 2-4. There would also be other enabling products that might appear in the product tree, but for simplicity only, the main products are shown in this example.



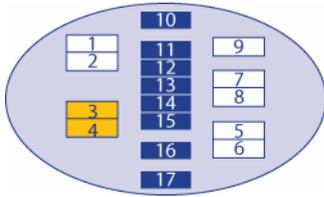
**Figure 2-4 Product Hierarchy Through One Pass of the SE Engine**

Now, obviously design solution is not yet at a detailed enough level to actually build the prototypes or models of any of these products. The requirements, ConOps, functional diagrams, and design solutions are still at a pretty high, general level. Note that the SE processes on the right side (i.e., the Product Realization Processes) of the SE engine have yet to be addressed. The design must first be at a level that something can actually be built, coded, or reused before that side of the SE engine can be used. So, a second pass of the left side of the SE engine will be started.



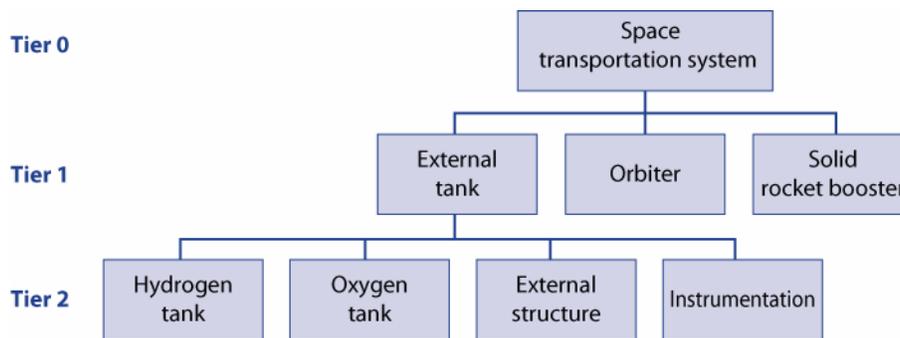
**2nd Pass:** The SE engine is completely recursive. That is, each of the three elements shown in the tier 1 diagram can now be considered a product of its own and the SE engine is therefore applied to each of the three elements separately. For example, the external tank is considered an end product and the SE engine resets back to the first processes. So now, just focusing on the external tank, who are the stakeholders and what they expect of the external tank is determined. Of course, one of the main stakeholders will be the owners of the tier 1 requirements and the STS as an end product, but there will also be other new stakeholders. A new ConOps for how the external tank would

operate is generated. The tier 1 requirements that are applicable (allocated) to the external tank would be “flowed down” and validated. Usually, some of these will be too general to implement into a design, so the requirements will have to be detailed out. To these derived requirements, there will also be added new requirements that are generated from the stakeholder expectations, and other applicable standards for workmanship, safety, quality, etc.

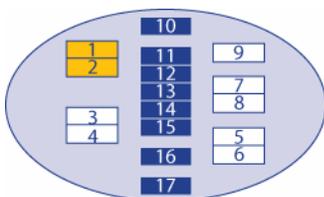


Next, the external tank requirements and the external tank ConOps are established, and functional diagrams are developed as was done in the first pass with the STS product. Finally, these diagrams, requirements, and ConOps documents are used to develop some feasible design solutions for the external tank. At this pass, there will also not be enough detail to actually build or prototype the external tank. The

design solution might be summarized as, “To build this external tank, since our trade studies showed the best option was to use cryogenic propellants, a tank for the liquid hydrogen will be needed as will another tank for the liquid oxygen, instrumentation, and an outer structure of aluminum coated with foam.” Thus, the tier 2 product tree for the external tank might look like Figure 2-5.

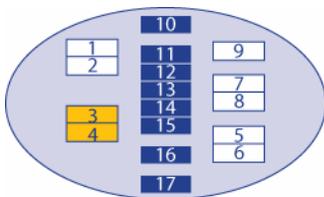


**Figure 2-5 Tier 2 Product Hierarchy for the External Tank**



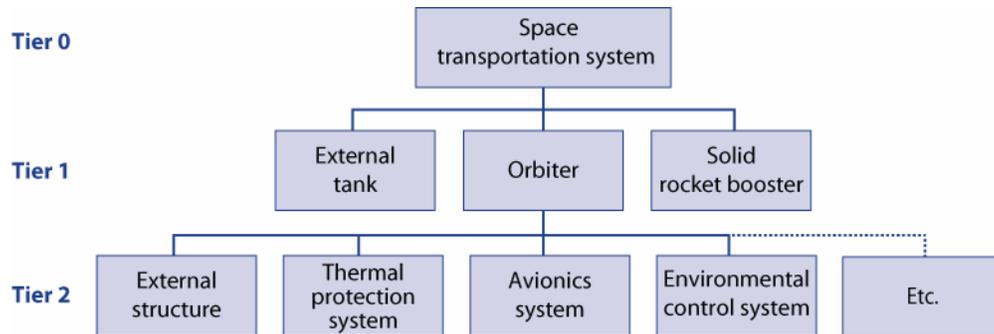
other elements) added.

In a similar manner, the orbiter would also take another pass through the SE engine starting with identifying the stakeholders and their expectations, and generating a ConOps for the orbiter element. The tier 1 requirements that are applicable (allocated) to the orbiter would be “flowed down” and validated, new requirements derived from them and any additional requirements (including interfaces with the



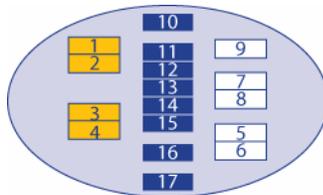
Next, the orbiter requirements and the ConOps are taken and functional diagrams are developed and one or more feasible design solutions for the orbiter are generated. As with the external tank, at this pass, there will not be enough detail to actually build or do a complex model of the orbiter. The orbiter design solution might be summarized as, “To build this orbiter, will require a winged vehicle

with a thermal protection system; an avionics system; a guidance, navigation and control system; a propulsion system; an environmental control system; etc.” So the tier 2 product tree for the orbiter element might look like Figure 2-6.



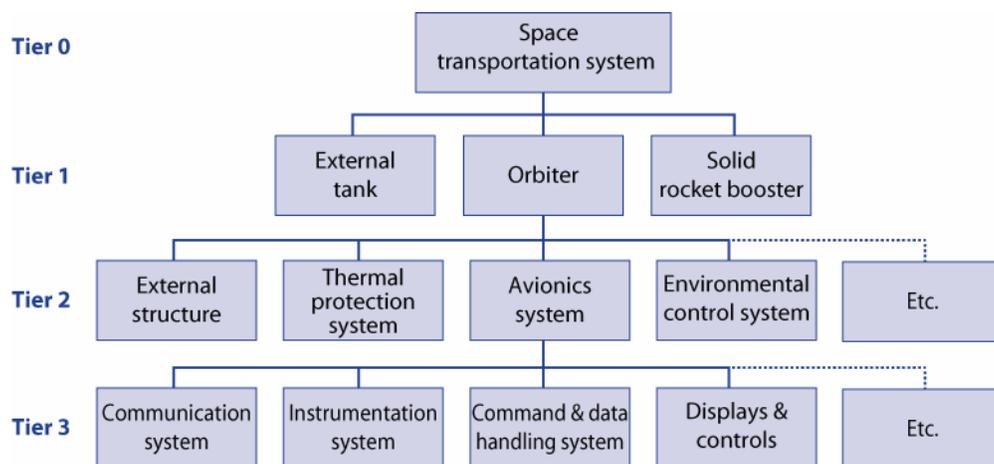
**Figure 2-6 Example Orbiter Tier 2 Product Hierarchy**

Likewise, the solid rocket booster would also be considered an end product and a pass through the SE engine would generate a tier 2 design concept, just as was done with the external tank and the orbiter.

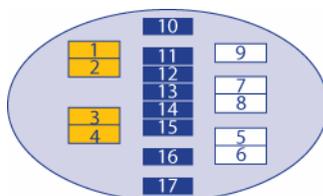


**3rd Pass:** Each of the tier 2 elements are also considered end products and each undergoes another pass through the SE engine, defining stakeholders, generating ConOps, flowing down allocated requirements, generating new and derived requirements and developing functional diagrams and design solution concepts. As an example of just the avionics system element, the tier 3 product

hierarchy tree might look like Figure 2-7.

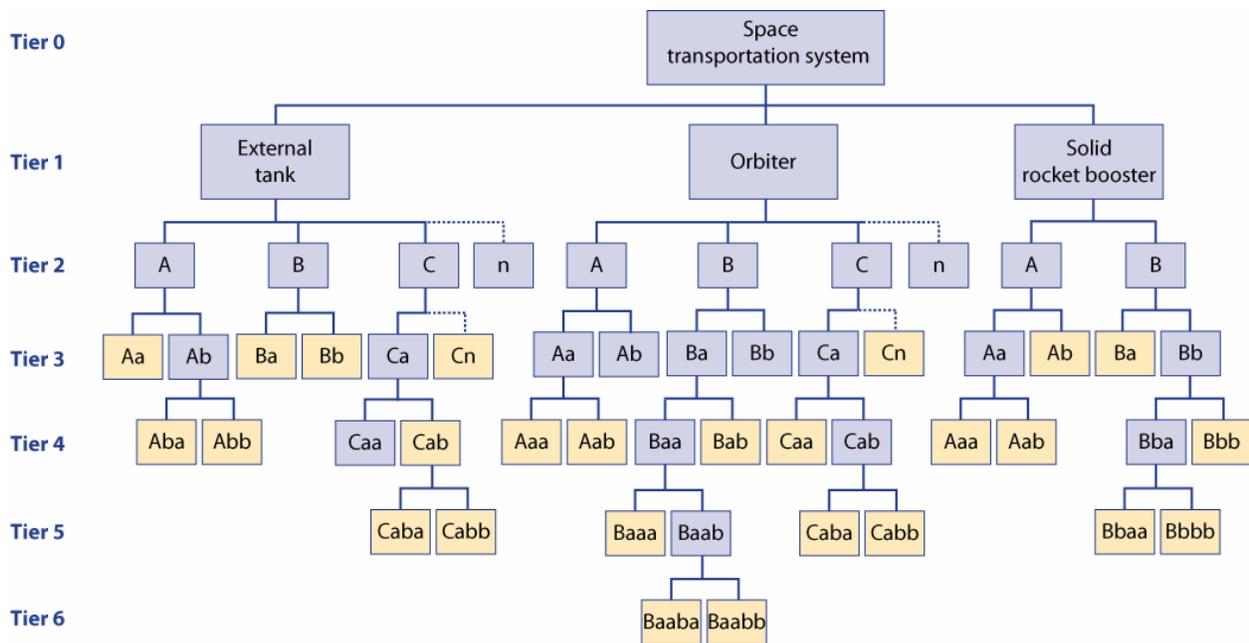


**Figure 2-7 Tier 3 Product Hierarchy Example for Avionics System**



**Pass 4 thru n:** For this Phase A set of passes, this recursive process is continued for each product (model) on each tier down to the lowest level in the product tree. Note that in some projects it may not be

feasible, given an estimated project cost and schedule, to perform this recursive process completely down to the smallest component during Phase A. So, engineering judgment must be used to determine how low in the product tier is feasible. Note that the lowest feasible level may occur at different tiers depending on the product line complexity. For example, for one product line it may occur at tier 2; whereas, for a more complex product, it could occur at tier 8. This also means that it will take different amounts of time to reach the bottom. Thus, for any given program or project, products will be at various stages of development. For this example, assume the completed STS product hierarchy is as depicted in Figure 2-8. At the end of this set of passes, system requirements, operational concepts, and high-level conceptual functional and physical architectures for each product in the tree would exist. Note that these would not yet be the detailed or even preliminary designs for the end products. These will come later in the life cycle. At this point, enough conceptual design work has been done to ensure that at least the high-risk requirements are achievable as will be shown in the following passes.



**Figure 2-8 Example Phase A Product Hierarchy after Completely Passing Through the System Design Process Side of the SE Engine**

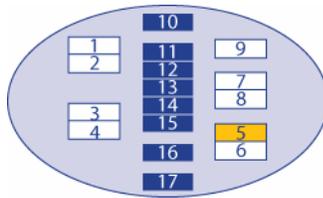
The shaded boxes represent bottom-level end products.

### 2.3.2.2 Product Realization Passes

So now that the requirements and conceptual designs for the high-risk Phase A products have been developed, they need to be checked to ensure they are achievable. Note that there are two types of products. The first product is the “end product” that is being designing for—the one that will actually be delivered to the final user. The second type of product will be called a “phase product.” A phase product is generated within a particular life-cycle phase that helps move the project toward delivering a final product. For example, while in Pre-Phase A, a foam-core mockup might be built to help visualize some of the concepts. Those mockups would not be the final “end product,” but would be the “phase product.” For this Phase A example, assume some

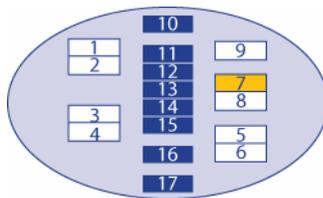
computer models will be created of these key concepts to show that they are achievable. These will be the phase product for our example.

Now the focus shifts to the right side (i.e., product realization processes) of the SE engine which will be applied recursively, starting at the bottom of the product hierarchy and moving upwards.



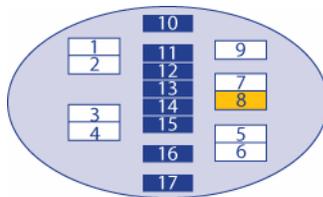
**1<sup>st</sup> Pass:** Each of the phase products, (i.e., our computer models) for the bottom-level product tier (ones that are shaded in Figure 2-8) are taken individually and realized—that is they are either bought, built, coded or reused. For our example, assume the external tank product model Aa is a standard commercial-off-the-shelf (COTS) product that is bought. Aab is a model that can be reused from another project, and

product Abb is a model that will have to be developed with an in-house design that is now built. Note that these models are parts of a larger model product that will be assembled or integrated on a subsequent run through of the SE engine. That is, to realize the model for product Ab of the external tank, models for products Aab and Abb must be first implemented and then later integrated together. This pass of the SE engine will be the realizing part. Likewise, each of the shaded bottom-level model products is realized in this first pass. The models will help us understand and plan the method to implement the final end product and will ensure the feasibility of the implemented method.



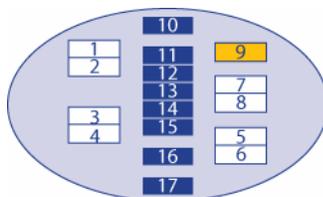
Next, each of the realized models (phase products) are used to verify that the end product would likely meet the requirements as defined in the Technical Requirements Definition Process during the system design pass for this product. This shows the product would likely meet the “shall” statements that were allocated, derived, or generated for it by method of test, analysis, inspection, or demonstration—that it

was “built right.” Verification is performed for each of the shaded bottom-level model products. Note that during this Phase A pass, this process is not the formal verification (certification) of the final end product. But using analysis, simulation, models, or other means shows that the requirements are good (verifiable) and that the concepts will most likely satisfy them. This also allows draft verification procedures of key areas to be developed. What can be formally verified, however, is that the phase product (the model) meets the requirements for the model.



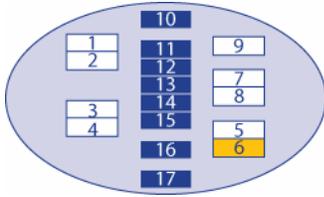
After the phase product (models) have been verified and used for planning the end product verification, the models are then used for validation. That is, additional test, analysis, inspection or demonstrations are conducted to ensure that the proposed conceptual designs will likely meet the expectations of the stakeholders for this phase product and for the end product. This will track back to the

ConOps that was mutually developed with the stakeholders during the Stakeholder Expectation Process of the system design pass for this product. This will help ensure that the project has “built the right” product at this level.



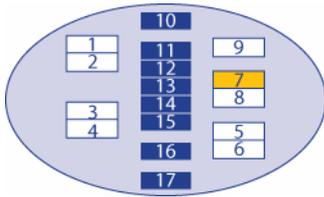
After verification and validation of the phase product (models) and using them for planning the verification and validation of the end product, its time to prepare the model for transition to the next level

up. Depending on complexity, where the model is to be transitioned to, security requirements, etc., transition may involve crating and shipment, transmitting over a network, or hand carrying over to the next lab. Whatever is appropriate, each model for the bottom-level product is prepared and handed to the next level up for further integration.



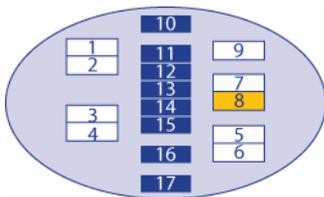
**2<sup>nd</sup> Pass:** Now that all the models (phase products) for the bottom-level end products are realized, verified, validated, and transitioned, its time to start integrating them into the next higher level product. For example, for the external tank, realized tier 4 models for product Aba and Abb are integrated to form the model for the Tier 3 product Ab. Note that the Product Implementation Process only occurs at the

bottom-most product. All subsequent passes of the SE engine will employ the Product Integration Process since already realized products will be integrated to form the new higher level products. Integrating the lower-tier phase products will result in the next-higher-tier phase product. This integration process can also be used for planning the integration of the final end products.



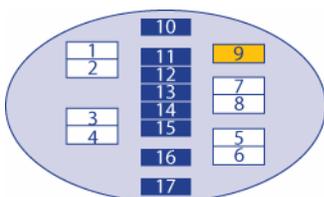
After the new integrated phase product (model) has been formed (tier 3 product Ab for example), it must now be proven that it meets its requirements. These will be the allocated, derived, or generated requirements developed during the Technical Requirements Definition Process during the system design pass for the model for this integrated product. This ensures that the integrated product was

built (assembled) right. Note that just verifying the component parts ( i.e., the individual models) that were used in the integration is not sufficient to assume that the integrated product will work right. There are many sources of problems that could occur— incomplete requirements at the interfaces, wrong assumptions during design, etc. The only sure way of knowing if an integrated product is good is to perform verification and validation at each stage. The knowledge gained from verifying this integrated phase product can also be used for planning the verification of the final end products.



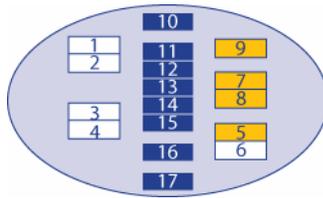
Likewise, after the integrated phase product is verified, it needs to be validated to show that it meets the expectations as documented in the ConOps for the model of the product at this level. Even though the component parts making up the integrated product will have been validated at this point, the only way to know that the project has built the “right” integrated product is to perform validation on the

integrated product itself. Again, this information will help in the planning for the validation of the end products.



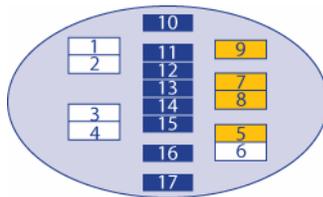
The model for the integrated phase product at this level (tier 3 product Ab for example) is now ready to be transitioned to the next higher level (tier 2 for the example). As with the products in the first pass, the integrated phase product is prepared according to its needs/requirements and shipped or handed over. In the example, the model for the external tank tier 3 integrated product Ab is transitioned

to the owners of the model for the tier 2 product A. This effort with the phase products will be useful in the planning for the transition of the end products.



**Pass 3 thru n:** In a similar manner as for pass 3, the models for the products are integrated together, verified, validated, and transitioned to the next higher tier. For the example, the model for external tank tier 3 integrated phase product Ab is integrated with the model for tier 3 realized phase product Aa to form the tier 2 phase product A. Note that tier 3 product Aa is a bottom-tier product that has yet to go

through the integration process. It may also have been realized some time ago and has been waiting for the Ab product line to become realized. Part of its transition might have been to place it in secure storage until the Ab product line became available. Or, it could be that Aa was the long-lead item and product Ab had been completed some time ago and was waiting for the Aa purchase to arrive before they could be integrated together. The length of the branch of the product tree does not necessarily translate to a corresponding length of time. This is why good planning in the first part of a project is so critical.



**Final Pass:** At some point all the models for the tier 1 phase products will each have been used to ensure the system requirements and concepts developed during this Phase A cycle can be assembled/integrated, verified, validated, and transitioned. The elements are now defined as the external tank, the orbiter, and the solid rocket boosters. One final pass through the SE engine will show

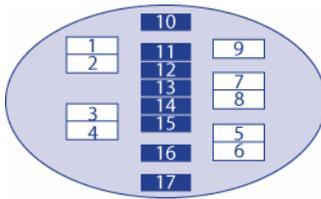
that they will likely to be successfully integrated, verified, and validated. The final transition of these products—the baselined system requirements, ConOps, conceptual functional and physical designs—are made to provide inputs into the next life-cycle phase (B) where they will be further matured.

### 2.3.2.3 Example in Phases B through D

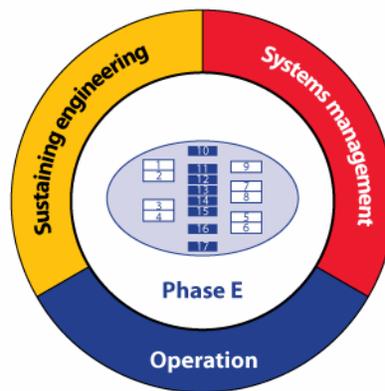
Phase B begins the preliminary design of the final end product. The recursive passes through the SE engine are repeated in a similar manner to that discussed in the detailed Phase A example. At this phase, the phase product might be a prototype of the product(s). Prototypes could be developed and then put through the planned verification and validation processes to ensure the design will likely meet all the requirements and expectations prior to the build of the final flight units. Any mistakes found on prototypes are much easier and less costly to correct than if not found until the flight units are built and in certification.

Whereas the previous phases dealt with the final product in the form of analysis, concepts, or prototypes, phases C and D work with the final end product itself. During Phase C we recursively use the left side of the SE engine to develop the final design. In Phase D we recursively use the right side of the SE engine to realize the final product and conduct the formal verification and validation of the final product. As we come out of the last pass of the SE engine in Phase D, we have the final fully realized end product, the STS system, ready to be delivered for launch

### 2.3.2.4 Use of the SE Engine in Phase E and F

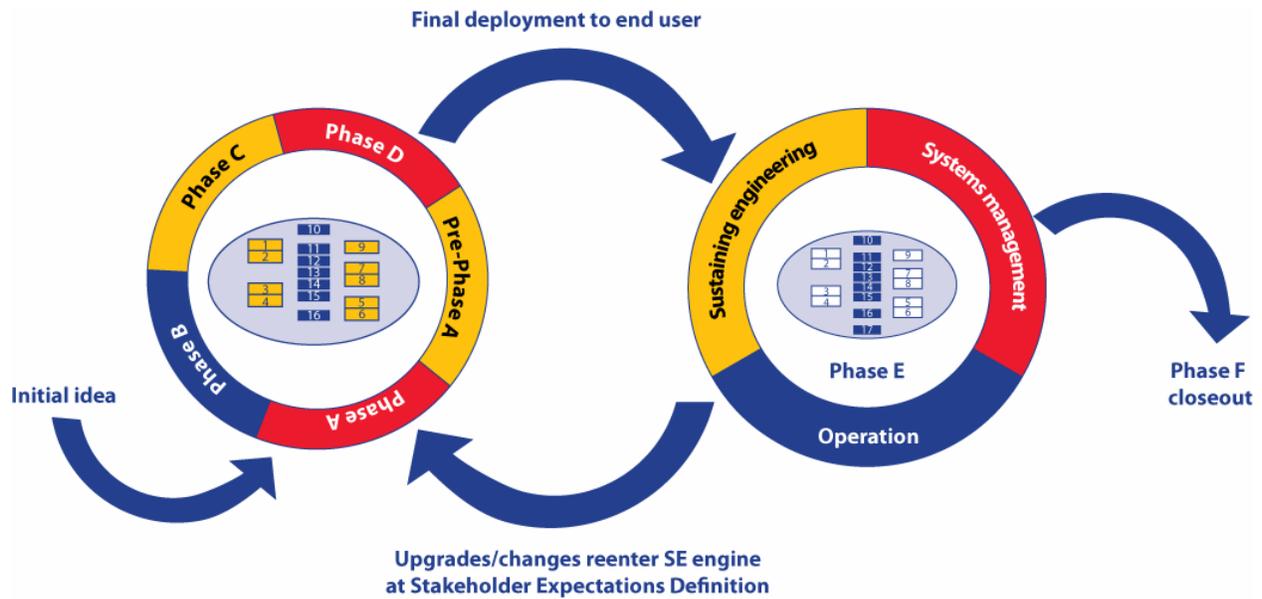


Even in these later phases of the life cycle, the technical management processes in the SE engine are still being used. During the operations phase of a project, a number of activities are still going on. In addition to the day-to-day use of the product, there is a need to monitor or manage various aspects of the system. This is where the key performance parameters (KDPs) and technical performance measures (TPMs) that were defined in the early stages of development continue to play a part. These are great measures to monitor to ensure the product continues to perform as designed and expected. Configurations are still under control, still executing the configuration management processes. Decisions are still being made using the decision analysis processes. Indeed all of the technical management processes still apply. For this discussion, the term “systems management” will be used for this aspect of operations. In addition to systems management and systems operation there may also be a need for periodic refurbishment, repairing broken parts, cleaning, sparing, logistics or other activities. Although other terms are used, for the purposes of this discussion the term “sustaining engineering” will be used for these activities. Again, all of the technical management processes still apply to these activities. Figure 2-9 represents these three activities occurring simultaneously and continuously throughout the operational lifetime of the final product.



**Figure 2-9 Model of Typical Activities During Operational Phase (Phase E) of a Product**

However, if at any point in this phase a new product, a change that affects the design or certification of a product, or an upgrade to an existing product is needed, the development processes of the SE engine are re-entered at the top. That is, the first thing that is done for an upgrade is to determine who the stakeholders are and what they expect. The entire SE engine is used just as for a newly developed product. This might be pictorially portrayed as in Figure 2-10. Note that in the figure although the SE engine is shown only once, it is used recursively down through the product hierarchy for upgraded products, just as described in our detailed example for the initial product.



**Figure 2-10 New Products or Upgrades Reentering the SE Engine**

## 2.4 Distinctions Between Product Verification and Validation

From a process perspective, Product Verification and Product Validation may be similar in nature, but the objectives are fundamentally different. Verification of a product shows proof of compliance with requirements—that the product can meet each “shall” statement as proven through performance of a test, analysis, inspection or demonstration. Validation of a product shows that the product accomplishes the intended purpose in the intended environment—that it meets the expectations of the customer as shown through performance of a test, analysis, inspection or demonstration.

Verification testing relates back to the approved requirements set and can be performed at different stages in the product life cycle. The specifications and design documents establish the configuration baseline of that product, which may have to be modified at a later time. Without a verified baseline and appropriate configuration controls, later modifications could be costly or cause major performance problems.

Validation relates back to the ConOps document. Validation testing is conducted under realistic conditions (or simulated conditions) on end products for the purpose of determining the effectiveness and suitability of the product for use in mission operations by typical users.

The selection of the verification or validation method is based on engineering judgment as to which is the most cost effective way to reliably show the products conformance to requirements or that it will operate as intended and described in the ConOps.

## 2.5 Cost Aspect of Systems Engineering

The objective of systems engineering is to see that the system is designed, built, and operated so that it accomplishes its purpose safely in the most cost-effective way possible considering performance, cost, schedule, and risk.

**Cost.** The cost of a system is the value of the resources needed to design, build, and operate it. Because resources come in many forms—work performed by NASA personnel and contractors, materials, energy, and the use of facilities and equipment such as wind tunnels, factories, offices, and computers—it is convenient to express these values in common terms by using monetary units (such as dollars).

**Effectiveness.** The effectiveness of a system is a quantitative measure of the degree to which the system's purpose is achieved. Effectiveness measures are usually very dependent upon system performance. For example, launch vehicle effectiveness depends on the probability of successfully injecting a payload onto a usable trajectory. The associated system performance attributes include the mass that can be put into a specified nominal orbit, the trade between injected mass and launch velocity, and launch availability.

**Cost Effectiveness.** The cost effectiveness of a system combines both the cost and the effectiveness of the system in the context of its objectives. While it may be necessary to measure either or both of those in terms of several numbers, it is sometimes possible to combine the components into a meaningful, single-valued *objective function* for use in design optimization. Even without knowing how to trade effectiveness for cost, designs that have lower cost and higher effectiveness are always preferred.

A cost-effective and safe system must provide a particular kind of balance between effectiveness and cost: the system must provide the most effectiveness for the resources expended or, equivalently, it must be the least expensive for the effectiveness it provides. This condition is a weak one because there are usually many designs that meet the condition. Think of each possible design as a point in the tradeoff space between effectiveness and cost. A graph plotting the maximum achievable effectiveness of designs available with current technology as a function of cost would, in general, yield a curved line such as the one shown in Figure 2-11. (In the figure, all the dimensions of effectiveness are represented by the ordinate (y axis) and all the dimensions of cost by the abscissa (x axis).) In other words, the curved line represents the envelope of the currently available technology in terms of cost effectiveness.

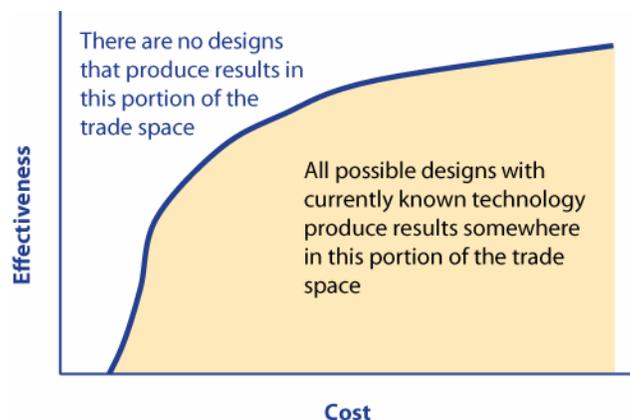
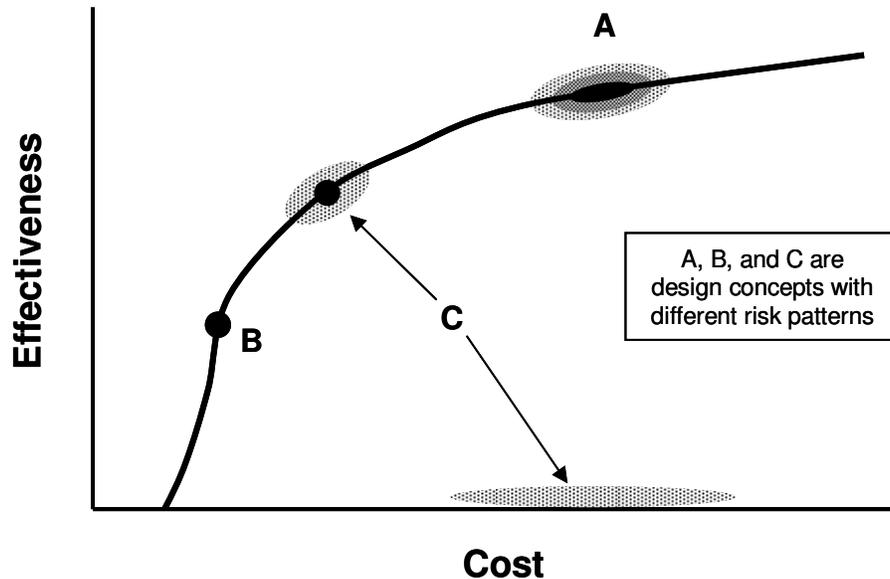


Figure 2-11 The Enveloping Surface of Non-Dominated Designs

Points above the line cannot be achieved with currently available technology; that is, they do not represent feasible designs. (Some of those points may be feasible in the future when further technological advances have been made.) Points inside the envelope are feasible, but are said to be dominated by designs whose combined cost and effectiveness lie on the envelope line. Designs represented by points on the envelope are called cost effective (or efficient or non-dominated) solutions.

Design trade studies, an important part of the systems engineering process, often attempt to find designs that provide a better combination of the various dimensions of cost and effectiveness. When the starting point for a design trade study is inside the envelope, there are alternatives that reduce costs without decreasing any aspect of effectiveness or increase some aspect of effectiveness without decreasing others and without increasing costs. Then, the systems engineer's decision is easy. Other than in the sizing of subsystems, such "win-win" design trades are uncommon, but by no means rare. When the alternatives in a design trade study require trading cost for effectiveness, or even one dimension of effectiveness for another at the same cost, the decisions become harder.

The process of finding the most cost-effective design is further complicated by uncertainty, which is shown in Figure 2-12. Exactly what outcomes will be realized by a particular system design cannot be known in advance with certainty, so the projected cost and effectiveness of a design are better described by a probability distribution than by a point. This distribution can be thought of as a cloud that is thickest at the most likely value and thinnest farthest away from the most likely point, as is shown for design concept A in the figure. Distributions resulting from designs that have little uncertainty are dense and highly compact, as is shown for concept B. Distributions associated with risky designs may have significant probabilities of producing highly undesirable outcomes, as is suggested by the presence of an additional low effectiveness/high cost cloud for concept C. (Of course, the envelope of such clouds cannot be a sharp line such as is shown in the figures, but must itself be rather fuzzy. The line can now be thought of as representing the envelope at some fixed confidence level, that is, a specific, numerical probability of achieving that effectiveness.)



**Figure 2-12 Estimates of Outcomes to be Obtained from Several Design Concepts Including Uncertainty**

Both effectiveness and cost may require several descriptors. Even the Echo balloons (circa 1960), in addition to their primary mission as communications satellites, obtained scientific data on the electromagnetic environment and atmospheric drag. Furthermore, Echo was the first satellite visible to the naked eye, an unquantified—but not unrecognized at the beginning of the space race—aspect of its effectiveness. Costs, the expenditure of limited resources, may be measured in the several dimensions of funding, personnel, use of facilities, and so on. Schedule may appear as an attribute of effectiveness or cost, or as a constraint. Sputnik (circa 1957), for example, drew much of its effectiveness from the fact that it was a “first”—a mission to Mars that misses its launch window has to wait about two years for another opportunity—a clear schedule constraint. Risk results from uncertainties in realized effectiveness, costs, timeliness, and budgets. Sometimes, the systems that provide the highest ratio of effectiveness to cost are the most desirable. However, this ratio is likely to be meaningless or—worse—misleading. To be useful and meaningful, that ratio must be uniquely determined and independent of the system cost. Further, there must be but a single measure of effectiveness and a single measure of cost. If the numerical values of those metrics are obscured by probability distributions, the ratios become uncertain as well. Then any usefulness the simple, single ratio of two numbers might have had disappears.

### The Systems Engineer's Dilemma

At each cost-effective solution:

- To reduce cost at constant risk, performance must be reduced.
- To reduce risk at constant cost, performance must be reduced.
- To reduce cost at constant performance, higher risks must be accepted.
- To reduce risk at constant performance, higher costs must be accepted.

In this context, time in the schedule is often a critical resource, so that *schedule* behaves like a kind of *cost*.

In some contexts, it is appropriate to seek the most effectiveness possible within a fixed budget; in other contexts, it is more appropriate to seek the least cost possible with specified effectiveness. In these cases, there is the question of what level of effectiveness to specify or what level of costs to fix. In practice, these may be mandated in the form of performance or cost requirements. It then becomes appropriate to ask whether a slight relaxation of requirements could produce a significantly cheaper system or whether a few more resources could produce a significantly more effective system.

Usually, the systems engineer must choose among designs that differ in terms of numerous attributes. A variety of methods have been developed that can be used to help managers uncover their preferences between attributes and to quantify their subjective assessments of relative value. When this can be done, trades between attributes can be assessed quantitatively. Often, however, the attributes seem to be truly incommensurate; managers must make their decisions in spite of this multiplicity.

### 3.0 NASA Program/Project Life Cycle

One of the fundamental concepts used within NASA for the management of major systems is the program/project life cycle, which consists of a categorization of everything that should be done to accomplish a program or project into distinct phases, separated by key decision points (KDPs). Phase boundaries are defined so that they provide more-or-less natural points for go/no-go decisions. Decisions to proceed may be qualified by liens that must be removed within a reasonable time. A program or project that fails to pass a KDP may be allowed to “go back to the drawing board” to try again later—or it may be terminated.

All systems start with the recognition of a need or the discovery of an opportunity and proceed through various stages of development to a final disposition. While the most dramatic impacts of the analysis and optimization activities associated with systems engineering are obtained in the early stages, decisions that affect millions of dollars of value or cost continue to be amenable to the systems approach even as the end of the system lifetime approaches.

Decomposing the program/project life cycle into phases organizes the entire process into more manageable pieces. The program/project life cycle should provide managers with incremental visibility into the progress being made at points in time that fit with the management and budgetary environments.

NASA Procedural Requirements (NPR) 7120.5, *NASA Space Flight Program and Project Management Requirements*, defines the major NASA life-cycle phases as Formulation and Implementation. For FS&GS projects, the NASA life-cycle phases of Formulation and Implementation divide into the following seven incremental pieces. The subphases of the project life cycle are:

- Pre-Phase A—Concept Studies ( i.e., identify feasible alternatives)
- Phase A—Concept and Technology Development (i.e., define the project and identify and initiate necessary technology)
- Phase B—Preliminary Design and Technology Completion (i.e., establish a preliminary design and develop necessary technology)
- Phase C—Final Design and Fabrication (i.e., complete the system design and build/code the components)
- Phase D — System Assembly, Integration and Test, Launch (i.e., integrate components, and verify the system, prepare for operations, and launch)
- Phase E—Operations and Sustainment (i.e., operate and maintain the system)
- Phase F—Closeout (i.e., disposal of systems and analysis of data)

Figure 3-1 (Program Life Cycle) and Figure 3-2 (Project Life Cycle) identify the KDPs and reviews that characterize the phases. Sections 3.1 and 3.2 contain narrative descriptions of the

purposes, major activities, products, and KDPs of the NASA program life-cycle phases. Sections 3.3 to 3.9 contain narrative descriptions of the purposes, major activities, products, and KDPs of the NASA project life-cycle phases. Section 3.10 describes the NASA budget cycle within which program/project managers and systems engineers must operate.

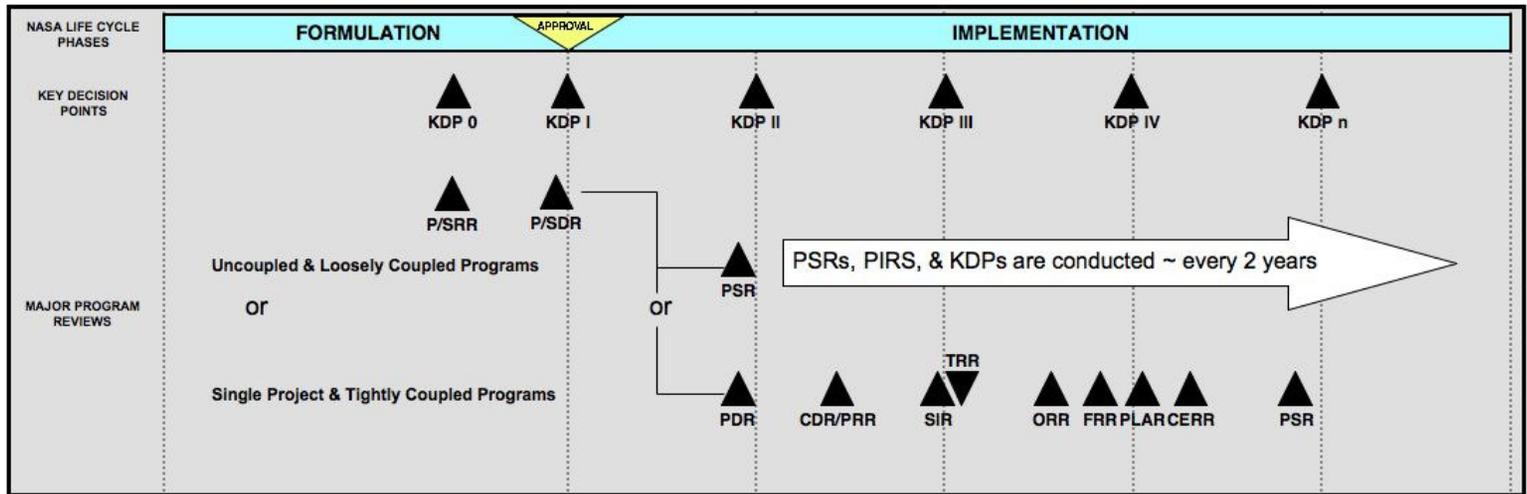


Figure 3-1 NASA Program Life Cycle

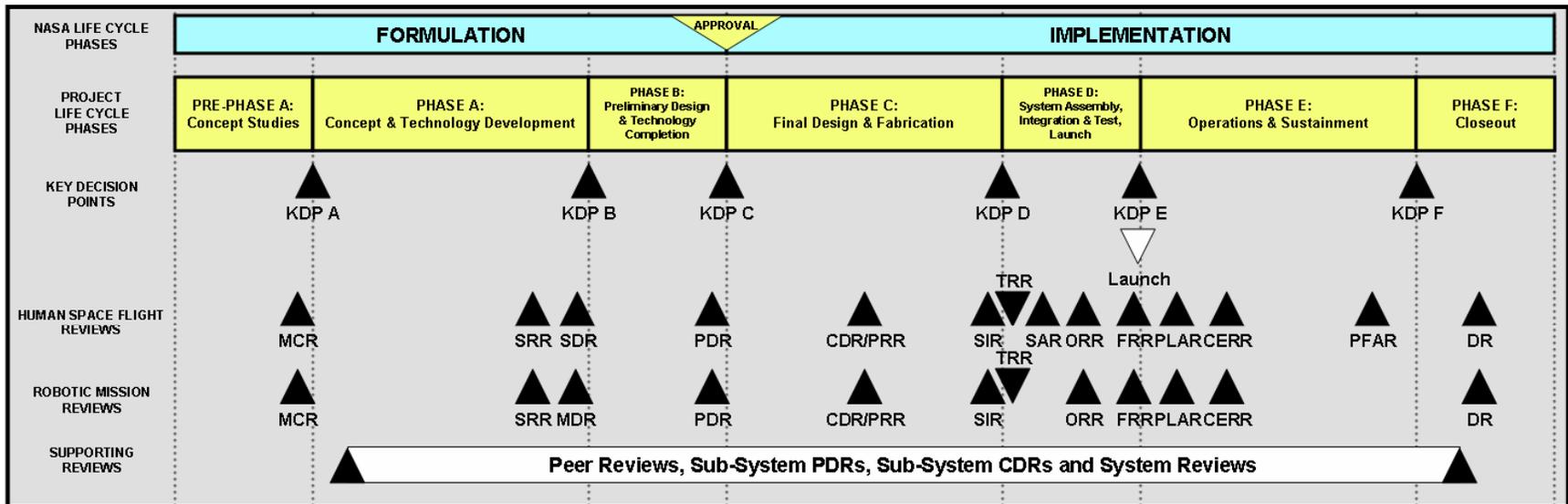


Figure 3-2 NASA Project Life Cycle

### 3.1 Program Formulation

The program Formulation phase establishes a cost-effective program that is demonstrably capable of meeting Agency and mission directorate goals and objectives. The program Formulation Authorization Document (FAD) authorizes a program manager to initiate the planning of a new program and to perform the analyses required to formulate a sound program plan. Major reviews leading to approval at KDP I are the Program/System Requirements Review (P/SRR), Program/System Definition Review (P/SDR), Program Approval Review (PAR), and governing PMC review. A summary of the required gate products for the program Formulation phase can be found in NPR 7120.5. Formulation for all program types is the same, involving one or more program reviews followed by KDP I where a decision is made approving a program to begin implementation. Typically, there is no incentive to move a program into implementation until the first project is ready for implementation.

#### Program Formulation

**Purpose:** To establish a cost-effective program that is demonstrably capable of meeting Agency and mission directorate goals and objectives.

**Major Activities and Their Products:**

Program requirements are developed and allocated to initial projects

Program acquisition strategies are defined and approved

Interfaces to other programs are developed

Initial development of technologies that cut across multiple projects within the program

Initial cost estimates are derived and a program budget is approved

Required Program Formulation Technical Activities of NPR 7120.5

Program Formulation Reviews Entrance/Success Criteria of NPR 7123.1

**Reviews:**

Program/System Requirements Review (P/SRR)

Program/System Definition Review (P/SDR)

### 3.2 Program Implementation

During the program Implementation phase, the program manager works with the mission directorate associate administrator (MDAA) and the constituent projects to execute the program plan cost effectively. Program reviews ensure that the program continues to contribute to Agency and mission directorate goals and objectives within funding constraints. A summary of the required gate products for the program Implementation phase can be found in NPR 7120.5. The program life cycle has two different implementation paths, depending on program type. Each implementation path has different types of major reviews. For uncoupled and loosely coupled programs, the Implementation phase only requires Program Status Reviews (PSRs) and Program Implementation Reviews (PIRs) to assess the program's performance and authorize its continuation at biennial key decision points (KDPs). Single-project and tightly coupled programs are more complex. For single-project programs, the Implementation phase program reviews shown in Figure 3-1 are synonymous (not duplicative) with the project reviews in the project life

cycle (see Figure 3-2) through Phase D. Once in operations, these programs usually have biennial KDPs preceded by attendant PSRs/PIRs. Tightly coupled programs during implementation have program reviews tied to the project reviews to ensure the proper integration of projects into the larger system. Once in operations, tightly coupled programs also have biennial PSRs/PIRs/KDPs to assess the program's performance and authorize its continuation.

### **Program Implementation**

**Purpose:** To execute the program and constituent projects and ensure the program continues to contribute to Agency goals and objectives within funding constraints.

**Major Activities and Their Products:**

Projects are initiated through direct assignment or competitive process (e.g., RFP, Announcement of Opportunity (AO))

Projects formulation, approval, implementation, integration, operation, and ultimate decommissioning are constantly monitored

Program is adjusted as resources and requirements change

Required Program Implementation Technical Activities of NPR 7120.5

Program Implementation Reviews Entrance/Success Criteria of NPR 7123.1

**Reviews:**

Program Status Reviews (PSRs)/Program Implementation Reviews (PIRs) (uncoupled and loosely coupled programs only)

Reviews synonymous (not duplicative) with the project reviews in the project life cycle (see Figure 3-2) through Phase D (single project and tightly coupled programs only)

### **3.3 Project Pre-Phase A—Concept Studies**

The purpose of this activity, which is usually performed more or less continually by concept study groups, is to devise various feasible concepts from which new projects (programs) can be selected. Typically, this activity consists of loosely structured examinations of new ideas, usually without central control and mostly oriented toward small studies. Its major product is a list of suggested projects, based on the identification of needs and the discovery of opportunities that are potentially consistent with NASA's mission, capabilities, priorities, and resources.

### Pre-Phase A—Concept Studies

**Purpose:** To produce a broad spectrum of ideas and alternatives for missions from which new programs/projects can be selected.

**Major Activities and Products: (NOTE: Announcement of Opportunity (AO) Projects will have defined the deliverable products)**

Identify missions consistent with charter

Identify and involve users

Identify and perform trades and analyses

Identify requirements, which include:

- mission
- science
- top level system

Define Measures of Effectiveness (MOEs) and Measures of Performance (MOPs)

Identify top level Technical Performance Measures (TPMs)

Perform preliminary evaluations of possible missions

Prepare program/project proposals, which include:

- Mission justification and objectives
- Possible ConOps

- High-level work breakdown structures (WBSs)

- Cost, schedule, and risk estimates.

- Technology assessment and maturation strategies

Prepare Preliminary Mission Concept Report

Required Pre-Phase A Technical Activities of NPR 7120.5

Mission Concept Review Entrance/Success Criteria of NPR 7123.1

**Reviews:**

Mission Concept Review (MCR)

Informal proposal reviews

Advanced studies may extend for several years, and may be a sequence of papers that are only loosely connected. These studies typically focus on establishing mission goals and formulating top-level system requirements and ConOps. Conceptual designs are often offered to demonstrate feasibility and support programmatic estimates. The emphasis is on establishing feasibility and desirability rather than optimality. Analyses and designs are accordingly limited in both depth and number of options.

### 3.4 Project Phase A—Concept and Technology Development

During Phase A, activities are performed to fully develop a baseline mission concept and begin or assume responsibility for the development of needed technologies. This work, along with interactions with stakeholders, helps with the baselining (Ref. section 6.5 on Configuration Management) of a mission concept and the program requirements on the project.

## Phase A—Concept and Technology Development

**Purpose:** To determine the feasibility and desirability of a suggested new major system and establish an initial baseline compatibility with NASA's strategic plans.

### **Major Activities and Their Products:**

Develop top-level requirements and constraints  
Define and document system requirements (hardware and software)  
Preliminary system requirements allocation to next lower level  
Define system software functionality description and requirements  
Develop corresponding evaluation criteria/metrics  
Document the concept of operations  
Baseline Mission Concept Report  
Demonstrate that credible, feasible design(s) exist  
Define and document internal and external interface requirements  
Perform and archive trade studies  
Establish technical resource estimates  
Define verification and validation approach and document in Verification and Validation Plan  
Acquire systems engineering tools and models  
Initiate environmental impact studies  
Baseline the Systems Engineering Management Plan (SEMP)  
Prepare and initiate a Risk Management Plan  
Prepare and Initiate a Configuration Management Plan  
Prepare and initiate a Data Management Plan  
Prepare engineering specialty plans (e.g., Contamination Control Plan, EMI/EMC Control Plan, Reliability Plan, Quality Control Plan)  
Define life-cycle cost estimates and develop system-level cost-effectiveness model  
Define the work breakdown structure (WBS)  
Develop statement(s) of work  
Prepare Safety and Mission Assurance Plan  
Prepare Software Development Plan  
Prepare Technology Development Plan and initiate advanced technology developments  
Identify integrated logistics support requirements  
Develop initial orbital debris assessment (NASA Safety Standard 1740.14)  
Establish Human Rating Plan  
Required Phase A Technical Activities of NPR 7120.5  
Phase A Reviews Entrance/Success Criteria of NPR 7123.1

### **Reviews:**

System Requirements Review (SRR)  
Mission Definition Review (MDR) (Robotic Mission Review only)  
System Definition Review (SDR) (Human Space Flight Review only)  
  
Lower-level SRRs  
Lower-level MDRs and SDRs

In Phase A, a team, often associated with a program or informal project office, readdresses the mission concept to ensure that the project justification and practicality are sufficient to warrant a place in NASA's budget. The team's effort focuses on analyzing mission requirements and establishing a mission architecture. Activities become formal, and the emphasis shifts toward establishing optimality rather than feasibility. The effort addresses more depth and considers many alternatives. Goals and objectives are solidified, and the project develops more definition in the system requirements, top-level system architecture, and ConOps. Conceptual designs are developed and exhibit more engineering detail than in advanced studies. Technical risks are identified in more detail and technology development needs become focused.

In Phase A, the effort focuses on allocating functions to particular items of hardware, software, personnel, etc. System functional and performance requirements along with architectures and designs become firm as system trades and subsystem trades iterate back and forth in the effort to seek out more cost-effective designs. (Trade studies should precede—rather than follow—system design decisions. Major products to this point include an accepted “functional” baseline and preliminary “design-to” baseline for the system and its major end items. The effort also produces various engineering and management plans to prepare for managing the project’s downstream processes, such as verification and operations, and for implementing engineering specialty programs.

### **3.5 Project Phase B—Preliminary Design and Technology Completion**

During Phase B, activities are performed to establish an initial project baseline, which (according to NPR 7120.5 and NPR 7123.1) includes “a formal flowdown of the project-level performance requirements to a complete set of system and subsystem design specifications for both flight and ground elements” and “corresponding preliminary designs.” The technical requirements should be sufficiently detailed to establish firm schedule and cost estimates for the project. It also should be noted, especially for AO driven projects, that Phase B is where the top level requirements, and the next level flowdown set of requirements are finalized, verified, and placed under configuration control. While they should be baselined in Phase A, there are just enough changes resulting from the trade studies and analyses in late Phase A and early Phase B that changes are inevitable. However, by mid-Phase B, the top level requirements should be finalized.

Actually, “the” Phase B baseline consists of a collection of evolving baselines covering technical and business aspects of the project: system (and subsystem) requirements and specifications, designs, verification and operations plans, and so on in the technical portion of the baseline, and schedules, cost projections, and management plans in the business portion. Establishment of baselines implies the implementation of configuration management procedures. (See section 6.5.)

## Phase B—Preliminary Design and Technology Completion

**Purpose:** To define the project in enough detail to establish an initial baseline capable of meeting mission needs.

### Major Activities and Their Products:

Review and update documents developed and baselined in Phase A  
Finalized and approve top level requirements and first level flowdown to the next level requirements specifications  
Perform and archive trade studies results  
Perform Design Analyses and report results  
Conduct Engineering Development Tests and report results  
Update engineering specialty plans (e.g., Contamination Control Plan, EMI/EMC Control Plan, Reliability Plan, Quality Control Plan)  
Select a baseline design solution  
Baseline Preliminary Design Report  
Define internal and external interface design solutions (e.g., Interface Control Documents)  
Establish and baseline “design-to” specifications (hardware and software) and drawings, verification and validation plans, and interface documents at lower levels  
Update technology maturation planning Report technology development results  
Update Risk Management Plan  
Update cost and schedule data  
Preliminary Missile System Pre-Launch Safety Package  
Preliminary Orbital Debris Assessment  
Required Phase B Technical Activities of NPR 7120.5  
Phase B Reviews Entrance/Success Criteria of NPR 7123.1

### Reviews:

Preliminary Design Review (PDR)  
Lower-level Preliminary Design Reviews (PDRs)  
Safety review(s)

In Phase B, the effort shifts to establishing a functionally complete preliminary design solution (i.e., a “design-to” baseline) that should meet mission goals and objectives. Trade studies continue. Interfaces among the major end items are defined. Engineering test items may be developed and used to derive data for further design work, and project risks are reduced by successful technology developments and demonstrations. Phase B culminates in a series of preliminary design reviews (PDRs), containing the system-level PDR and PDRs for lower-level end items as appropriate. The PDRs reflect the successive refinement of requirements into designs. (See the doctrine of successive refinement in section 4.4.1.2 of Design Solution and Figure 4-17.) Design issues uncovered in the PDRs should be resolved so that final design can begin with unambiguous “design-to” specifications. From this point on, almost all changes to the baseline are expected to represent successive refinements, not fundamental changes. Prior to baselining, the system architecture, preliminary design, and ConOps must have been validated by enough technical analysis and design work to establish a credible, feasible design at a lower level of detail than was sufficient for Phase A.

### 3.6 Project Phase C—Final Design and Fabrication

During Phase C, activities are performed to establish a complete design (“build-to” baseline) and fabricate/produce (hardware) and code (software) in preparation for integration. Trade studies continue. Engineering test units more closely resembling actual hardware are built and tested so

as to establish confidence that the design will function in the expected environments. Engineering specialty analysis results are integrated into the design, and the manufacturing process and controls are defined and validated. All the planning for the testing and operational equipment, processes and analysis, integration of the engineering specialty analysis, and manufacturing processes and controls are initiated back in Phase A. Configuration management continues to track and control design changes as detailed interfaces are defined. At each step in the successive refinement of the final design, corresponding integration and verification activities are planned in greater detail. During this phase, technical parameters, schedules, and budgets are closely tracked to ensure that undesirable trends (such as an unexpected growth in spacecraft mass or increase in its cost) are recognized early enough to take corrective action. These activities focus on preparing for the Critical Design Review (CDR), Production Readiness Review (PRR) (*if required*) and the System Integration Review (SIR).

Phase C contains a series of critical design reviews (CDRs) containing the system-level CDR and CDRs corresponding to the different levels of the system hierarchy. A CDR for each end item should be held prior to the start of fabrication/production for hardware and prior to the start of coding of deliverable software products. Typically, the sequence of CDRs reflects the integration process that will occur in the next phase— that is, from lower-level CDRs to the system-level CDR. Projects, however, should tailor the sequencing of the reviews to meet their individual needs. If there is a production run of products, a Production Readiness Review (PRR) will be performed to ensure the production plans, facilities, and personnel are ready to begin production. Phase C culminates with a System Integration Review (SIR). The final product of this phase is a product ready for integration.

### Phase C—Final Design and Fabrication

**Purpose:** To complete the detailed design of the system (and its associated subsystems, including its operations systems) and fabricate (hardware) and code (software).

**Major Activities and Their Products:**

Updates to documentation developed and baselined in Phase B  
Add remaining lower-level design specifications to the system architecture  
Maintain requirements documents  
Maintain verification and validation plans  
Develop Verification and Validation Procedures  
Update interface documents  
Develop hardware and software detailed designs  
Establish and baseline “build-to” specifications (hardware and software) and drawings, verification and validation plans, and interface documents at all levels  
Augment baselined documents to reflect the growing maturity of the system: system architecture, work breakdown structure, project plans  
Monitor project progress against project plans  
Develop the system integration plan and the system operation plan  
Perform and archive trade studies  
Complete manufacturing and assembly plans and procedures  
Update and baseline production plans  
Develop the end-to-end information system design  
Baseline Detailed Design Report  
Baseline logistics support plan  
Identify opportunities for pre-planned product improvement  
Develop spares planning  
Fabricate (or code) the product  
Refine Integration Procedures  
Baseline Missile System Pre-Launch Safety Package  
Develop Command and Telemetry List  
Prepare Launch Site Checkout and Operations Plans  
Prepare Operations and Activation Plan  
Preliminary Operations Handbook  
Baseline Orbital Debris Assessment  
Prepare System Decommissioning/Disposal Plan for later use in Phase F including human capital transition  
Required Phase C Technical Activities of NPR 7120.5  
Phase C Reviews Entrance/Success Criteria of NPR 7123.1

**Reviews:**

Critical Design Review (CDR)  
Lower-level Critical Design Reviews (CDRs)  
Production Readiness Review(s)  
System Integration Review(s)  
Lower-level Integration Reviews

### 3.7 Project Phase D—System Assembly, Integration and Test, Launch

During Phase D, activities are performed to assemble, integrate, test, and launch the system. These activities focus on preparing for the Flight Readiness Review (FRR). Activities include, assembly, integration, and verification of the system, including testing the flight system to expected environments, with margin. Other activities include the initial training of operating personnel and implementation of the logistics and spares planning. For flight projects, the focus

of activities then shifts to pre-launch integration and launch. Although all these activities are conducted in this phase of a project, the planning for these activities are initiated back in Phase A. The planning for the activities cannot be delayed until Phase D begins because the design of the project is too advanced to incorporate requirements for testing and operations. Phase D concludes with a system that has been shown to be capable of accomplishing the purpose for which it was created.

#### **Phase D—System Assembly, Integration and Test, Launch**

**Purpose:** To assemble and integrate the products to create the system, meanwhile developing confidence that it will be able to meet the system requirements. Launch and prepare for operations.

##### **Major Activities and Their Products:**

Integrate those items according to the integration plan and perform verifications, yielding verified components and subsystems

Refine verification and validation procedures at all levels

Perform system qualification verification(s)

Perform system acceptance verification(s)

Perform system environmental testing

Monitor project progress against project plans

Assess and approve verification and validation results for compliance

Resolve verification and validation discrepancies

Archive documentation for verifications and validations performed

Baseline Verification and Validation Report

Baseline “as-built” hardware and software documentation

Document Lessons Learned

Prepare and baseline operator’s manuals

Prepare and baseline maintenance manuals

Train initial system operators and maintainers

Train on contingency planning

Finalize and implement spares planning

Approve and baseline Operations Handbook

Update Logistics Support Plan

Confirm telemetry validation and ground data processing

Confirm system and support elements are ready for flight

Integrate with launch vehicle(s) and launch, perform orbit insertion, etc., to achieve a deployed system

Perform initial operational verification(s) and validation(s)

Required Phase D Technical Activities of NPR 7120.5

Phase D Reviews Entrance/Success Criteria of NPR 7123.1

##### **Reviews:**

Test Readiness Reviews (TRRs) (at all levels)

System Acceptance Review (SAR) (Human Space Flight Review only)

Operational Readiness Review (ORR)

Flight Readiness Review(s) (FRRs)

System functional and physical configuration audits

Safety reviews

### **3.8 Project Phase E—Operations and Sustainment**

During Phase E, activities are performed to conduct the prime mission and meet the initially identified need and maintain support for that need. The products of the phase are the results of

the mission. This phase encompasses evolution of the system only insofar as that evolution does not involve major changes to the system architecture; changes of that scope constitute new “needs,” and the project life cycle starts over. For large flight projects, there may be an extended period of cruise, orbit insertion, on-orbit assembly, and initial shake-down operations. Near the end of the prime mission, the project may apply for a mission extension to continue mission activities or attempt to perform additional mission objectives.

### **Phase E—Operations and Sustainment**

**Purpose:** To conduct the mission and meet the initially identified need and maintain support for that need.

**Major Activities and Their Products:**

- Conduct launch vehicle performance assessment
- Train replacement operators and maintainers
- Train the flight team for future mission phases (e.g., planetary landed operations)
- Conduct in-orbit spacecraft checkout
- Commission and activate science instruments
- Conduct the intended prime mission(s)
- Collecting Engineering and Science Data
- Maintain and approve operations and maintenance logs
- Address problem/failure reports
- Maintain and upgrade the system
- Apply for mission extensions and conduct if awarded
- Prepare for deactivation, disassembly, decommissioning as planned
- Complete post flight evaluation reports
- Final Mission Report
- Required Phase E Technical Activities of NPR 7120.5
- Phase E Reviews Entrance/Success Criteria of NPR 7123.1

**Reviews:**

- Post Launch Assessment Reviews (PLARs)
- Critical Event Readiness Reviews (CERRs)
- Post Flight Assessment Reviews (PFARs) (Human Space Flight Review only)
- System upgrade reviews
- Safety reviews

### **3.9 Project Phase F—Closeout**

During Phase F, activities are performed to implement the systems decommissioning disposal planning and analyze any returned data and samples. The products of the phase are the results of the mission.

## Phase F —Closeout

**Purpose:** To implement the Systems Decommissioning/Disposal Plan developed in Phase E, and perform analyses of the returned data and any returned samples

### Major Activities and Their Products:

Dispose of the system and supporting processes  
Document Lessons Learned  
Process and analyze mission data  
Baseline Mission Final Report  
Archive Data  
Begin transition of human capital (if applicable)  
Required Phase F Technical Activities of NPR 7120.5  
Phase F Reviews Entrance/Success Criteria of NPR 7123.1

### Reviews:

Decommissioning Review (DR)

Phase F deals with the final closeout of the system when it has completed its mission; the time at which this occurs depends on many factors. For a flight system with a short mission duration, which returns to Earth, closeout may require little more than de-integration of the hardware and its return to its owner. On flight projects of long duration, closeout may proceed according to established plans, or may begin as a result of unplanned events, such as failures. Refer to *NPD 8010.3, Notification of Intent to Decommission or Terminate Operating Space Systems and Terminate Missions*, for terminating and operating mission. Alternatively, technological advances may make it uneconomic to continue operating the system either in its current configuration or an improved one.

To limit space debris there are guidelines for removing earth-orbiting robotic satellites from their operational orbits at the end of their useful life. [See *NPD 8710.3B, NASA Policy for Limiting Orbital Debris Generation*.] For low Earth orbiting (LEO) missions, the satellite is usually de-orbited. For small satellites, this is accomplished by allowing the orbit to slowly decay until the satellite eventually burns up in the earth's atmosphere. Larger, more massive satellites and observatories must be designed to demise or de-orbited in a controlled manner so that they can be safely targeted for impact in a remote area of the ocean. The geostationary (GEO) satellites at 35,790 km above the earth cannot be practically de-orbited, so they are boosted to a higher orbit well beyond the crowded operational GEO orbit.

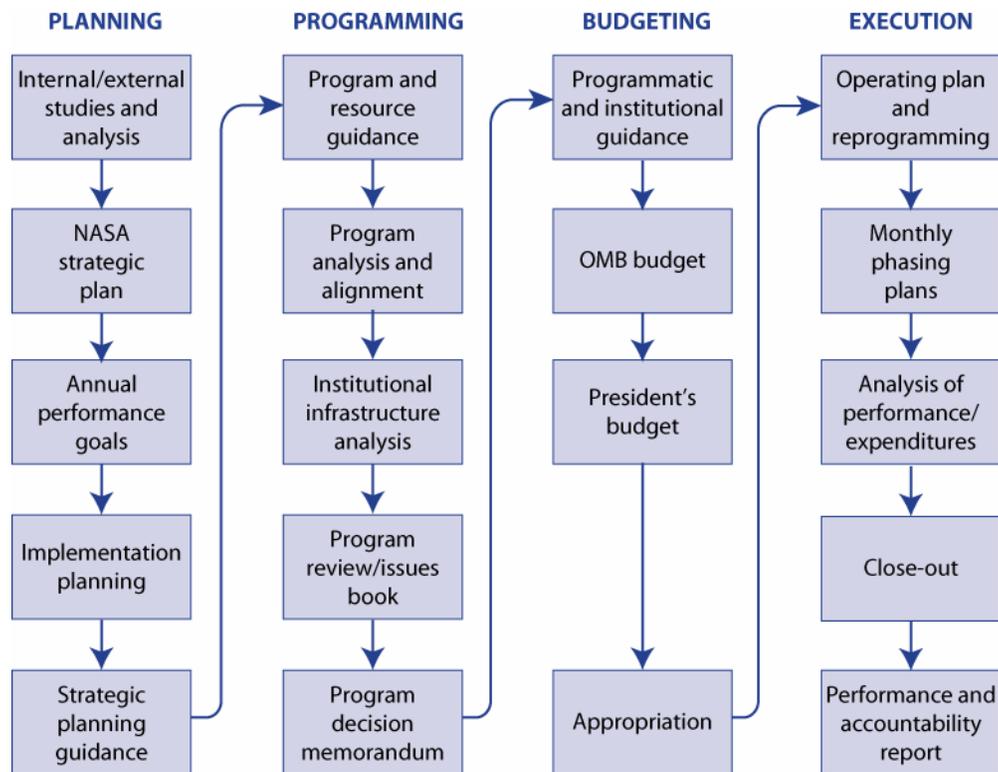
In addition to uncertainty as to when this part of the phase begins, the activities associated with safe closeout of a system may be long and complex and may affect the system design.

Consequently, different options and strategies should be considered during the project's earlier phases along with the costs and risks associated with the different options.

### 3.10 Funding: The Budget Cycle

NASA operates with annual funding from Congress. This funding results, however, from a continuous rolling process of budget formulation, budget enactment, and finally, budget execution. NASA's Financial Management Requirements (FMR) Volume 4 provides the concepts, the goals, and an overview of NASA's budget system of resource alignment referred to

as Planning, Programming, Budgeting, and Execution (PPBE) and establishes guidance on the Programming and Budgeting Phases of the PPBE process, which are critical to budget formulation for NASA. The volume includes strategic budget planning and resources guidance, program review, budget development, budget presentation, and justification of estimates to the Office of Management and Budget (OMB) and to Congress. It also provides detailed descriptions of the roles and responsibilities for key players in each step of the process. It consolidates current legal, regulatory, and administrative policies and procedures applicable to NASA. A highly simplified representation of the typical NASA budget cycle is shown in Figure 3-3.



**Figure 3-3 Typical NASA Budget Cycle**

NASA typically starts developing its budget each February with economic forecasts and general guidelines as identified in the most recent President’s Budget. By late August, NASA has completed the planning, programming, and budgeting phase of the PPBE process and prepares for submittal of a preliminary NASA budget to the OMB. A final NASA budget is submitted to the OMB in September for incorporation into the President’s budget transmittal to Congress, which generally occurs in January. This proposed budget is then subjected to congressional review and approval, culminating in the passage of bills authorizing NASA to obligate funds in accordance with congressional stipulations and appropriating those funds. The congressional process generally lasts through the summer. In recent years, however, final bills have often been delayed past the start of the fiscal year on October 1. In those years, NASA has operated on continuing resolutions (CR) by Congress.

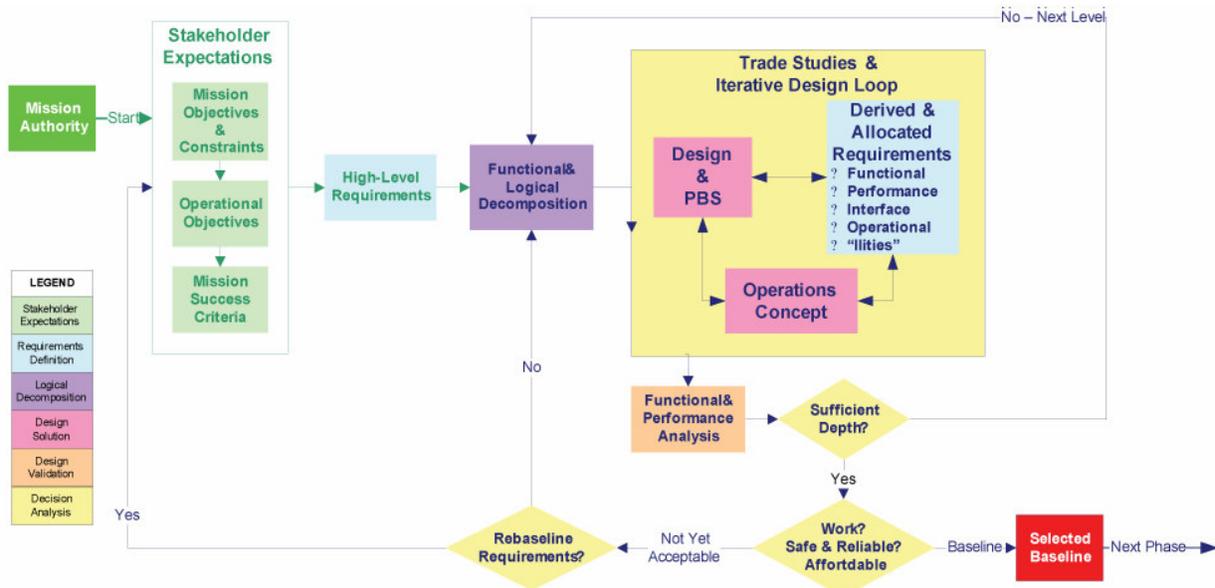
With annual funding, there is an implicit funding control gate at the beginning of every fiscal year. While these gates place planning requirements on the project and can make significant

replanning necessary, they are not part of an orderly systems engineering process. Rather, they constitute one of the sources of uncertainty that affect project risks, and they are essential to consider in project planning.

## 4.0 System Design

The system design processes are four interdependent, highly iterative and recursive processes, resulting in a validated set of requirements and a validated design solution that satisfies a set of stakeholder expectations. The four system design processes are to develop stakeholder expectations, technical requirements, logical decompositions and design solutions.

Figure 4-1 illustrates the recursive relationship between the four system design processes. These processes start with the study team defining the stakeholder expectations, including the mission objectives, constraints, design drivers, operational objectives, and the criteria for defining mission success. This set of stakeholder expectations and high-level requirements are used to drive an iterative design loop where a strawman architecture/design, concept of operations (ConOps), and the derived requirements are developed. These three products must be consistent with each other and will require iterations and design decisions to achieve this consistency. Once consistency is achieved, analyses allow the project team to validate the design against the Stakeholder Expectations. A simplified validation asks the questions: Does the system work? Is the system safe and reliable? Is the system affordable? If the answer to any of these questions is no, then changes to the design or stakeholder expectations will be required and the process started again. This process continues until the system—architecture, ConOps, and requirements—meets the stakeholder expectations.



**Figure 4-1 Interrelationships between the System Design Processes**

The depth of the design effort must be sufficient to meet several needs. The design must penetrate sufficiently to allow analytical validation of the design to the requirements. It must be a feasible and credible design when judged by a knowledgeable independent review team and must have sufficient depth to support cost modeling.

Once the system meets the stakeholder expectations, the study team baselines the products and prepares for the next phase. Often intermediate levels of decomposition are validated as part of the process. In the next level of decomposition, the baselined derived (and allocated) requirements become the set of high level requirements for the decomposed elements and the process begins again. These system design processes are primarily applied in Pre-Phase A and continue through Phase C.

The system design processes during Pre-Phase A focus on producing a feasible design that will lead to formulation approval. During Phase A, alternative designs and additional analytical maturity are pursued to optimize the design architecture. Phase B results in a preliminary design that satisfies the approval criteria. During Phase C, detailed, build-to designs are completed.

This is a simplified description intended to demonstrate the recursive relationship between the system design processes. These processes should be used as guidance and tailored for each study team depending on the size of the project and the hierarchical level of the study team. The next sections describe each of the four system design processes and their associated products for a given NASA mission.

#### **System Design Keys**

- Successfully understanding and defining the objectives of the mission and operational concepts are keys to capturing the stakeholders expectations which will translate into quality requirements over the life cycle of the project.
- Complete and thorough requirements traceability is a critical factor in successful validation of requirements.
- Clear and unambiguous requirements will help avoid misunderstanding when developing the overall system and when making major or minor changes.
- If all decisions made during the development of the original design concept are documented in the technical data package, the original design philosophy and negotiation results will be available to make it easier to assess future required changes and modifications to the original design concept.
- The differences between the design solution verification and the design solution validation have to be understood for the solution to be successful. The design solution verification is in contrast to the verification of the end product described in the end product verification plan which is part of the technical data package. The design solution verification occurs when an acceptable design solution has been selected and documented in a technical data package. The design solution is verified against the system requirements and constraints. However, the validation of a design solution is a continuing recursive and iterative process during which the design solution is evaluated against stakeholder expectations.

## 4.1 Stakeholder Expectations Definition

The Stakeholder Expectations Definition Process is the initial process within the SE engine that establishes the foundation from which the system is designed and the product realized. The main purpose of this process is to identify who the stakeholders are and how they intend to use the product. This is usually accomplished through use case scenarios, Design Reference Missions (DMRs), and operational concepts.

### 4.1.1 Process Description

Figure 4-2 provides a typical flow diagram for the Stakeholder Expectations Definition Process and identifies typical inputs, outputs, and activities to consider in addressing stakeholder expectations definition.

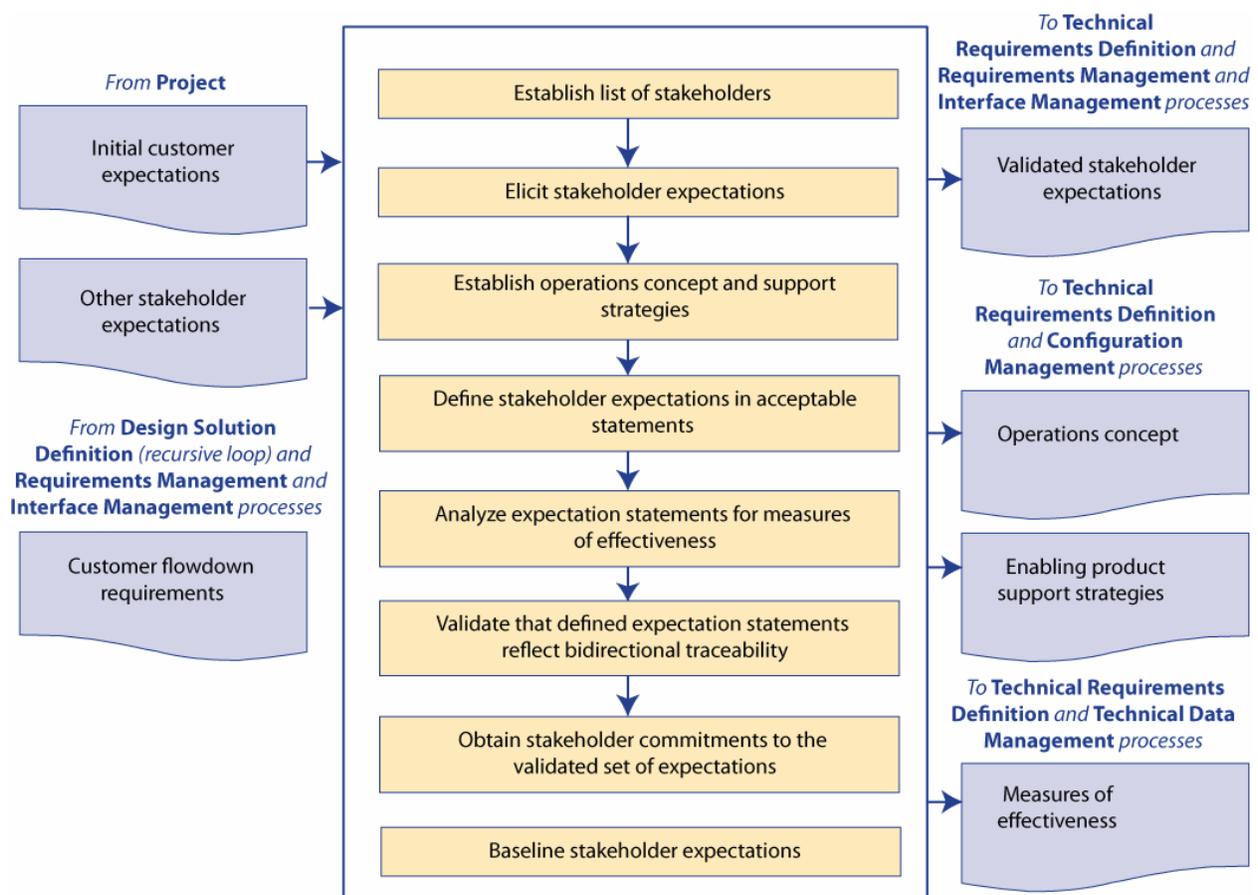


Figure 4-2 Stakeholder Expectations Definition Process

#### 4.1.1.1 Inputs

Typical inputs needed for the stakeholder expectations process would include the following:

- **Upper Level Requirements and Expectations:** These would be the requirements and expectations (e.g., needs, wants, desires, capabilities, constraints, external interfaces) that are being flowed down to a particular system of interest from a higher level (e.g., program, project, etc.).
- **Identified Customers and Stakeholders:** The organization/individual who has requested the product(s) and those who are affected by or is in some way accountable for the product's outcome.

#### 4.1.1.2 Process Activities

### Identifying Stakeholders

Advocacy for new programs and projects may originate in many organizations. These include Presidential Directives, Congress, NASA Headquarters (HQ), the NASA Centers, NASA Advisory Committees, the National Academy of Sciences, the National Space Council, and many other groups in the science and space communities. These organizations are commonly referred to as stakeholders. A stakeholder is any organization or individual that has a stake in the project, meaning that they have a legitimate interest in the project.

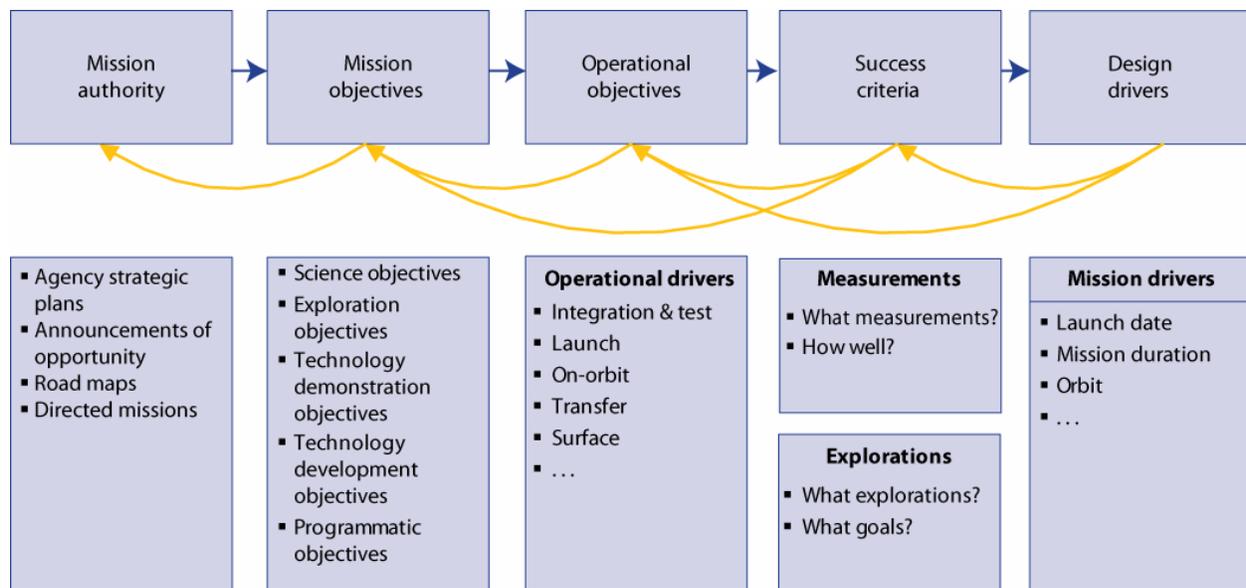
Stakeholders can generally be divided into two classes; customers and other interested parties. Each class sets differing boundaries. In general, other interested parties provide broad, overarching constraints or boundaries within which the customer's needs must be achieved. The customer stakeholders provide direct, tangible product-focused statements that can be targeted for development.

Customer stakeholders are organizations or individuals that request a product and will receive the product. The customer may be an end-user of the product, the acquiring agent for the end user, or the requestor of the work products for a technical effort. Examples of customer stakeholders are scientists, project managers, and subsystems engineers. Other interested parties are groups or individuals that may be affected by the resulting product, the manner in which the product is used, or have a responsibility for providing life-cycle support services. Examples include Congress, advisory planning teams, program managers and NASA contractors. It is important that a list of stakeholders be identified early in the process and that all parties are aware of the primary stakeholders.

### Identifying Stakeholder Expectations

Stakeholder expectations, the vision of a particular stakeholder individual or group, result when they specify what is desired as an end state or item to be produced and put bounds upon the achievement of the goals. These bounds may encompass expenditures (resources), time to deliver, performance objectives, or other less obvious quantities such as organizational needs or geo-political goals.

Figure 4-3 shows the type of information needed when defining stakeholder expectations and depicts how the information evolves into a set of high level requirements. The yellow paths depict validation paths.



**Figure 4-3 Product Flow for Stakeholder Expectations**

Defining stakeholder expectations begins with the mission authority and strategic objectives that the mission is meant to achieve. *Mission authority* changes depending on the category of the mission. For example, science missions are usually driven by NASA Science Mission Directorate strategic plans whereas the exploration missions may be driven by a Presidential Directive.

An early task in defining stakeholder expectation is understanding the *objectives of the mission*. Clearly describing and documenting them helps ensure that the project team is working toward a common goal. These objectives form the basis for developing the mission so they need to be clearly defined and articulated. For science missions, these are usually science objectives and for exploration missions, exploration objectives.

Defining the objectives is done by eliciting the needs, wants, desires, capabilities, external interfaces, and constraints from the stakeholders. Arriving at an agreed-to set of objectives can be a long and arduous task. The proactive iteration with the stakeholders throughout the systems engineering process is the way that all parties can come to a true understanding of what should be done and what it takes to do the job. It is important to know who the primary stakeholders are and who has the decision authority to help resolve conflicts.

The project team should also identify the *constraints* that may apply. A constraint is a condition that must be met. Sometimes a constraint is dictated by external factors such as orbital mechanics or the state of technology; sometimes constraints are the result of the overall budget environment. It is important to document the constraints and assumptions along with the mission objectives.

*Operational objectives* also need to be included in defining the stakeholder expectations. The operational objectives identify how the mission must be operated in order to achieve the mission objectives.

The *mission and operational success criteria* define what the mission must accomplish to be successful. This will be in the form of a Measurement Concept (MC) for science missions, and Exploration Concept (EC) for human exploration missions. The success criteria also define how well the concept measurements or exploration activities must be accomplished. The success criteria capture the stakeholder expectations, and along with programmatic requirements and constraints, are used within the high level requirements.

The *design drivers* will be strongly dependent upon the ConOps, including the operational environment, orbit, and the mission duration requirements. For science missions, the design drivers may include, at a minimum, the mission launch date, duration, and orbit. If alternative orbits are to be considered, a separate concept is needed for each orbit. Exploration missions must consider the destination, the duration, the operational sequence (and system configuration changes), and the in situ exploration activities, which allow the exploration to succeed.

The end result of this step is the discovery and delineation of the system's goals, which generally express the agreements, desires and requirements of the eventual users of the system. The high level requirements and success criteria are examples of the products representing the consensus of the stakeholders.

It is extremely important to involve stakeholders in all phases of a project. Such involvement should be built in as a self-correcting feedback loop that will significantly enhance the chances of mission success. Involving stakeholders in a project builds confidence in the end-product and serves as a validation and acceptance with the target audience.

#### 4.1.1.3 *Outputs*

Typical outputs for capturing stakeholder expectations would include the following:

- **Top-Level Requirements and Expectations:** These would be the top-level requirements and expectations (e.g., needs, wants, desires, capabilities, constraints, external interfaces) for the product(s) to be developed.
- **ConOps:** This describes how the system will be operated during the life-cycle phases in order to meet stakeholder expectations. It describes the system characteristics from an operational perspective and helps facilitate an understanding of the system goals. Examples would be an Operational Concept Document or a Design Reference Mission (DRM).

### 4.1.2 *Stakeholder Expectations Definition Guidance*

#### 4.1.2.1 *Concept of Operations*

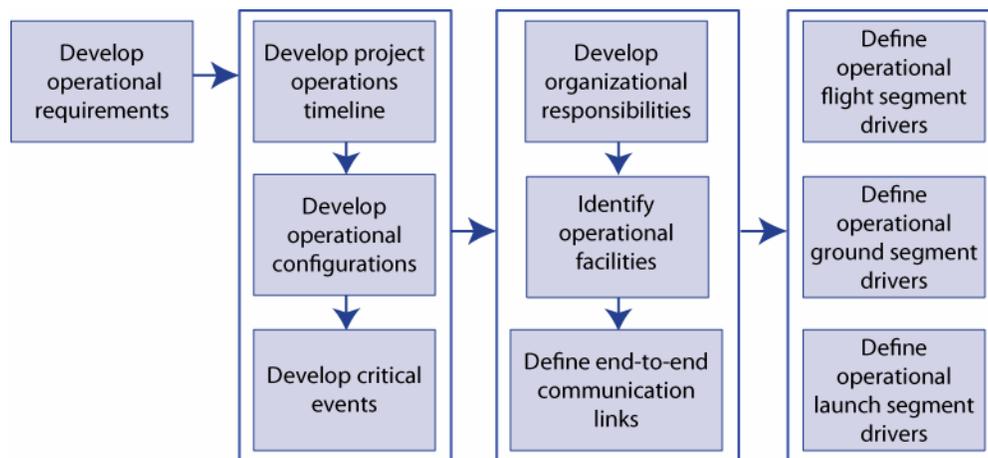
The ConOps is an important component in capturing stakeholder expectations, requirements, and the architecture of a project. The ConOps describes how the system will be operated during the life-cycle phases in order to meet stakeholder expectations. It describes the system characteristics from an operational perspective and helps facilitate an understanding of the system goals. It stimulates the development of the requirements and architecture related to the user elements of the system. It serves as the basis for subsequent definition documents such as the Operations

Plan, Launch and Early Orbit Plan, and Operations Handbook and provides the foundation for the long range operational planning activities such as operational facilities, staffing, and network scheduling.

The ConOps is an important driver in the system requirements and therefore must be considered early in the system design processes. Thinking through the ConOps and use cases often reveals requirements and design functions that might otherwise be overlooked. A simple example to illustrate this point is adding system requirements to allow for communication during a particular phase a mission. This may require an additional antenna in a specific location that may not be required during the nominal mission.

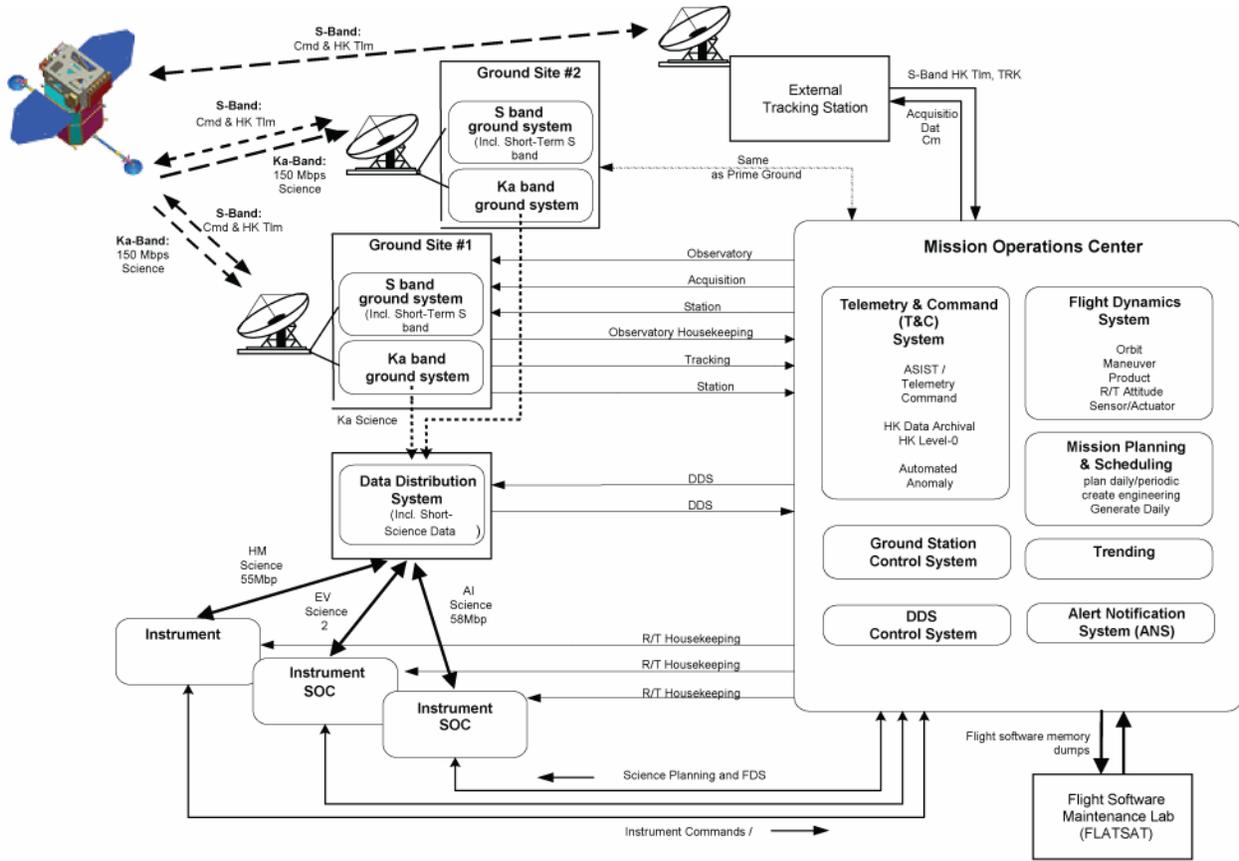
The ConOps is important for all projects. For science projects, the ConOps describes how the systems will be operated to achieve the measurement set required for a successful mission. They are usually driven by the data volume of the measurement set. The ConOps for exploration projects is likely to be more complex. There are typically more operational phases, more configuration changes and additional communication links required for human interaction. The success of these projects are defined by the operations required to achieve the project objectives.

The ConOps should consider all aspects of operations including Integration and Test and launch through disposal. Typical information contained in ConOps includes a description of the major phases, operation timelines, operational scenarios and/or Design Reference Mission (DRM), end-to-end communications strategy, command and data architecture, operational facilities, integrated logistic support (resupply, maintenance, and assembly) and critical events. The operational scenarios describe the dynamic view of the systems operations and includes how the system is perceived to function throughout the various modes and mode transitions, including interactions with external interfaces. For exploration missions, multiple DRMs make up a ConOps. The design and performance analysis leading to the requirements must satisfy all of them. Figure 4-4 illustrates typical information included in the ConOps for a science mission and Figure 4-5 is an example of a end-to-end operational architecture. For more information about ConOps, see ANSI/AIAA G-043-1992, *Guide for the Preparation of Operational Concept Documents*.



**Figure 4-4 Typical ConOps Development for a Science Mission**

The operation timelines provides the basis for defining system configurations, operational activities and other sequenced related elements necessary to achieve the mission objectives for each operational phase. It describes the activities, tasks, and other sequenced related elements necessary to achieve the mission objectives in each of the phases. Depending on the type of project (science, exploration, operational) the timeline could become quite complex.



**Figure 4-5 Example of an Associated End-to-End Operational Architecture**

The timeline matures along with the design. It starts as a simple time sequenced order of the major events and matures into a detailed description of subsystem operations during all major mission modes or transitions. An example of a lunar sortie timeline and design reference mission early in the life cycle are shown in Figure 4-6a and b, respectively. An example of a more detailed, integrated timeline later in the life cycle for a science mission is shown in Figure 4-7.

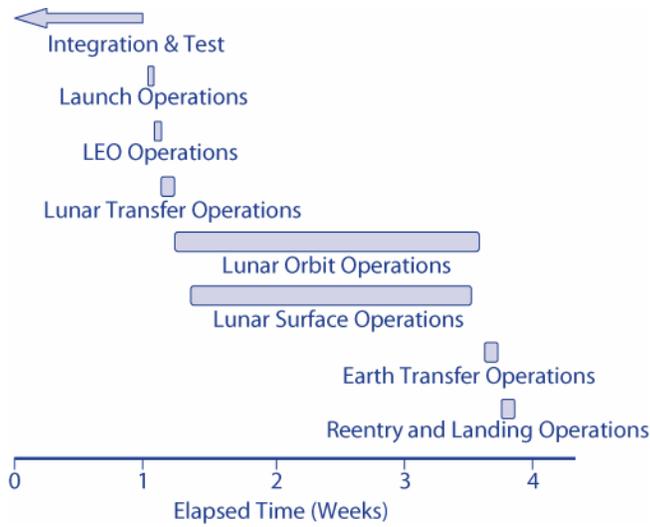


Figure 4-6a. Example of a Lunar Sortie Timeline Developed Early in the Life Cycle

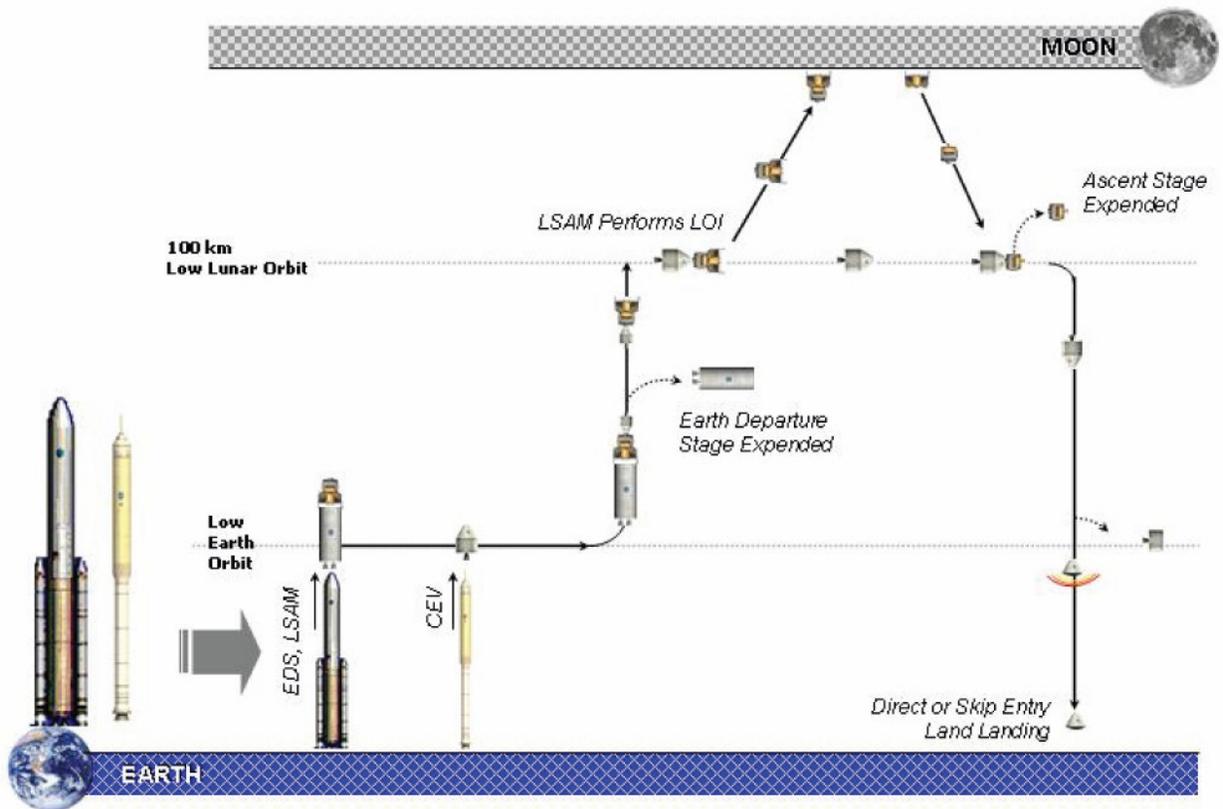
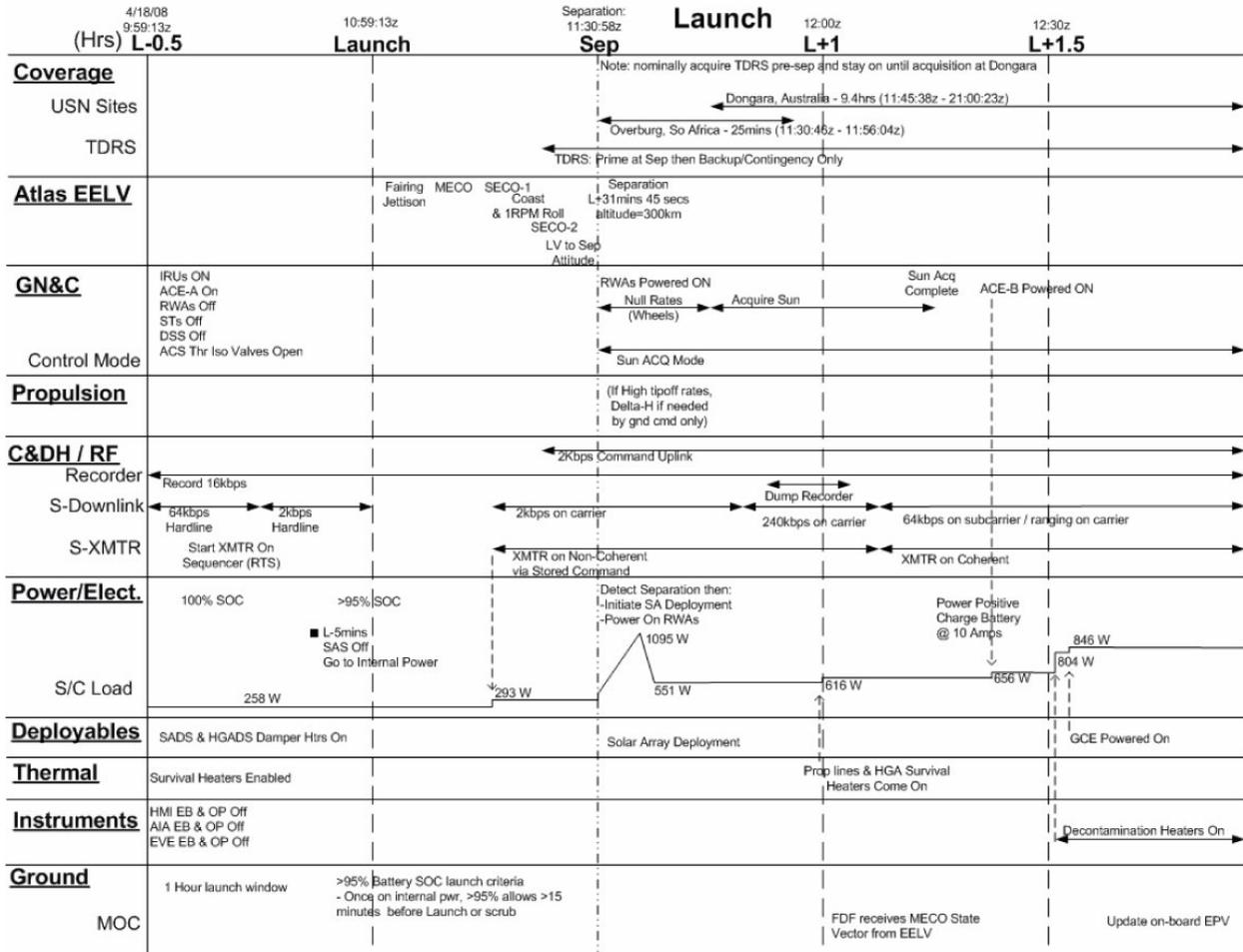


Figure 4-6b. Example of a Lunar Sortie Design Reference Mission



**Figure 4-7 Example of a More Detailed Integrated Timeline Later in the Life Cycle for a Science Mission**

An important part of the ConOps is defining the operational phases which will span project Phases D, E, and F. The operational phases provide a time-sequenced structure for defining the configuration changes and operational activities needed to be carried out to meet the goals of the mission. For each of the operational phases, facilities, equipment, and critical events should also be included. Table 4-1 identifies some common examples of operational phases for a NASA mission.

**Table 4-1 Typical Operational Phases for a NASA Mission**

Operational Phase	Description
Integration and Test Operations	<p><i>Project Integration and Test</i>—During the later period of project I&amp;T, the system is tested by performing operational simulations during functional and environmental testing. The simulations typically exercise the end-to-end command and data system to provide a complete verifications of system functionality and performance against simulated project operational scenarios.</p> <p><i>Launch Integration</i>—The launch integration phase repeats I&amp;T operational and functional verification in the launch integrated configuration.</p>
Launch Operations	<p><i>Launch</i>—Launch operation occur during the launch countdown, launch ascent, and orbit injection. Critical event telemetry is an important driver during this phase.</p> <p><i>Deployment</i>—Following orbit injection, spacecraft deployment operations reconfigure the spacecraft to its orbital configuration. Typically, critical events covering solar array, antenna and other deployments, and orbit trim maneuvers occur during this phase.</p> <p><i>In-orbit checkout</i>—In-orbit checkout performs is used to perform a verification that all systems are healthy. This is followed by on-orbit alignment, calibration, and parameterization of the flight systems to prepare for science operations.</p>
Science Operations	<p>The majority of the operational lifetime is used to perform science operations.</p>
Safe Hold Operations	<p>As a result of on-board fault detection, or by ground command, the spacecraft may transition to a safe hold mode. This mode is designed to maintain the spacecraft in a power positive, thermally stable state until the fault is resolved and science operations can resume.</p>
Anomaly Resolution and Maintenance Operations	<p>Anomaly resolution and maintenance operations often requires the support of personnel beyond those used for science operations.</p>
Disposal Operations	<p>Disposal operations occur at the end of project life. These operations are used to either provide a controlled reentry of the spacecraft, or a repositioning of the spacecraft to a disposal orbit. In the latter case, the dissipation of stored fuel and electrical energy is required.</p>

## 4.2 Technical Requirements Definition

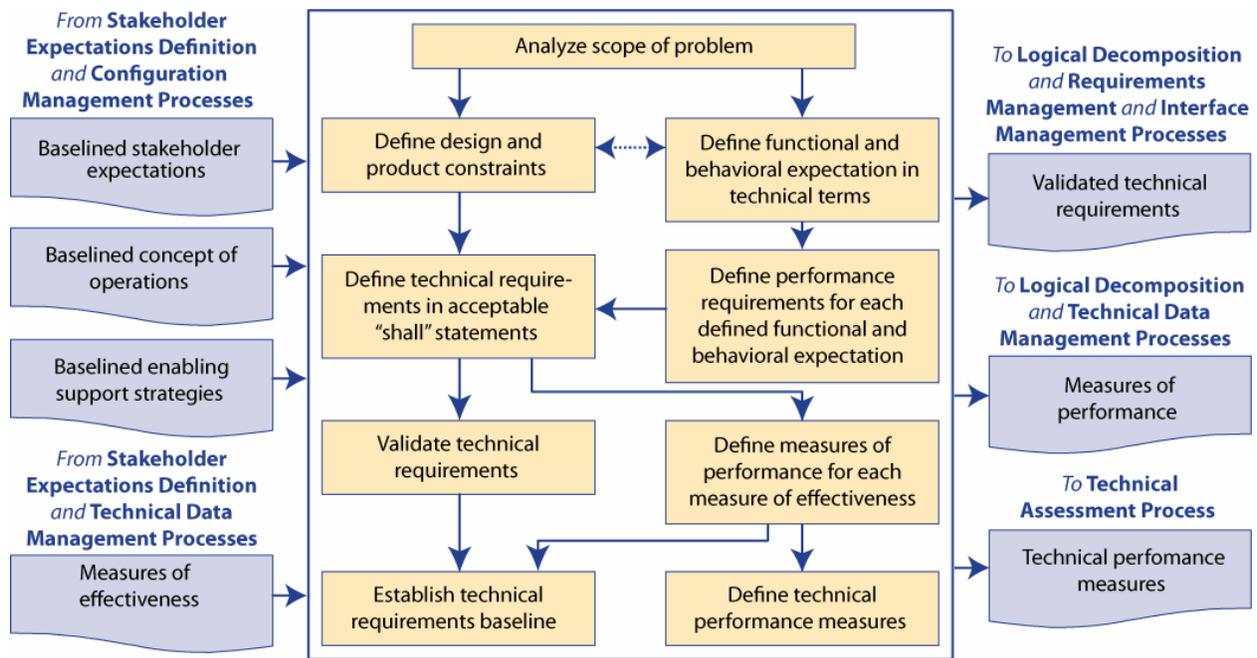
The Technical Requirements Definition process is used to transform the stakeholder expectations into a complete set of validated technical requirements expressed as “shall” statements that can be used for defining a design solution for the Product Breakdown Structure (PBS) model and related enabling products. The process of requirements definition is a recursive and iterative one in that it is performed to develop the customer requirements, the product requirements, and the lower level product component requirements (e.g., PBS model products such as systems or subsystems, and related enabling products such as external systems that provide or consume data.) The requirements should enable the description of all inputs, outputs, and required relationships between inputs and outputs. The requirements documents organize and communicate requirements to the customer and the technical community.

It is important to note that the team must not rely solely on the requirements received to design and build the system. Communication and iteration with the relevant customers are essential to ensure a mutual understanding of each requirement. Otherwise, the designers run the risk of misunderstanding and implementing an unwanted solution to a different interpretation of the requirements. Requirements that are subject to different interpretations should be clarified in writing.

Technical Requirements Definition activities apply to the definition of all technical requirements from program, project, and system level down to the lowest level product component requirements document.

### 4.2.1 Process Description

Figure 4-8 provides a typical flow diagram for the Technical Requirements Definition Process and identifies typical inputs, outputs, and activities to consider in addressing technical requirements definition.



**Figure 4-8 Technical Requirements Definition Process**

#### 4.2.1.1 Inputs

Typical inputs needed for the requirements process would include the following:

- **Top-Level Requirements and Expectations:** These would be the agreed to top-level requirements and expectations (e.g., needs, wants, desires, capabilities, constraints, external interfaces) for the product(s) to be developed coming from the customer and stakeholders.
- **Concept of Operations:** This describes how the system will be operated during the life-cycle phases to meet stakeholder expectations. It describes the system characteristics from an operational perspective and helps facilitate an understanding of the system goals. Examples would be a ConOps document or a design reference mission (DRM).

#### 4.2.1.2 Process Activities

The top-level requirements and expectations are initially assessed to understand the technical problem to be solved and establish the design boundary. This boundary is typically established by performing the following activities:

- Defining constraints that the design must adhere to or how the system will be used. The constraints are typically not able to be changed based on tradeoff analyses.
- Identifying those elements that are already under design control and cannot be changed. This helps establish those areas where further trades will be performed to narrow potential design solutions.

- Establishing physical and functional interfaces (e.g., mechanical, electrical, thermal, human, etc.) that the system must interact with
- Defining functional and behavioral expectations for the range of anticipated uses of the system as identified in the ConOps. The ConOps will describe how the system will be operated and the possible use case scenarios.

With an overall understanding of the constraints, physical/functional interfaces, and functional/behavioral expectations, the requirements can be further defined by establishing performance criteria. The performance is expressed as the quantitative part of the requirement to indicate how well each product function is expected to be accomplished.

Finally, the requirements should be defined in acceptable “shall” statements that are complete sentences with a single “shall” per statement. See appendix: 4-2a for the rules on how to write good requirements and appendix 4-2c for validating requirements. A well written requirements document provides several specific benefits to both the customer and the technical team, as shown in Table 4-2.

**Table 4-2 Benefits of Well-Written Requirements**

Establish the basis for agreement between the customers and the developers on what the product is to do.	The complete description of the functions to be performed by the product specified in the requirements will assist the potential users in determining if the product specified meets their needs or how the product must be modified to meet their needs. During systems design, requirements are allocated to subsystems (e.g., hardware, software, and other major components of the system) people or processes.
Reduce the development effort because less rework is required to address poorly written, missing, and misunderstood requirements.	The Technical Requirements Definition activities force the relevant stakeholders to consider rigorously all of the requirements before design begins. Careful review of the requirements can reveal omissions, misunderstandings, and inconsistencies early in the development cycle when these problems are easier to correct thereby reducing costly redesign, remanufacture, recoding, and retesting in later life-cycle phases.
Provide a basis for estimating costs and schedules.	The description of the product to be developed as given in the requirements is a realistic basis for estimating project costs and can be used to evaluate bids or price estimates.
Provide a baseline for validation and verification.	Organizations can develop their validation and verification plans much more productively from a good requirements document. Both system and subsystem test plans and procedures are generated from the requirements. As a part of the development contract, the requirements document provides a baseline against which compliance can be measured. The requirements are also used to provide the customer with a basis for acceptance of the system.
Facilitate transfer.	The requirements make it easier to transfer the product to new users or new machines. Customers thus find it easier to transfer the product to other parts of their organization, and developers find it easier to transfer it to new customers or reuse it.
Serve as a basis for enhancement.	The requirements serve as a basis for later enhancement or alterations of the finished product.

### 4.2.1.3 *Outputs*

Typical outputs for requirements definition process would include the following:

- **Technical Requirements:** This would be the approved set of requirements that represents a complete description of the problem to be solved and requirements that have been validated and approved by the customer and stakeholders. Examples of documentation that capture the requirements are a System Requirements Document (SRD), Project Requirements Document (PRD), Interface Requirements Document (IRD), etc.
- **Technical Measures:** An established set of measures based on the expectations and requirements that will be tracked and assessed to determine overall system or product effectiveness and customer satisfaction. Common terms for these measures are Measures of Effectiveness (MOEs), Measures of Performance (MOPs), and Technical Performance Measures (TPMs). See the Technical Assessment section of this handbook for further details.

## 4.2.2 *Technical Requirements Definition Guidance*

### 4.2.2.1 *Types of Requirements*

A complete set of project requirements includes the functional needs requirements, (what functions need to be performed) performance requirements (how well these functions must be performed) and interface requirements (design element interface requirements). For space projects, these requirements are decomposed and allocated to design elements, down through the product breakdown structure (PBS).

Functional, performance, and interface requirements are very important but do not constitute the entire set of requirements necessary for project success. The space segment design elements must also survive and continue to perform in the project environment. These environmental drivers include radiation, thermal, acoustic, mechanical loads, contamination, RF and others. In addition, reliability requirements drive design choices in design robustness, failure tolerance and redundancy. Safety requirements drive design choices in providing diverse functional redundancy. Other specialty requirements also may affect design choices. These may include producibility, maintainability, availability, upgradeability, human factors, and others. Unlike functional needs requirements which are decomposed and allocated to design elements, these requirements are levied across major project elements. Designing to meet these requirements requires careful analysis of design alternatives. Figure 4-9 shows the characteristics of functional, operational, reliability, safety, and specialty requirements. Top-level mission requirements are generated from mission objectives, programmatic constraints, and assumptions. These are normally grouped into function and performance requirements and include the categories of requirements in Figure 4-9.



**Figure 4-9 Characteristics of Functional, Operational, Reliability, Safety, and Specialty Requirements**

## **Functional Requirements**

The functional requirements need to be specified for all intended uses of the product over its entire lifetime. Functional analysis is used to draw out both functional and performance requirements. Requirements are partitioned into groups, based on established criteria (e.g., similar functionality, performance, or coupling, etc.), to facilitate and focus the requirements analysis. Functional and performance requirements are allocated to functional partitions, and sub functions, objects, people, or processes. Sequencing of time-critical functions is considered. Each function is identified and described in terms of inputs, outputs, and interface requirements from top down so that the decomposed functions are recognized as part of larger functional groupings. Functions are arranged in a logical sequence so that any specified operational usage of the system can be traced in an end-to-end path to indicate the sequential relationship of all functions that must be accomplished by the system.

It is helpful to walk through the concept of operations and scenarios asking the following types of questions: what functions need to be performed, where do they need to be performed, how often, under what operational and environmental conditions, etc. Thinking through this process often reveals additional functional requirements.

Functional requirements define what functions need to be done to accomplish the objectives.

Performance requirements define how well the system needs to perform the functions.

### Example of Functional and Performance Requirements

**Initial function statement:** The Thrust Vector Controller (TVC) shall provide vehicle control about the pitch and yaw axes.

This statement describes a high level function that the TVC must perform. The technical team needs to transform this statement into a set of “design to” functional and performance requirements.

**Functional requirements with associated performance requirements:**

- The TVC shall gimbal the engine a maximum of 9 degrees, +/- 0.1 degree.
- The TVC shall gimbal the engine at a maximum rate of 5 degrees/sec, +/-0.3 degrees/sec.
- The TVC shall provide a force of 40,000 pounds, +/- 500 pounds.
- The TVC shall have a frequency response of 20 Hz, +/- 0.1 Hz.

### Performance Requirements

Performance requirements quantitatively define how well the system needs to perform the functions. Again, walking through the concept of operations and the scenarios often draws out the performance requirements by asking the following types of questions: how often and how well; to what accuracy; what is the quality and quantity of the output; under what stress (maximum simultaneous data requests) or environmental conditions; for what duration; at what range of values, at what tolerance; at what maximum throughput or bandwidth capacity.

Be careful not to make performance requirements too restrictive. For example, for a system that must be able to run on rechargeable batteries, if the performance requirements specify that the time to recharge should be less than 3 hours when a 12-hour recharge time would be sufficient, potential design solutions are eliminated.

Wherever possible, define the performance requirements in terms of 1) a threshold value (the minimum acceptable value needed for the system to carry out its mission) and 2) the baseline level of performance desired. Specifying performance in terms of thresholds and baseline requirements provides the system designers with ‘trade space’ in which to investigate alternative designs.

All qualitative performance expectations must be analyzed and translated into quantified performance requirements. Trade studies often help quantify performance requirements. For example, tradeoffs can show whether a slight relaxation of the performance requirement could produce a significantly cheaper system or whether a few more resources could produce a significantly more effective system. The rationale for thresholds and goals should be documented

with the requirements in order to understand the reason and origin for the performance requirement in case it must be changed. The performance requirements that can be quantified by or changed by tradeoff analysis should be identified. See Section 6.8, Decision Analysis, for more information on tradeoff analysis.

## **Interface Requirements**

It is important to define all external interface requirements for the system, including those to enabling systems. The external interfaces form the boundaries between the product and the rest of the world. Types of interfaces include: operational command and control; computer to computer; mechanical; electrical; thermal; data. One useful tool in defining interfaces is the context diagram that depicts the product and all of its external interfaces. Once the product components are defined, a block diagram showing the major components of the system, interconnections, and external interfaces should be developed to define both the components and their interactions.

Interfaces associated with all product life-cycle phases should also be considered. Examples include interfaces with test equipment, transportation systems, ILS systems, and manufacturing facilities, operators, users, maintainers etc.

As the technical requirements are defined, revisit the interface diagram and refine the documented interface requirements to include newly identified interfaces information for requirements both internal and external. More information regarding interfaces could be found in section 6.3.

## **Environmental Requirements**

Each space mission has a unique set of environmental requirements that apply to the flight segment elements. It is a critical function of systems engineering to identify the external and internal environments for their particular mission, analyze and quantify the expected environments, develop design guidance and establish a margin philosophy against the expected environments.

The environments envelope should consider what can be encountered during ground test, storage, transportation, launch, deployment and normal operations from beginning-of-life to end-of-life. Requirements derived from the mission environments should be included in the system requirements.

External environments include vibration, shock, static loads, acoustic, thermal, contamination, total dose radiation, single event effects (SEE), surface and internal charging, orbital debris, atmospheric (atomic oxygen), attitude control system disturbance (atmospheric drag, gravity gradient, and solar pressure), magnetic, and radio frequency (RF) exposure on the ground and on orbit.

The requirements structure must address the specialty engineering disciplines that apply to the mission environments across project elements. These discipline areas levy requirements on system elements regarding electromagnetic interference, electromagnetic compatibility (EMI/EMC), grounding, radiation and other shielding, contamination protection, and reliability.

## **Reliability Requirements**

Reliability can be defined as the probability that a device, product, or system will not fail for a given period of time under specified operating conditions. Reliability is an inherent system design characteristic. As a principal contributing factor in operations and support costs and in system effectiveness, reliability plays a key role in determining the system's cost-effectiveness.

Reliability engineering is a major specialty discipline that contributes to the goal of a cost-effective system. This is primarily accomplished in the systems engineering process through an active role in implementing specific design features to ensure that the system can perform in the predicted physical environments throughout the mission, and by making independent predictions of system reliability for design trades and for test program, operations, and integrated logistics support planning.

Reliability requirements ensure that the system (and subsystems e.g., software, hardware, etc.) can perform in the predicted environments and conditions as expected, throughout the mission and that the system has the ability to withstand certain numbers and types of faults, errors, or failures (e.g., withstand vibration, predicted data rates, command and/or data errors, single event upsets, and temperature variances to specified limits, etc.). Environments can include ground (transportation and handling), launch, on-orbit (Earth or other), planetary, re-entry, and landing; or they might be for software within certain modes or states of operation. Reliability addresses design and verification requirements to meet the requested level of operation as well as fault and/or failure tolerance for all expected environments and conditions. Reliability requirements cover fault/failure prevention, detection, isolation, and recovery.

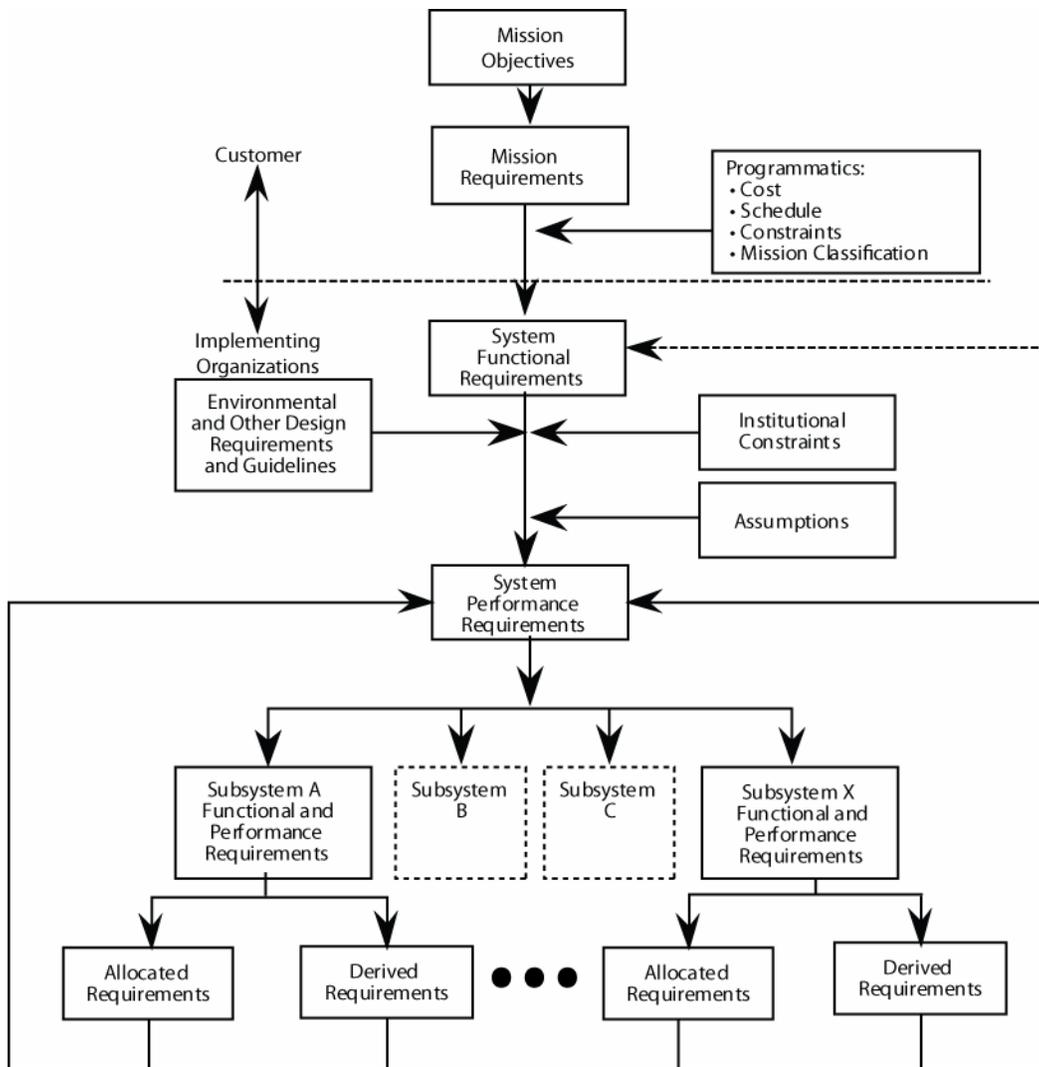
## **Safety Requirements**

Safety requirements ensure freedom from conditions that can cause death, injury, occupational illness, damage to or loss of equipment or property, or damage to the environment. This is accomplished through identification and elimination of hazards, reducing the likelihood of the accidents associated with hazards, or reducing the impact from the hazard associated with these accidents to within acceptable levels. For instance, safety related requirements include: incorporation of safety devices (e.g., build physical hardware stops into the system to prevent the hydraulic lift/arm from extending past allowed safety height/length limits); limits on the range of values a system input variable is allowed to take on; limit checks on input commands to ensure they are within specified safety limits/constraints for that mode or state of the system (e.g., the command to retract the landing gear is only allowed if the airplane is in the airborne state); for those components identified as safety critical, require functional redundancy or failure tolerance to allow the system to meet its requirements in the presence of one or more failures, or to take the system to a safe state with reduced functionality (e.g., dual redundant computer processors, safe-state back-up processor); detection and automatic system shutdown if specified values (e.g., temperature value) exceeds prescribed safety limits; use of only a subset of a particular computer language that is approved for safety critical software; caution/warning devices; barriers; safety procedures. (For additional information concerning safety, see for example NPR 8705.2, Human-Rating Requirements for Space Flight Systems, NPR 8715.3, NASA General Safety Program Requirements, and NASA-STD-8719.13, Software Safety Standard.)

#### *4.2.2.2 Requirements Decomposition, Allocation, and Validation*

Requirements are decomposed in a hierarchical structure starting with the highest-level requirements imposed by Presidential Directives, Mission Directorates, program, Agency and customer stakeholders. These high-level requirements are decomposed into functional and performance requirements and allocated across the system. These are then further decomposed and allocated among the elements and subsystems. This decomposition and allocation process continues until a complete set of design-to requirements is achieved. At each level of decomposition (system, subsystem, component, etc.), the total set of derived requirements must be validated against the stakeholder expectations or higher-level parent requirements before proceeding to the next level of decomposition.

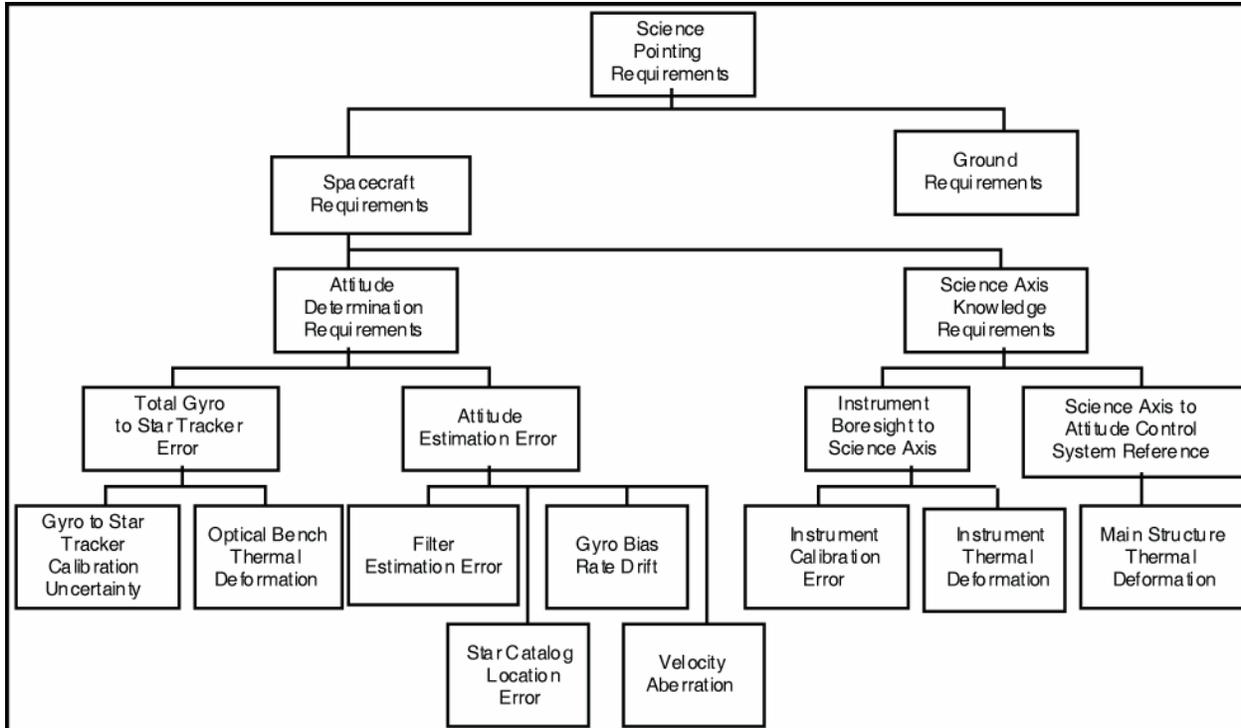
The traceability of requirements to the lowest level ensures that each requirement is necessary to meet the stakeholder expectations. Requirements that are not allocated to lower-levels or are not implemented at a lower-level result in a design that does not meet objectives and is, therefore, not valid. Conversely, lower-level requirements that are not traceable to higher-level requirements result in an over-design that is not justified. This hierarchical flow down is illustrated in Figure 4-10.



**Figure 4-10 The Flow Down of Requirements**

Figure 4-11 is an example of how science pointing requirements are successively decomposed and allocated from the top-down for a typical science mission. It is important to understand and document the relationship between requirements. This will reduce the possibility of misinterpretation and the possibility of an unsatisfactory design and associated cost increases.

Throughout Phases A and B, changes in requirements and constraints will occur. It is imperative that all changes be thoroughly evaluated to determine the impacts on both higher and lower hierarchical levels. All changes must be subjected to a review and approval cycle to maintain traceability and to ensure the impacts of any changes are fully assessed for all parts of the system. A more formal change control process is required if the mission is very large and involves more than one Center or crosses other jurisdictional or organizational boundaries.



**Figure 4-11 Allocation and Flow-Down of Science Pointing Requirements**

#### 4.2.2.3 Capturing Requirements and the Requirements Database

At the time the requirements are written, it is important to capture the requirement statement along with the meta data associated with each requirement. The meta data is the supporting information necessary to help clarify and link the requirements.

The method of verification must also be thought through and captured for each requirement at the time they are developed. The verification method includes test, inspection, analysis, and demonstration. Be sure to document any new or derived requirements that are uncovered during determination of the verification method. An example is requiring an additional test port to give visibility to an internal signal during integration and test. If a requirement cannot be verified then it either should not be a requirement or the requirement statement needs to be rewritten. For example, the requirement to “minimize noise” is vague and cannot be verified. If the requirement is restated as “the noise level of the component X shall remain under Y decibels ...” then it is clearly verifiable. Examples of the types of meta data is provided in Table 4-3.

**Table 4-3 Requirements Meta-Data**

Requirement ID	Provides a unique numbering systems for sorting and tracking
Rationale	Provides additional information to help clarify the intent of the requirements at the time it was written. See the sidebar as to what should be captured as part of the rationale.
Traced from	Captures the bidirectional traceability between parent requirements and lower level (derived) requirements and the relationships between requirements.
Owner	Person/group responsible for writing/managing and/or approving changes to this requirement
Verification method	Captures the method of verification (test, inspection, analysis) and should be determined as the requirements are developed.
Verification lead	Person/group assigned responsibility for verifying the requirement
Verification level	Specifies the level in the hierarchy that the requirements will be verified. (e.g., card, subsystem, system)

The requirements database is an extremely useful tool for capturing the requirements and the associated meta-data and for showing the bidirectional traceability between requirements. The database evolves over time and could be used for tracking status information related to requirements such as TBD/TBR status, resolution date, and verification status. Each project should decided what meta data will be captured. The database is usually in a central location that is made available to the entire project team. (See Appendix 4-2b for a sample requirements verification matrix.)

<b>Rationale</b>
<p>The rationale should be kept up to date and include the following information:</p> <p><i>Reason for the requirement:</i> Often the reason for the requirement is not obvious and it may be lost if not recorded as the requirement is being documented. The reason may point to a constraint or concept of operations. If there is a clear parent requirement or trade study that explains the reason, then reference it.</p> <p><i>Document Assumptions:</i> If a requirement was written assuming the completion of a technology development program or a successful technology mission, document it.</p> <p>Document the <i>relationships</i> with the product's expected operations (e.g., expectations about how customers will use a product). This may be done with a link to the concept of operations.</p> <p>Document <i>design constraints</i> imposed by the results from decisions made as the design evolves. If the requirement states a method of implementation the rationale should state why the decision was made to limit the solution to this one method of implementation.</p>

#### 4.2.2.4 Technical Standards

### Importance of Standards Application

Standards provide a proven basis for establishing common technical requirements across a program/project to avoid incompatibilities and ensure that at least minimum requirements are met. Common standards can also lower implementation cost as well as costs for inspection, common supplies, etc. Typically, standards (and specifications) are used throughout the product

life cycle to establish design requirements and margins, materials and process specifications, test methods, and interface specifications. Standards are used as requirements (and guidelines) for design, fabrication, verification, validation, acceptance, operations, and maintenance.

### **Selection of Standards**

NASA policy for technical standards is provided in *NPD 8070.6, Technical Standards*, which addresses selection, tailoring, application, and control of standards. In general, the order of authority among standards for NASA programs and projects is as follows:

- Standards mandated by law (e.g., environmental standards)
- National or international voluntary consensus standards recognized by industry
- Other Government standards
- NASA policy directives
- NASA technical standards

NASA may also designate mandatory or “core” standards that must be applied to all programs where technically applicable. Waivers to designated core standards must be justified and approved at the Agency level unless otherwise delegated.

Access to standards for NASA programs is provided through the NASA Technical Standards Data Base. Use of the data base’s search engine enables comparison of alternative standards. The data base contains summary information and access to the full-text standards. In addition, the data base features a process for connecting with the context for use of standards and improving changes to the standards. The home page for each standard provides hot-links to lessons learned related to specific, real-world use of relevant standards on programs and projects, application notes (recommendations based on experience), and registration for automatic notification of updates for identified standards. Consideration is given to non-NASA standards to increase technical and operational efficiencies by using industry-accepted practices often associated with and inherent to supplier implementation of one-of-a-kind requirements.

### 4.3 Logical Decomposition

Logical Decomposition is the process for creating the detailed functional requirements that enable NASA programs and projects to meet the ends desired by Agency stakeholders. This process identifies the “what” that must be achieved by the system at each level to enable a successful project. Logical decomposition utilizes functional analysis to create a system architecture and to decompose top-level (or parent) requirements and allocate them down to the lowest desired levels of the project.

The logical decomposition process is used to:

1. Improve understanding of the defined technical requirements and the relationships among the requirements (e.g., functional, behavioral, and temporal), and
2. Decompose the parent requirements into a set of logical decomposition models and their associated set of derived technical requirements for input to the design solution definition process.

#### 4.3.1 Process Description

Figure 4-12 provides a typical flow diagram for the Logical Decomposition Process and identifies typical inputs, outputs, and activities to consider in addressing logical decomposition.

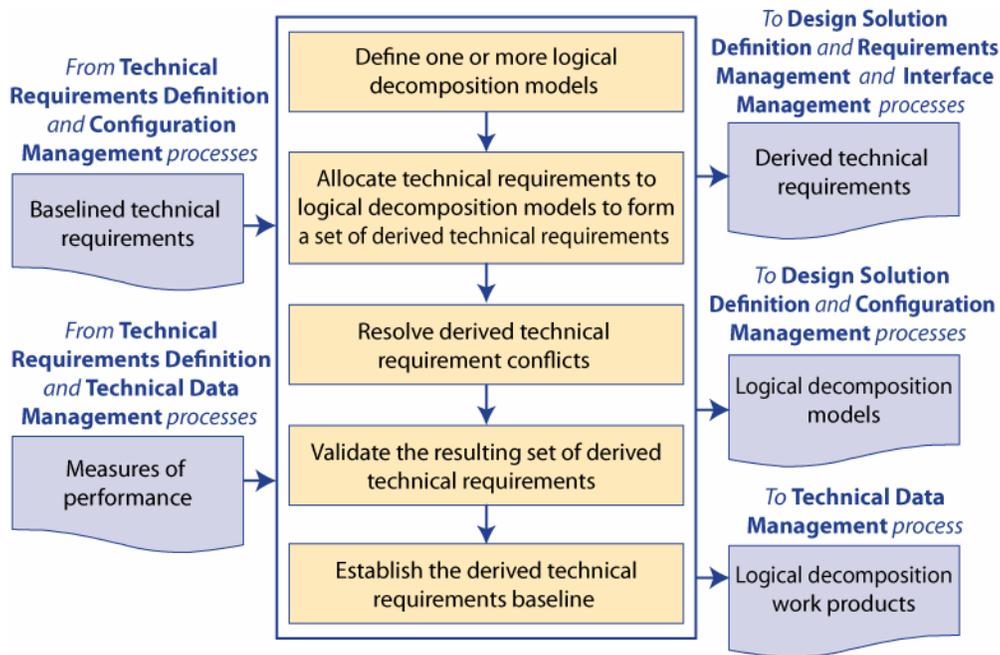


Figure 4-12 Logical Decomposition Process

#### 4.3.1.1 Inputs

Typical inputs needed for the logical decomposition process would include the following:

- **Technical Requirements:** This is a validated set of requirements that represent a description of the problem to be solved, have been established by functional and performance analysis, and have been approved by the customers. Examples of documentation that capture the requirements are a System Requirements Document (SRD), Project Requirements Document (PRD), Interface Requirements Document (IRD).
- **Technical Measures:** An established set of measures based on the expectations and requirements that will be tracked and assessed to determine overall system or product effectiveness and customer satisfaction. These measures are Measures of Effectiveness (MOEs), Measures of Performance (MOPs), and a special subset of these called Technical Performance Measures (TPMs). See the Technical Assessment section of this handbook for further details.

#### 4.3.1.2 Process Activities

The key first step in the logical decomposition process is establishing the system architecture model. The system architecture activity defines the underlying structure and relationship of hardware, software, communications, operations, etc. that provide for the implementation of Agency, Mission Directorate, program, project and subsequent levels of the requirements. System architecture activities drives the partitioning of system elements and requirements to lower-level functions and requirements to the point that design work can be accomplished. Interfaces and relationships between partitioned subsystems and elements are defined as well.

Once the top-level (or parent) functional requirements and constraints have been established, the system designer uses functional analysis to begin to formulate a conceptual system architecture. The system architecture can be seen as the strategic organization of the functional elements of the system, laid out to enable the roles, relationships, dependencies and the interfaces between elements to be clearly defined and understood. It is strategic in its focus on the overarching structure of the system and how its elements fit together to contribute to the whole, instead of on the particular workings of the elements themselves. It enables the elements to be developed separately from each other while ensuring that they work together effectively to achieve the top-level (or parent) requirements.

Much like the other elements of functional decomposition, the development of a good system-level architecture is a creative, recursive, and iterative process that combines an excellent understanding of the project's end objectives and constraints with an equally good knowledge of various potential technical means of delivering the end products.

Focusing on the project's ends, top-level (or parent) requirements and constraints, the system architect must develop at least one, but preferably multiple concept architectures capable of achieving program objectives. Each architecture concept involves specification of the functional elements (what the pieces do), their relationships to each other (interface definition) *and* the

concept of operations, i.e., how the various segments, subsystems, elements, units, etc., will operate as a system when distributed by location and environment from the start of the operations to the end-of-mission.

The development process for the architectural concepts must be recursive and iterative, with feedback from stakeholders and external reviewers, as well as from subsystem designers and operators, provided as often as possible to increase the likelihood of achieving the program's ends, while reducing the likelihood of cost and schedule overruns.

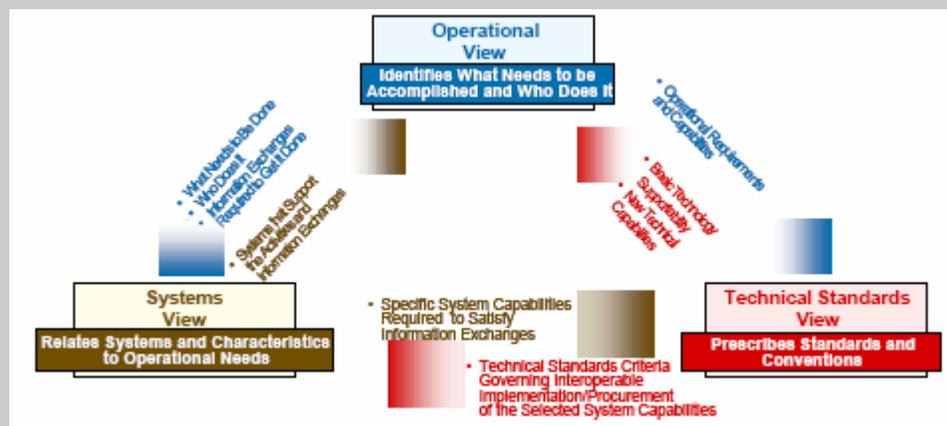
In the early stages of the mission, multiple concepts are developed. Cost and schedule constraints will ultimately limit how long a program or project can maintain multiple architectural concepts. For all NASA programs, architecture design is completed during the formulation phase. For most NASA projects (and tightly coupled programs), the selection of a single architecture will happen during Phase A and the architecture and concept of operations will be baselined during Phase B. Architectural changes at higher levels occasionally occur as decomposition to lower levels produces complications in design, cost, or schedule that necessitate such changes.

Aside from the creative minds of the architects, there are multiple tools that can be utilized to develop a system's architecture. These are primarily modeling and simulation tools, functional analysis tools, architecture frameworks and trade studies. (For example, one way of doing architecture is the DoD Architecture Framework. See sidebar.) As each concept is developed, analytical models of the architecture, its elements and their operations will be developed with increased fidelity as the project evolves. Functional decomposition, requirements development and trade studies are subsequently undertaken. Multiple iterations of these activities feed back to the evolving architectural concept as the requirements flow down and the design matures.

## DoD Architecture Framework

New ways, called *architecture frameworks*, have been developed in the last decade to describe and characterize evolving complex system-of-systems. In such circumstances, architecture descriptions are very useful in ensuring that stakeholder needs are clearly understood and prioritized, that critical details such as interoperability are addressed upfront, and that major investment decisions are made strategically. In recognition of this, the U.S. Department of Defense (DoD) has established policies that mandate the use of the DoD Architecture Framework (DoDAF) in capital planning, acquisition, and joint capabilities integration.

An architecture can be understood as “the structure of components, their relationships, and the principles and guidelines governing their design and evolution over time.”<sup>\*</sup> To describe an architecture, the DoDAF defines several views: Operational, Systems, and Technical Standards. In addition, a dictionary and summary information are also required. (See figure below.)



Within each of these views, DoDAF contains specific *products*. For example, within the Operational View is a description of the operational nodes, their connectivity, and information exchange requirements. Within the Systems View is a description of all the systems contained in the operational nodes and their interconnectivity. Not all DoDAF products are relevant to NASA systems engineering, but its underlying concepts and formalisms may be useful in structuring complex problems for the requirements definition and decision analysis processes.

<sup>\*</sup> Definition based on IEEE STD 610.12

Functional analysis is the primary method used in system architecture development and functional requirement decomposition. It is the systematic process of identifying, describing, and relating the functions a system must perform to fulfill its goals and objectives. Functional analysis identifies and links system functions, trade studies, interface characteristics, and rationales to requirements. It is usually based on the ConOps for the system of interest.

Three key steps in performing functional analysis are:

- Translate top-level requirements into functions that must be performed to accomplish the requirements,
- Decompose and allocate the functions to lower levels of the PBS
- Identify and describe functional and subsystem interfaces.

The process involves analyzing each system requirement to identify all of the functions that must be performed to meet the requirement. Each function identified is described in terms of inputs, outputs, and interface requirements. The process is repeated from the top down so that sub-functions are recognized as part of larger functional areas. Functions are arranged in a logical sequence so that any specified operational usage of the system can be traced in an end-to-end path.

The process is recursive and iterative and continues until all desired levels of the architecture/system have been analyzed, defined, and baselined. There will almost certainly be alternate ways to decompose functions; therefore, the outcome is highly dependent on the creativity, skills, and experience of the engineers doing the analysis. As the analysis proceeds to lower levels of the architecture and system and the system is better understood, the systems engineer must keep an open mind and a willingness to go back and change previously established architecture and system requirements. These changes will then have to be decomposed down through the architecture and systems again, with the recursive process continuing until the system is fully defined, with all of the requirements understood and known to be viable, verifiable, and internally consistent. Only at that point should the system architecture and requirements be baselined.

#### *4.3.1.3 Outputs*

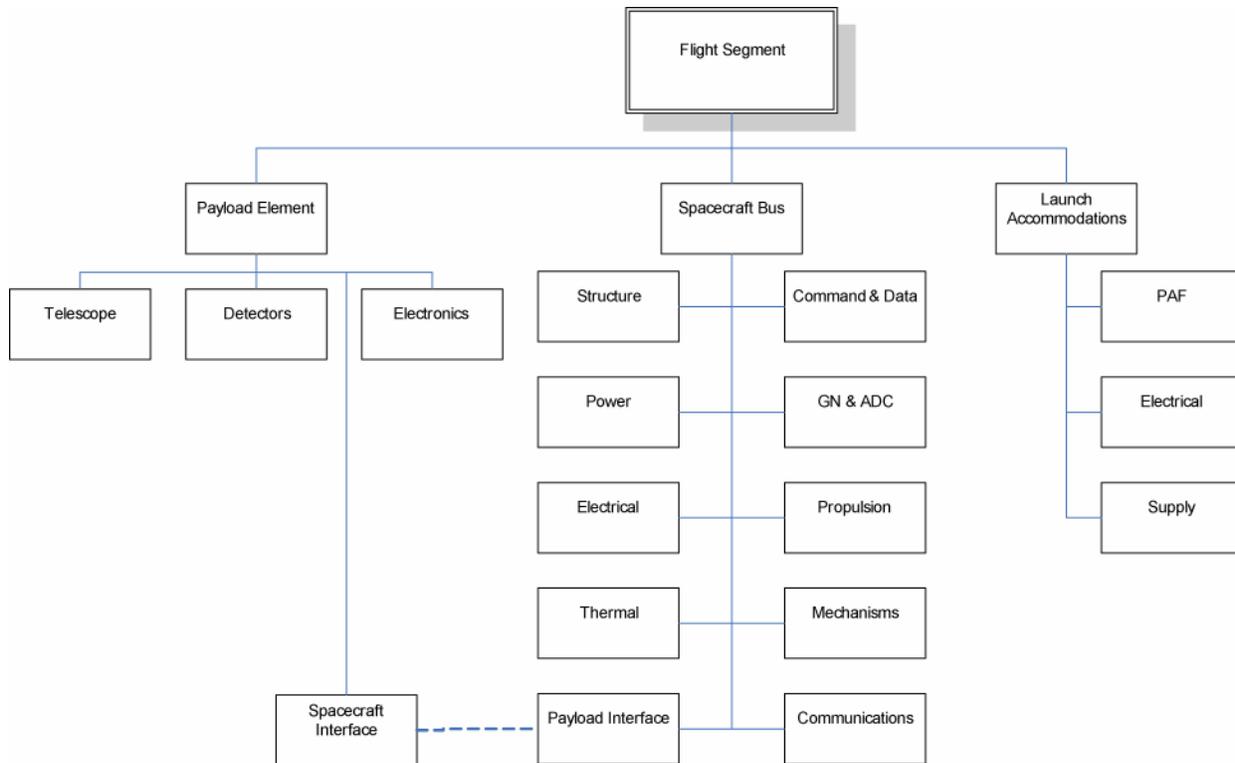
Typical outputs of the logical decomposition process would include the following:

- **System Architecture Model:** Defines the underlying structure and relationship of the elements of the system (e.g., hardware, software, communications, operations, etc.) and the basis for the partitioning of requirements into lower levels to the point that design work can be accomplished.
- **End Product Requirements:** A defined set of “make-to”, “buy-to”, “code-to”, etc. requirements from which design solutions can be accomplished.

### **4.3.2 Logical Decomposition Guidance**

#### *4.3.2.1 Product Breakdown Structure*

The decompositions represented by the product breakdown structure (PBS) and the work breakdown structure (WBS) form important perspectives on the desired product system. The WBS is a hierarchical breakdown of the *work* necessary to complete the project. See the Technical Planning section for further information on WBS development. The WBS contains the PBS, which is the hierarchical breakdown of the products such as hardware items, software items, and information items (documents, databases, etc.). The PBS is used during the logical decomposition and functional analysis processes. The PBS should be carried down to the lowest level for which there is a cognizant engineer or manager. Figure 4-13 is an example of a PBS.



**Figure 4-13 An Example of a PBS**

#### 4.3.2.2 Functional Analysis Techniques

Although there are many techniques available to perform functional analysis, some of the more popular are: 1) functional flow block diagrams (FFBDs) to depict task sequences and relationships, 2) N2 diagrams to develop data interfaces, and 3) time-line analyses to depict the time sequence of time-critical functions.

#### Functional Flow Block Diagrams

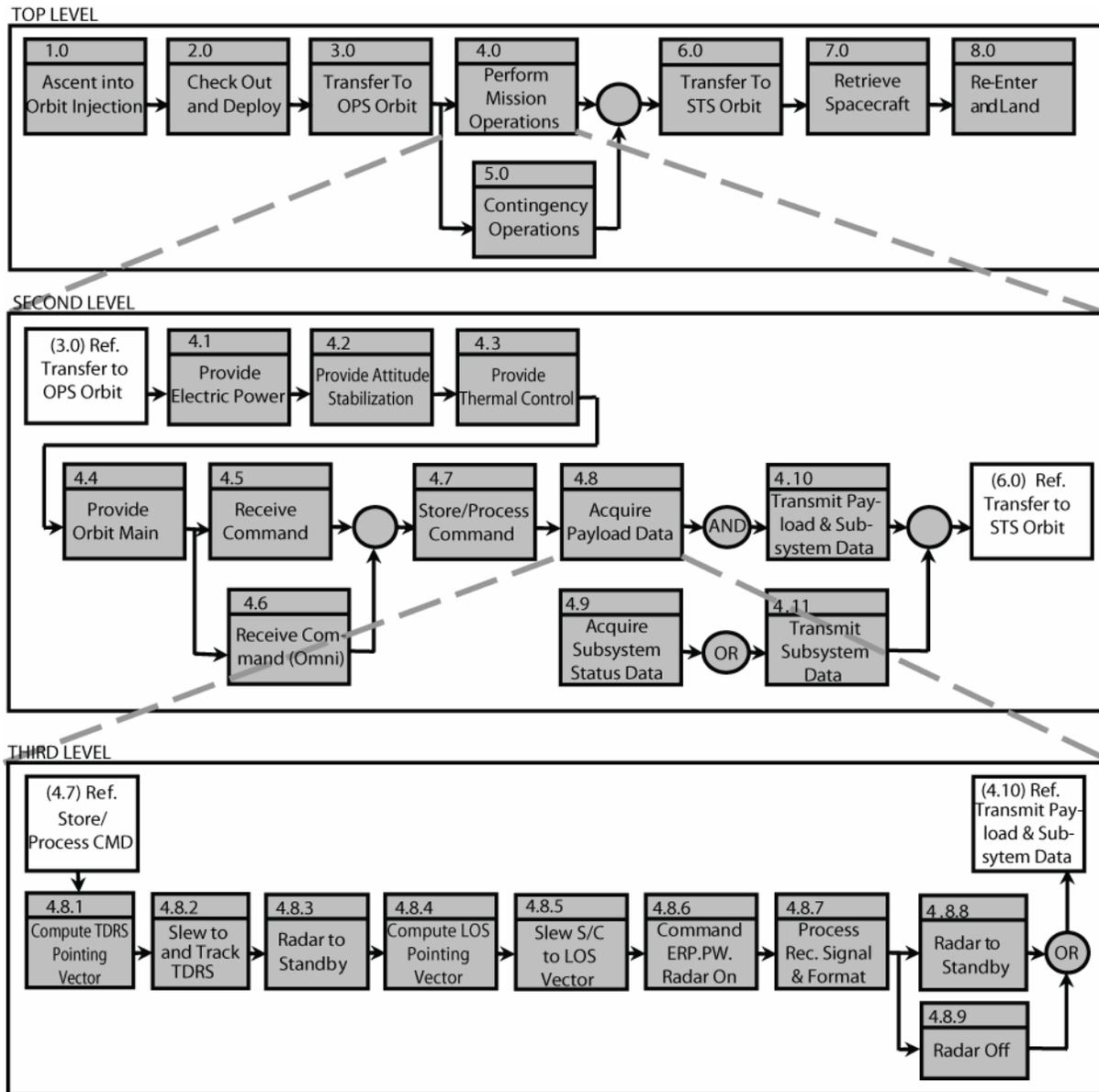
The primary functional analysis technique is the Functional Flow Block Diagram (FFBD). The purpose of the FFBD is to indicate the sequential relationship of all functions that must be accomplished by a system. When completed, these diagrams show the entire network of actions that lead to the fulfillment of a function.

FFBDs specifically depict each functional event (represented by a block) occurring following the preceding function. Some functions may be performed in parallel, or alternate paths may be taken. The FFBD network shows the logical sequence of “what” must happen, it does not ascribe a time-duration to functions or between functions. The duration of the function and the time between functions may vary from a fraction of a second to many weeks. To understand *time critical* requirements, a Time-Line Analysis (TLA) is used (see the TLA discussion below).

The FFBDs are function oriented, not equipment oriented. In other words, they identify “what” must happen and must not assume a particular answer to “how” a function will be performed.

The “how” is then defined for each block at a given level by defining the “what” functions at the next lower level necessary to accomplish that block. In this way, FFBDs are developed from the top down, in a series of levels, with tasks at each level identified through functional decomposition of a single task at a higher level. The FFBD displays all of the tasks at each level in their logical, sequential relationship, with their required inputs and anticipated outputs (including metrics, if applicable), plus a clear link back to the single, higher level task.

An example of a FFBD is shown in Figure 4-14. The FFBD depicts the entire flight mission of a spacecraft. Each block in the first level diagram is expanded to a series of functions, as shown in the second level diagram for “perform mission operations.” Note that the diagram shows both input (transfer to operational orbit) and output (transfer to space transportation system orbit), thus initiating the interface identification and control process. Each block in the second level diagram can be progressively developed into a series of functions, as shown in the third level diagram.



**Figure 4-14 Example of a Functional Flow Block Diagram**

These diagrams are used to develop, analyze and flow down requirements, as well as to identify profitable trade studies, by identifying alternative approaches to performing each function. In certain cases, alternate FFBDs may be used to represent various means of satisfying a particular function until trade study data are acquired to permit selection among the alternatives.

The flow diagram also provides an understanding of total operation of the system, serves as a basis for development of operational and contingency procedures, and pinpoints areas where changes in operational procedures could simplify the overall system operation.

## N2 Diagrams

The N-squared ( $N^2$ ,  $N^2$ ) diagram is used to develop system interfaces. An example of an N2 diagram is shown in Figure 4-15. The system components or functions are placed on the diagonal; the remainder of the squares in the  $N \times N$  matrix represent the interface inputs and outputs. Where a blank appears, there is no interface between the respective components or functions. The N2 diagram can be taken down into successively lower levels to the component functional levels. In addition to defining the interfaces, the N2 diagram also pinpoints areas where conflicts could arise in interfaces, and highlights input and output dependency assumptions and requirements.

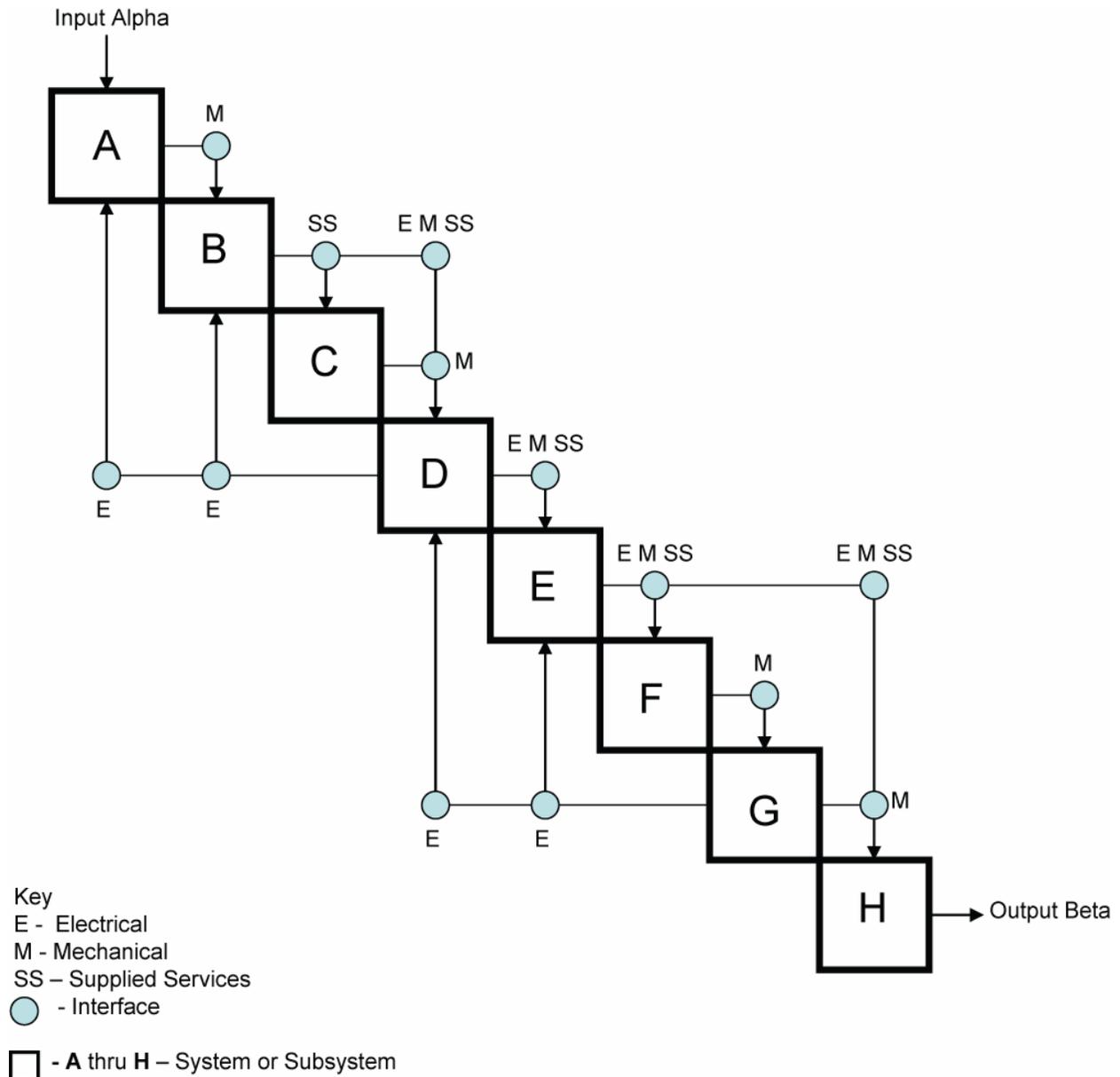


Figure 4-15 Example of an N-squared diagram for orbital equipment

## **Time-Line Analysis**

Time-Line Analysis (TLA) adds consideration of functional durations and is performed on those areas where time is critical to mission success, safety, utilization of resources, minimization of down time, and/or increasing availability. TLA can be applied to such diverse operational functions as spacecraft command sequencing and launch, but for those functional sequences where time is not a critical factor, FFBDs or N2 diagrams are sufficient. The following areas are often categorized as time-critical: 1) functions affecting system reaction time, 2) mission turnaround time, 3) time countdown activities, and 4) functions for which optimum equipment and/or personnel utilization are dependent on the timing of particular activities.

The Time-Line Sheet (TLS) is used to perform and record the analysis of time-critical functions and functional sequences. For time-critical functional sequences, the time requirements are specified with associated tolerances. Additional tools such as mathematical models and computer simulations may be necessary to establish the duration of each time line.

For additional information on FFBD, N2 diagrams, time line analysis, and other functional analysis methods, see appendix 4-3.

## 4.4 Design Solution Definition

The design solution definition process is used to translate the high-level requirements derived from the stakeholder expectations and the outputs of the logical decomposition process into a design solution. This involves transforming the defined logical decomposition models and their associated sets of derived technical requirements into alternative solutions. These alternative solutions are then analyzed through detailed trade studies which result in the selection of a preferred alternative. This preferred alternative is then fully defined into a final design solution that will satisfy the technical requirements. This design solution definition will be used to generate the end product specifications which will be used produce the product and to conduct product verification. This process may be further refined depending on whether there are additional subsystems of the end product that need to be defined.

### 4.4.1 Process Description

Figure 4-16 provides a typical flow diagram for the Design Solution Definition Process and identifies typical inputs, outputs, and activities to consider in addressing design solution definition.

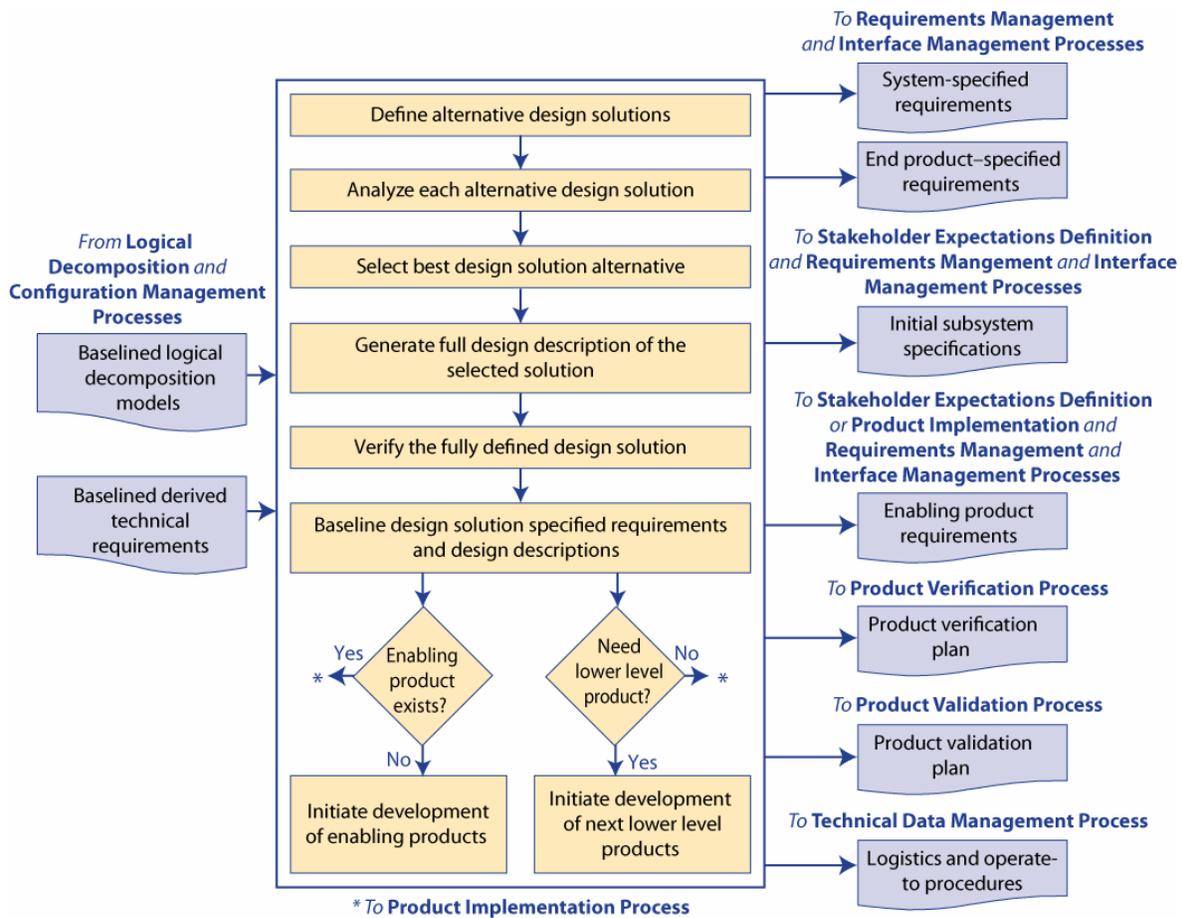


Figure 4-16 Design Solution Definition Process

#### 4.4.1.1 Inputs

There are several fundamental inputs needed to initiate the design solution definition process:

- **Technical Requirements:** the customer and stakeholder needs that have been translated into a reasonably complete set of validated requirements for the system, including all interface requirements.
- **Logical Decomposition Models:** requirements decomposed by one or more different methods (e.g., function, time, behavior, data flow, states, modes, system architecture, etc.)
- **System Goals:** both quantitative and qualitative guidance on such things as performance criteria, technology gaps, system context, effectiveness, cost, schedule, and risk.

#### 4.4.1.2 Process Activities

##### Define Alternative Design Solutions

The realization of a system over its life cycle involves a succession of decisions among alternative courses of action. If the alternatives are precisely defined and thoroughly understood to be well differentiated in the cost-effectiveness space, then the systems engineer can make choices among them with confidence.

To obtain assessments that are crisp enough to facilitate good decisions, it is often necessary to delve more deeply into the space of possible designs than has yet been done, as is illustrated in Figure 4-17. It should be realized, however, that this illustration represents neither the project life cycle, which encompasses the system development process from inception through disposal, nor the product development process by which the system design is developed and implemented.

Each *create concepts* step in Figure 4-17 involves a recursive and iterative design loop driven by the set of stakeholder expectations where a strawman architecture/design, the associated ConOps, and the derived requirements are developed. These three products must be consistent with each other and will require iterations and design decisions to achieve this consistency. This recursive and iterative design loop is illustrated in the System Design Process figure (Figure 4-1).

Each *create concepts* step also involves an assessment of potential capabilities offered by the continually changing state of technology and potential pitfalls captured through experience-based review of prior program/project lessons learned data. It is imperative that there be a continual interaction between the technology development process and the design process in order to ensure that the design reflects the realities of the available technology. This interaction is facilitated through a periodic assessment of the design with respect to the maturity of the technology required to implement the design. (See section 4.4.2.1 for a more detailed discussion of technology assessment.) These technology elements usually exist at a lower-level in the product breakdown structure. Although the process of design concept development by the integration of lower-level elements is a part of the systems engineering process, there is always a danger that the top-down process cannot keep up with the bottom-up process. Therefore, system

architecture issues need to be resolved early so that the system can be modeled with sufficient realism to do reliable trade studies.

As the system is realized, its particulars become clearer—but also harder to change. The purpose of systems engineering is to make sure that the design solution definition process happens in a way that leads to the most cost-effective final system. The basic idea is that before those decisions that are hard to undo are made, the alternatives should be carefully assessed, particularly with respect to the maturity of the required technology.



Figure 4-17 The Doctrine of Successive Refinement

### Create Alternative Design Concepts

Once it is understood what the system is to accomplish, it is possible to devise a variety of ways that those goals can be met. Sometimes, that comes about as a consequence of considering alternative functional allocations and integrating available subsystem design options all of which can have technologies at varying degrees of maturity. Ideally, as wide a range of plausible alternatives as is consistent with the design organization’s charter should be defined, keeping in mind the current stage in the process of successive refinement. When the bottom-up process is operating, a problem for the systems engineer is that the designers tend to become fond of the designs they create, so they lose their objectivity; the systems engineer often must stay an “outsider” so that there is more objectivity. This is particularly true in the assessment of the technological maturity of the subsystems and components required for implementation. There is

a tendency on the part of technology developers and project management to overestimate the maturity of a technology that is required to implement a design. This is especially true of “heritage” equipment. The result is that critical aspects of systems engineering are often overlooked.

On the first turn of the successive refinement in Figure 4-17, the subject is often general approaches or strategies, sometimes architectural concepts. On the next, it is likely to be functional design, then detailed design, and so on. The reason for avoiding a premature focus on a single design is to permit discovery of the truly best design. Part of the systems engineer’s job is to ensure that the design concepts to be compared take into account all interface requirements. “Did you include the cabling?” is a characteristic question. When possible, each design concept should be described in terms of controllable *design parameters* so that each represents as wide a class of designs as is reasonable. In doing so, the systems engineer should keep in mind that the potentials for change may include organizational structure, schedules, procedures, and any of the other things that make up a *system*. When possible, constraints should also be described by parameters.

### **Analyze Each Alternative Design Solution**

The technical team analyzes how well each of the design alternatives meets the system goals (technology gaps, effectiveness, cost, schedule, and risk, both quantified and otherwise). This assessment is accomplished through the use of trade studies. The purpose of the trade study process is to ensure that the system architecture and design decisions move towards the best solution that can be achieved with the available resources. The basic steps in that process are:

- Devise some alternative means to meet the functional requirements. In the early phases of the project life cycle, this means focusing on system architectures; in later phases, emphasis is given to system designs.
- Evaluate these alternatives in terms of the Measures of Effectiveness (MOE) and system cost. Mathematical models are useful in this step not only for forcing recognition of the relationships among the outcome variables, but also for helping to determine what the Measures of Performance must be quantitatively.
- Rank the alternatives according to an appropriate selection criteria.
- Drop less-promising alternatives and proceed to next level of resolution, if needed.

This process must be done openly and inclusively. While quantitative techniques and rules are used, subjectivity also plays a significant role. To make the process work effectively, participants must have open minds and individuals with different skills—systems engineers, design engineers, specialty engineers, program analysts, decision scientists, and project managers—must cooperate. The right quantitative methods and selection criteria must be used. Trade study assumptions, models, and results must be documented as part of the project archives. The participants must remain focused on the functional requirements including those for enabling products. For an in-depth discussion of the trade study process see Section 6.8, Decision Analysis, Trade Studies. The ability to perform these studies is enhanced by the development of

system models that relate the design parameters to those assessments—but it does not depend upon them.

The technical team must consider a broad range of concepts when developing the system models. The model must define the roles of crew, hardware and software in the system. It must identify the critical technologies required to implement the mission and consider the entire life cycle, from fabrication to disposal. Evaluation criteria for selecting concepts must be established. Cost is always a limiting factor. However other criteria such as time to develop and certify a unit, and reliability, are also critical. This stage cannot be accomplished without addressing the roles of operators and maintainers. These contribute significantly to life-cycle costs and to the system reliability. Reliability analysis should be performed based upon estimates of component failure rates for hardware. If probabilistic risk assessment models are applied, it may be necessary to include occurrence rates or probabilities for software faults or human error events. Assessments of the maturity of the required technology must be done and a technology development plan developed.

Controlled modification and development of design concepts, together with such system models, often permits the use of formal optimization techniques to find regions of the design space that warrant further investigation.

Whether system models are used or not, the design concepts are developed, modified, reassessed, and compared against competing alternatives in a closed-loop process that seeks the best choices for further development. System and subsystem sizes are often determined during the trade studies. The end result is the determination of bounds on the relative cost-effectiveness of the design alternatives, measured in terms of the quantified system goals. (Only bounds, rather than final values, are possible because determination of the final details of the design is intentionally deferred.) Increasing detail associated with the continually improving resolution reduces the spread between upper and lower bounds as the process proceeds.

### **Selecting the “Best” Design Solution Alternative**

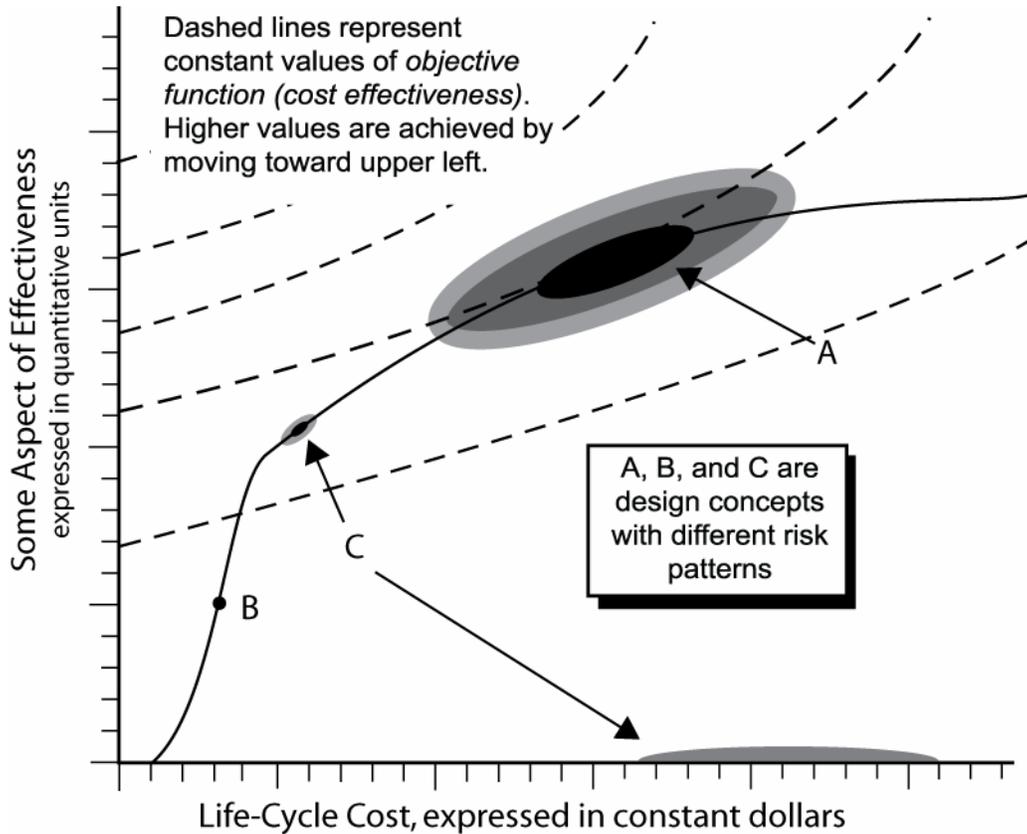
The technical team selects the best design solution from among the alternative design concepts, taking into account subjective factors that the team was unable to quantify, as well as estimates of how well the alternatives meet the quantified goals, the maturity of the available technology (and any effectiveness, cost, schedule, risk, or other constraints).

The decision analysis process, as described in section 6.8, should be used to make an evaluation of the alternative design concepts and to recommend the “best” design solution.

When it is possible, it is usually well worth the trouble to develop a mathematical expression, called an *objective function*, that expresses the values of combinations of possible outcomes as a single measure of cost-effectiveness, as is illustrated in Figure 4-18, even if both cost and effectiveness must be described by more than one measure.

The objective function (or *cost function*) assigns a real number to candidate solutions or *feasible solutions* in the alternative space or *search space*. A feasible solution that minimizes (or maximizes, if that is the goal) the objective function is called an *optimal solution*. When achievement of the goals can be quantitatively expressed by such an objective function, designs

can be compared in terms of their value. Risks associated with design concepts can cause these evaluations to be somewhat nebulous (because they are uncertain and are best described by probability distributions). In this illustration, the risks are relatively high for design concept A. There is little risk in either effectiveness or cost for concept B, while the risk of an expensive failure is high for concept C, as is shown by the cloud of probability near the x axis with a high cost and essentially no effectiveness. Schedule factors may affect the effectiveness values, the cost values, and the risk distributions.



**Figure 4-18 A Quantitative Objective Function, Dependent on Life-Cycle Cost and All Aspects of Effectiveness**

The mission success criteria for systems differ significantly. In some cases, effectiveness goals may be much more important than all others. Other projects may demand low costs, have an immutable schedule, or require minimization of some kinds of risks. Rarely (if ever) is it possible to produce a combined quantitative measure that relates *all* of the important factors, even if it is expressed as a vector with several components. Even when that can be done, it is essential that the underlying factors and relationships be thoroughly revealed to and understood by the systems engineer. The systems engineer must weigh the importance of the unquantifiable factors along with the quantitative data provided by the systems engineer.

Technical reviews of the data and analyses, including technology maturity assessments are an important part of the *decision support packages* prepared for the systems engineer. The decisions that are made are generally entered into the configuration management system as changes to (or

elaborations of) the system baseline. The supporting trade studies are archived for future use. An essential feature of the systems engineering process is that trade studies are performed *before* decisions are made. They can then be baselined with much more confidence.

### **Increase the Resolution of the Design**

The successive refinement process of Figure 4-17 illustrates a continuing refinement of the system design. At each level of decomposition, the baselined derived (and allocated) requirements become the set of high level requirements for the decomposed elements, and the process begins again. One might ask, “When do we stop refining the design?” The answer is that the design effort precedes to a depth that is sufficient to meet several needs: the design must penetrate sufficiently to allow analytical validation of the design to the requirements; it must also have sufficient depth to support cost modeling, and to convince a review team of a feasible design, with performance, cost and risk margins.

The systems engineering process is applied again and again as the system is developed. As the system is realized, the issues addressed evolve and the particulars of the activity change. Most of the major system decisions (goals, architecture, acceptable life-cycle cost, etc.) are made during the early phases of the project, so the successive refinements do not correspond precisely to the phases of the system life cycle. Much of the system architecture can be “seen” even at the outset, so the successive refinements do not correspond exactly to development of the architectural hierarchy, either. Rather, they correspond to the successively greater resolution by which the system is defined.

It is reasonable to expect the system to be defined with increasingly better resolution as time passes. This tendency is formalized at some point (in Phase B) by defining a *baseline* system definition. Usually, the goals, objectives, and constraints are baselined as the *requirements* portion of the baseline. The entire baseline is then subjected to configuration control in an attempt to ensure that any subsequent changes are indeed justified and affordable.

At this point in the systems engineering process, there is a logical branch point. For those issues for which the process of successive refinement has proceeded far enough, the next step is to implement the decisions at that level of resolution (that is, unwind the recursive process). For those issues that are still insufficiently resolved, the next step is to refine the development further.

### **Fully Describe the Design Solution**

Once the preferred design alternative has been selected and the proper level of refinement has been completed then the design is fully defined into a final design solution that will satisfy the technical requirements. This output of the design solution definition process is typically called a *technical data package*. The design solution definition will be used to generate the end product specifications which will be used to produce the product and to conduct product verification. This process may be further refined depending on whether there are additional subsystems of the end product that need to be defined.

The scope and content of the full design description must be appropriate for the product life-cycle phase, the phase exit criteria and the product position in the product breakdown structure

(system structure). Depending on these factors the form of the design solution definition could be simply a simulation model or a paper study report. The technical data package evolves from phase to phase, starting with conceptual sketches or models and ending with complete drawings, parts list and other details needed for product implementation or product integration. Typical output definitions from the design solution process are shown in Figure 4-16 and are described in section 4.4.1.3.

### **Verify the Design Solution**

Once an acceptable design solution has been selected from among the various alternative designs and once it has been documented in a technical data package, the design solution must next be verified against the system requirements and constraints. The best way to achieve this verification is by means of a peer review to evaluate the resulting design solution definition. Guidelines for conducting a peer review are discussed in section 6.7. Typically a peer review is a review of the technical data package by cognizant engineers not directly associated with the program or project, but knowledgeable in the design and development of similar systems. The peer review should be conducted according to established criteria. Such criteria will reflect the life-cycle phase exit criteria and the location of the design within the system product breakdown structure. Any design anomalies or review action items must be recorded. The goal is to resolve any action items or anomalies prior to next higher level technical or programmatic review. Any action item or anomaly that cannot be resolved in a timely fashion must be reported at the next higher review and carried as an action item from that review.

As such, peer reviews not only verify that the system meets all of its requirements but also play a significant role as the detailed technical component of higher level technical and programmatic reviews. For example, the peer review of a component battery design can go into much more technical detail on the battery than the integrated power subsystem review. Peer reviews can cover the components of a subsystem down to the level appropriate for verifying the design against the requirements. Concerns raised at the peer review might have implications on the power subsystem design and verification and therefore must be reported at the next higher level review of the power subsystem.

The verification must show that the design solution definition:

- is realizable within the constraints imposed on the technical effort;
- has specified requirements that are stated in acceptable statements and have bidirectional traceability with the derived technical requirements, technical requirements and stakeholder expectations; and
- has decisions and assumptions made in forming the solution consistent with its set of derived technical requirements, separately allocated technical requirements and identified system product and service constraints.

This design solution verification is in contrast to the verification of the end product described in the end product verification plan which is part of the technical data package. That verification

occurs in a later life-cycle phase and is a result of the product verification evaluation process (see section 5.3) applied to the realization of the design solution as an end product.

### **Validate the Design Solution**

The validation of the design solution is a recursive and iterative process as described in the System Design Process figure (Figure 4-1). Each alternate design concept is validated against the set of stakeholder expectations. The stakeholder expectations drive the iterative design loop in which a strawman architecture/design, the ConOps, and the derived requirements are developed. These three products must be consistent with each other and will require iterations and design decisions to achieve this consistency. Once consistency is achieved, functional analyses allow the study team to validate the design against the stakeholder expectations. A simplified validation asks the questions, does the system work? Is the system safe and reliable? Is the system affordable? If the answer to any of these questions is no, then changes to the design or stakeholder expectations will be required and the process is started over again. This process continues until the system—architecture, ConOps and requirements—meet the stakeholder expectations.

This design solution validation is in contrast to the validation of the end product described in the end product validation plan which is part of the technical data package. That validation occurs in a later life-cycle phase and is a result of the product validation evaluation process (see section 5.4) applied to the realization of the design solution as an end product.

### **Identify Enabling Products**

Enabling products are the life-cycle support products and services (e.g., production, test, deployment, training, maintenance, and disposal) that facilitate the progression and use of the operational end product through its life cycle. Since the end product and its enabling products are independent, they are viewed as a system. Project responsibility thus extends to responsibility for acquiring services from the relevant enabling products in each life-cycle phase. When a suitable enabling product does not already exist, the project that is responsible for the end product can also be responsible for creating and using the enabling product.

Therefore an important activity in the design solution definition process is the identification of the enabling products that will be required during the life cycle of the selected design solution and then initiating the acquisition or development of those enabling products. Need dates for the enabling products must be realistically identified on the project schedules incorporating appropriate schedule slack. Then firm commitments in the form of contracts, agreements, and/or operational plans must be put in place to ensure that the enabling products will be available when needed to support the product-line life-cycle phase activities. The enabling product requirements are documented as part of the technical data package for the design solution definition process.

An environmental test chamber would be an example of an enabling product whose use would be acquired at an appropriate time during the test phase of a space flight system.

Special test fixtures or special mechanical handling devices would be examples of enabling products that would have to be created by the project. Because of long development times as

well as oversubscribed facilities, it is important to identify enabling products and secure the commitments as early in the design phase as possible.

### **Baseline the Design Solution**

As shown earlier in Figure 4-1, once the selected system design solution meets the stakeholder expectations, the study team baselines the products and prepares for the next life-cycle phase. Because of the recursive nature of successive refinement, intermediate levels of decomposition are often validated and baselined as part of the process. In the next level of decomposition, the baselined requirements become the set of high level requirements for the decomposed elements, and the process begins again.

By baselining a particular design solution enables the technical team to focus on one design out of all the alternative design concepts. This is a critical point in the design process. It puts a stake in the ground and gets everyone on the design team focused on the same concept. When dealing with such complex systems it is difficult for an individual member of the team to design his portion of the system if the system design is a moving target. The baselined design is documented and placed under configuration control. This includes the system requirements, specifications and configuration descriptions.

While baselining a design is beneficial to the design process, there is a danger if it is exercised too early in the design solution definition process. The early exploration of alternative designs should be free and open to a wide range of ideas, concepts and implementations. Baselining too early takes the inventive nature out of the concept exploration. Therefore baselining should be one of the last steps in the design solution definition process.

#### *4.4.1.3 Outputs*

Outputs of the design solution definition process are the specifications and plans that are passed on to the product realization processes. They contain the “design-to”, “build-to” and “code-to” documentation that complies with the approved baseline for the system.

As mentioned earlier, the scope and content of the full design description must be appropriate for the product life-cycle phase, the phase exit criteria and the product position in the product breakdown structure.

- The system specification contains the “design-to” baseline for the system that is the result of the Design Solution Definition Process. The system design specification provides sufficient guidance, constraints, and system requirements for the design engineers to execute the design.
- The system external interface specifications describe the “design-to” baseline for the behavior and characteristics of all physical interfaces that the system has with the external world. This includes all structural, thermal, electrical and signal interfaces as well as the human-system interfaces.
- The end product specifications contain the detailed “build-to” and “code-to” requirements for the end product. They are detailed, exact statements of design particulars, such as

statements prescribing materials, dimensions, and quality of work in order to build, install, or manufacture the end product.

- The end product interface specifications contain the detailed “build-to” and “code-to” requirements for the behavior and characteristics of all physical interfaces that the end product has with external elements including the human-system interfaces.
- The end product subsystem initial specifications, if subsystems are required.
- The requirements for associated supporting enabling products. Enabling products are the life-cycle support products and services that facilitate the progression and use of the operational end product through its life cycle. They are viewed as part of the system since the end product and its enabling products are interdependent.
- The end product verification plan provides the content and depth of detail necessary to provide full visibility of all verification activities for the end product. Depending on the scope of the end product, the plan encompasses qualification, acceptance, pre-launch, operational, and disposal verification activities for flight hardware and software.
- The end product validation plan provides the content and depth of detail necessary to provide full visibility of all activities to validate the realized product against the baselined stakeholder expectations. The plan identifies the type of validation, the validation procedures and the validation environment that are appropriate to confirm that the realized end product conforms to stakeholder requirements.
- The applicable logistics and operate-to procedures for the system describing, for example, guidelines on such things as handling, transportation, maintenance, long-term storage, and operational considerations for the particular design solution.

## **4.4.2 Design Solution Definition Guidance**

### *4.4.2.1 Technology Assessment*

As mentioned in the process description (section 4.4.1), the creation of alternate design solutions involves an assessment of potential capabilities offered by the continually changing state of technology. A continual interaction between the technology development process and the design process ensures that the design reflects the realities of the available technology. This interaction is facilitated through a periodic assessment of the design with respect to the maturity of the technology required to implement the design.

After identifying the technology gaps existing in a given design concept it will frequently be necessary to undertake technology development in order to ascertain viability. Given that resources will always be limited it will be necessary to pursue only the most promising technologies that are required to enable a given concept.

If requirements are defined without fully understanding the resources required to accomplish needed technology developments then the program/project is at risk. Technology assessment must be done iteratively until requirements and available resources are aligned within an

acceptable risk posture. Technology development plays a far greater role in the life cycle of a program/project than has been traditionally considered and it is the role of the systems engineer to develop an understanding of the extent of program/project impacts—maximizing benefits and minimizing adverse effects. Traditionally, from a program/project perspective, technology development has been associated with the development and incorporation of any “new” technology necessary to meet requirements. However, a frequently overlooked area is that associated with the modification of “heritage” systems incorporated into different architectures and operating in different environments from the ones for which they were designed. If the required modifications and/or operating environments fall outside of the realm of experience then this too should be considered technology development.

To understand whether or not technology development is required—and to subsequently quantify the associated cost, schedule and risk—it is necessary to systematically assess the maturity of each system, subsystem, or component in terms of the architecture and operational environment. It is then necessary to assess what is required in the way of development to advance the maturity to a point where it can be successfully incorporated within cost, schedule and performance constraints. A process for accomplishing this assessment is described in appendix 4-4. Because technology development has the potential for such significant impacts on a program/project, technology assessment needs to play a role throughout the design and development process from concept development through PDR. Lessons learned from a technology development point-of-view should then be captured in the final phase of the program.

#### *4.4.2.2 Integrating Engineering Specialties into the Systems Engineering Process*

As part of the technical effort, specialty engineers in cooperation with systems engineering and subsystem designers often perform tasks that are common across disciplines. Foremost, they apply specialized analytical techniques to create information needed by the project manager and systems engineer. They also help define and write system requirements in their areas of expertise, and they review data packages, engineering change requests test results, and documentation for major project reviews. The project manager and/or systems engineer need to ensure that the information and products so generated add value to the project commensurate with their cost. The specialty engineering technical effort should be well integrated into the Project. The roles and responsibilities of the specialty engineering disciplines should be summarized in the SEMP.

The specialty engineering disciplines included in this handbook are safety and reliability, quality assurance, integrated logistics support, maintainability, producibility and human factors. An overview of these specialty engineering disciplines is provided to give systems engineers a brief introduction. It is not intended to be a handbook for any of these discipline specialties.

### **Safety and Reliability**

#### *Overview and Purpose*

A reliable system ensures mission success by functioning properly over its intended life. It has a low and acceptable probability of failure, achieved through simplicity, proper design, and proper application of reliable parts and materials. In addition to long life, a reliable system is robust and

fault tolerant, meaning it can tolerate failures and variations in its operating parameters and environments.

### *Safety and Reliability in the System Design Process*

A focus on Safety and Reliability throughout the mission life cycle is essential for ensuring mission success. The fidelity to which safety and reliability are designed and built into the system depends on the information needed and the type of mission. For human rated systems, safety and reliability is the primary objective throughout the design process. For science missions, safety and reliability should be commensurate with the funding and level of risk a program or project is willing to accept. Regardless of the type of mission safety and reliability considerations must be an intricate part of the system design processes.

To realize the maximum benefit from reliability analysis, it is essential to integrate the risk and reliability analysts within the design teams. The importance of this cannot be overstated. In many cases the reliability and risk analysts perform the analysis on the design after it has been formulated. In this case, safety and reliability features are added on or out-sourced rather than designed in. This results in unrealistic analysis that is not focused on risk drivers and does not provide value to the design.

Risk and reliability analyses evolve to answer key questions about design trades as the design matures. Reliability analyses lead the growth of information about the system and provide guidance for system risk drivers. MIL STD 882, NASA-STD-8729, "Planning, Developing, and Maintaining an Effective Reliability and Maintainability Program" outlines engineering activities that should be tailored for each specific project. The concept is to choose an effective set of reliability and maintainability engineering activities to ensure that the systems designed, built, and deployed will operate successfully for the required mission life cycle.

In the early phases of a project, risk and reliability analysis help designers understand the interrelationships of requirements, constraints, and resources, and uncover key relationships and drivers so they can be properly considered. The analyst must help designers go beyond the requirements to understand implicit dependencies that emerge as the design concept matures. It is unrealistic to assume that design requirements will correctly capture all risk and reliability issues and "force" a reliable design. System impacts of designs must play a key role in the design. Making designers aware of impacts of their decisions on overall mission reliability is key.

As the design matures, preliminary reliability analysis occurs using established techniques. The design and concept of operations should be thoroughly examined for accident initiators and hazards that could lead to mishaps. Conservative estimates of likelihood and consequences of the hazards can be used as a basis for applying design resources to reduce the risk of failures. The team should also ensure that the goals can be met and failure modes are considered and take into account the entire system. Preliminary reliability analysis should also occur.

During the latter phases of a project, the team uses risk assessments and reliability techniques to verify that the design is meeting its risk and reliability goals and help develop mitigation strategies when the goals are not met or discrepancies/failures occur.

### *Analyses Techniques and Methods*

This section provides a brief summary of the types of analysis techniques and methods.

1. Event Sequence Diagrams/Event Trees are models that describe the sequence of events and responses to off-nominal conditions that can occur during a mission.
2. FMEAs are a bottom up analysis that identify the types of failures that can occur within a system and identify the causes, effects, and mitigating strategies that can be employed to control the effects of the failure.
3. Qualitative Top Down Logic Models identify how failures within a system can combine to cause an undesired event.
4. Quantitative Logic Models (Probabilistic Risk Assessment) extend the qualitative models to include the likelihood of failure. These models involve developing failure criteria based on system physics and system success criteria, and employing statistical techniques to estimate the likelihood of failure along with uncertainty.
5. Reliability Block Diagrams are diagrams of the elements to evaluate the reliability of a system to provide a function.
6. Preliminary Hazard Analysis (PHA) is performed early based on the functions performed during the mission. PHA is a “What if” process that considers the potential hazard, initiating event scenarios, effect, and potential corrective measures and controls. The objective is to determine if the hazard can be eliminated, and if not, how it can be controlled.
7. Hazard Analysis (HA) evaluates the completed design. HA is a “What if” process that considers the potential hazard, initiating event, effect, and potential corrective measures and controls. The objective is to determine if the hazard can be eliminated, and if not, how it can be controlled.
8. Human Reliability Analysis is a method to understand how human failures can lead to system failure and estimate the likelihood of those failures.
9. Probabilistic Structural Analysis provides a way to combine uncertainties in materials and loads to evaluate the failure of a structural element.
10. Sparing/Logistics Models provide a means to estimate the interactions of systems in time. These models include ground-processing simulations and mission campaign simulations.

### *Limitations on Reliability Analysis*

The engineering design team must understand that reliability is expressed as the *probability* of mission success. Probability is a mathematical measure expressing the likelihood of occurrence of a specific event. Therefore probability should be based on engineering and historical data and includes uncertainty.

The confidence level is a mathematical expression of the degree of confidence that the analysts have that their estimates are correct. The level of uncertainty improves as the quality of data improves. The initial estimates of failure rates or failure probability might be based on comparison to similar equipment, historical data (heritage), failure rate data from handbooks, or expert elicitation.

In summary,

1. Reliability estimates express probability of success.
2. Uncertainty should be included with reliability estimates.
3. Reliability estimates combined with FMEA provide additional and valuable information to aid in the decisionmaking process.

## **Quality Assurance**

Even with the best designs, hardware fabrication and testing are subject to human error. The systems engineer needs to have some confidence that the system actually produced and delivered is in accordance with its functional, performance, and design requirements. Quality Assurance (QA) provides an independent assessment to the project manager/systems engineer of the items produced and processes used during the project life cycle. The project manager/systems engineer must work with the quality assurance engineer to develop a quality assurance program (the extent, responsibility, and timing of QA activities) tailored to the project it supports.

QA is the mainstay of quality as practiced at NASA. NPD 8730.5, NASA Quality Assurance Program Policy, states that NASA's policy is "to comply with prescribed requirements for performance of work and to provide for independent assurance of compliance through implementation of a quality assurance program." The Quality function of Safety and Mission Assurance (SMA) ensures that both contractors and other NASA functions do what they say they will do and say what they intend to do. This ensures end product and program quality, reliability and overall risk are at the level planned.

### *The Systems Engineer's Relationship to QA*

As with reliability, producibility, and other characteristics, quality must be designed as an integral part of any system. It is important that the systems engineer understand Safety and Mission Assurance's safeguarding role in the broad context of total risk and supports the quality role explicitly and vigorously. All of this is easier if the SMA Quality function is actively included, and if quality is designed in with buy-in by all roles, from the beginning.

Quality is a vital part of risk management. Errors, variability, escapes and other problems cost time, program resources, taxpayer dollars, and even lives. It is incumbent on the systems engineer to know how quality affects their projects and encourage best practices to achieve the quality level.

Rigid adherence to procedural requirements is necessary in high-risk, low volume manufacturing. In the absence of large samples and long production runs, compliance to these

written procedures is a strong step toward ensuring process, and thereby product, consistency. To address this, NASA requires quality assurance programs to be designed in a manner that mitigates risks associated with noncompliance to those requirements.

There will be a large number of requirements and procedures thus created. These must be flowed down to the supply chain, even to lowest tier suppliers. For circumstances where noncompliance can result in loss of life or loss of mission, there is a requirement to insert into procedures Government Mandatory Inspection Points (GMIPs) to ensure 100 percent compliance with safety/mission critical attributes. Safety/mission critical attributes include hardware characteristics, manufacturing process requirements, operating conditions, and functional performance criteria that, if not met, can result in loss of life or loss of mission. There will be in place a Program/Project Quality Assurance Surveillance Plan (PQASP) as mandated by Federal Acquisition Requirement (FAR) Subpart 46.4. Preparation and content for PQASPs are outlined in NPR 8735.2A, *Management of Government Quality Assurance Functions for NASA Contracts*. This document covers quality assurance requirements for both low risk and high-risk acquisitions and includes functions such as document review, product examination, process witnessing, quality system evaluation, nonconformance reporting and corrective action, planning for quality assurance and surveillance, and GMIPs. In addition, most NASA projects are required to adhere to either ISO 9001 (non critical work) or AS9100 (critical work) requirements for management of quality systems. Training in these systems is mandatory for most NASA functions, so knowledge of their applicability by the systems engineer is assumed. Their texts and intent are strongly reflected in NASA's quality procedural documents.

### **Integrated Logistics Support**

The objective of Integrated Logistics Support (ILS) activities within the systems engineering process is to ensure that the product system is supported during development (Phase D) and operations (Phase E) in a cost-effective manner. This is primarily accomplished by early, concurrent consideration of supportability characteristics, performing trade studies on alternative system and ILS concepts, quantifying resource requirements for each ILS element using best-practice techniques, and acquiring the support items associated with each ILS element. During operations, ILS activities support the system while seeking improvements in its cost-effectiveness by conducting analyses in response to actual operational conditions. These analyses continually reshape the ILS system and its resources requirements. Neglecting ILS or poor ILS decisions invariably have adverse effects on the life-cycle cost of the resultant system. Table 4-4 below summarizes the key ILS elements.

**Table 4-4 Key ILS Elements**

Element	Definition
Maintenance	The process of planning and executing life-cycle repair/services concepts and requirements necessary to ensure sustained operation of the system.
Design Interface	The interaction and relationship of logistics with the systems engineering process to ensure that supportability influences the definition and design of the system so as to reduce life-cycle cost
Technical Data	The recorded scientific, engineering, technical, and cost information used to define, produce, test, evaluate, modify, deliver, support, and operate the system
Training	The processes, procedures, devices, and equipment required to train personnel to operate and support the system
Supply Support	Actions required to provide all the necessary material to ensure the system's supportability and usability objectives are met
Test and Support Equipment	The equipment required to facilitate development, production, and operation of the system
Transportation and Handling:	The actions, re-sources, and methods necessary to ensure the proper and safe movement, handling, packaging, and storage of system items and materials
Human Resources and Personnel Planning	Actions required to determine the best skills-mix, considering current and future operator, maintenance, engineering, and administrative personnel costs
System Facilities	Real property assets required to develop and operate a system

ILS planning should begin early in the project life cycle, and should be documented. This plan should address the nine elements above including how they will be considered, conducted and integrated into the systems engineering process needs.

### **Maintainability**

Maintainability is defined as the measure of the ability of an item to be retained in or restored to specified conditions when maintenance is performed by personnel having specified skill levels, using prescribed procedures and resources, at each prescribed level of maintenance. It is the inherent characteristics of a design or installation that contribute to the ease, economy, safety, and accuracy with which maintenance actions can be performed.

#### *Role of the Maintainability Engineer*

Maintainability engineering is another major specialty discipline that contributes to the goal of a supportable system. This is primarily accomplished in the systems engineering process through an active role in implementing specific design features to facilitate safe and effective maintenance actions in the predicted physical environments, and through a central role in developing the integrated logistics support (ILS) system. Example tasks of the maintainability engineer include: Developing and maintaining a system maintenance concept; establishing and allocating maintainability requirements; performing analysis to quantify the system's maintenance resource requirements; and verifying the systems maintainability requirements.

## **Producibility**

Producibility is a system characteristic associated with the ease and economy with which a completed design can be transformed (i.e., fabricated, manufactured, or coded) into a hardware and/or software realization. While major NASA systems tend to be produced in small quantities, a particular producibility feature can be critical to a system's cost-effectiveness, as experience with the Shuttle's thermal tiles has shown.

### *Role of the Production Engineer*

The production engineer supports the systems engineering process (as a part of the multi-disciplinary PDT) through an active role in implementing specific design features to enhance producibility, and by performing the production engineering analyses needed by the project. These tasks and analyses include:

1. Performing the manufacturing/fabrication portion of the system risk management program. This is accomplished by conducting a rigorous production risk assessment and by planning effective risk mitigation actions.
2. Identifying system design features that enhance producibility. Efforts usually focus on design simplification, fabrication tolerances, and avoidance of hazardous materials.
3. Conducting producibility trade studies to determine the most cost-effective fabrication/manufacturing process.
4. Assessing production feasibility within project constraints. This may include assessing contractor and principal subcontractor production experience and capability, new fabrication technology, special tooling, and production personnel training requirements. Identifying long-lead items and critical materials.
5. Estimating production costs as a part of life-cycle cost management.
6. Supporting Technology Readiness Assessments.
7. Developing production schedules.
8. Developing approaches and plans to validate fabrication/manufacturing processes. The results of these tasks and production engineering analyses are documented in the Manufacturing Plan with a level of detail appropriate to the phase of the project. The production engineer also participates in and contributes to major project reviews (primarily PDR and CDR) on the above items, and to special interim reviews such as the Production Readiness Review (PRR).

## Prototypes

Experience has shown that prototype systems can be effective in enabling efficient producibility even when building only a single flight system. Prototypes are built early in the life cycle and they are made as close to the flight item in form, fit and function as is feasible at that stage of the development. The prototype is used to “wring out” the design solution so that experience gained from the prototype can be fed back into design changes that will improve the manufacture, integration and maintainability of a single flight item or the production run of several flight items. Unfortunately prototypes are often deleted from projects to save cost. Along with that decision the project accepts an increased risk in the development phase of the life cycle. Fortunately advancements in computer aided design and manufacturing has mitigated that risk somewhat by enabling the designer to visualize the design and “walk through” the integration sequence to uncover problems before they become a costly reality.

## Human Factors Engineering

### *Overview and Purpose*

Human operators and maintainers of systems are critical considerations in the design process. Human factors engineering is the discipline that studies the human-system interfaces and provides requirements, standards and guidelines to ensure the human component of the integrated system is able to function as intended. Human roles include operator, maintainer and passenger.

Human factors are generally considered in four categories. The first is anthropometry and biomechanics—the physical size, shape, and strength of the humans. The second is sensation and perception—primarily vision and hearing, but senses such as touch are also important. The environment is a third factor—ambient noise and lighting, vibration, temperature and humidity, atmospheric composition and contaminants. Psychological factors comprise memory; information processing components such as pattern recognition, decision making and signal detection; and affective factors—e.g., emotions, cultural patterns, and habits.

### *Human Factors Engineering in the System Design Process*

**Stakeholder Expectations:** The operators, maintainers and passengers are all stakeholders in the system. The human factors engineer (HFE) identifies roles and responsibilities that can be performed by humans, and scenarios that exceed human capabilities. The HFE ensures that system operational concept development includes task analysis and human/system function allocation. As these are refined, function allocation distributes operator roles and responsibilities for subtasks to the crew, external support teams and automation. (For example, in aviation, tasks may be allocated to crew, air traffic controllers, or autopilots. In spacecraft, tasks may be performed by crew, mission control, or onboard systems.)

**Requirements Definition:** Human factors requirements are program/project dependent, derived from NASA-STD-3001 Space Flight Human Systems Standards, Volumes 1 and 2.

**Technical Solution—Logical Decomposition and Physical Solution:** Consider the human as a central component when developing design concepts. The users—operators or maintainers—will not see the entire system as the designer does, only as the system interfaces with them. In engineering design reviews, HFEs promote the usability of the design solution. With early involvement, human factors assessments may catch usability problems at very early stages. For

example, in one ISS payload design project, a human factors assessment of a very early PowerPoint block diagram of the layout of stowage and hardware identified problems that would have made operations very difficult. Changes were made to the conceptual design at negligible cost—i.e., rearranging conceptual block diagrams based on the sequence in which users would access items.

Usability evaluations of design concepts: Can be performed easily using rapid prototyping tools for hardware and software interfaces, standard human factors engineering data-gathering and analysis tools, and metrics such as task completion time and number of errors. Systematically collected subjective reports from operators also provide useful data. New technologies provide detailed objective information—e.g., eye tracking for display and control layout assessment. HFEs provide assessment capabilities throughout the iterative design process.

Verification and Validation: As mentioned, verification of requirements for usability, error rates, task completion times and workload is challenging. Methods range from tests with trained personnel in mockups and simulators, to models of human performance, to inspection by experts. As members of the systems engineering team, HFEs provide verification guidance from the time requirements are first developed.

#### *Human Factors Engineering Analyses Techniques and Methods*

The principle methods used to analyze systems, provide human performance data, predict human-system performance, and evaluate human-system designs are:

1. **Task Analysis:** Produces a detailed description of the things a person must do in a system to accomplish a task, with emphasis on requirements for information presentation, decisions to be made, task times, operator actions, and environmental conditions.
2. **Timeline Analysis:** Follows from task analysis. Durations of tasks are identified in task analyses, and the times at which these tasks occur are plotted in graphs, which also show the task sequences. The purpose is to identify requirements for simultaneous incompatible activities and activities which take longer than is available. Timelines for a given task can describe the activities of multiple operators or crew members.
3. **Simulation:** Models or mockups are used to make predictions about system performance, compare configurations, evaluate procedures, and evaluate alternatives. Simulations can be as simple as positioning a graphical human model with realistic anthropometric dimensions with a graphical model of an operator station, or they can be complex stochastic models capturing decision points, error opportunities, etc.
4. **Usability Testing:** Based on a task analysis and preliminary design, realistic tasks are carried out in a controlled environment with monitoring and recording equipment. Objective measures such as performance time and number of errors are evaluated; subjective ratings are collected. The outputs systematically report on strengths and weaknesses of candidate design solutions.
5. **Workload Assessment:** Measurement using a standardized scale such as the NASA-TLX or the Cooper-Harper rating scales of the amount and type of work. It assesses operator

and crew task loading, which determines the ability of the human to perform the required tasks in the desired time with the desired accuracy.

### *Roles of the Human Factors Engineer*

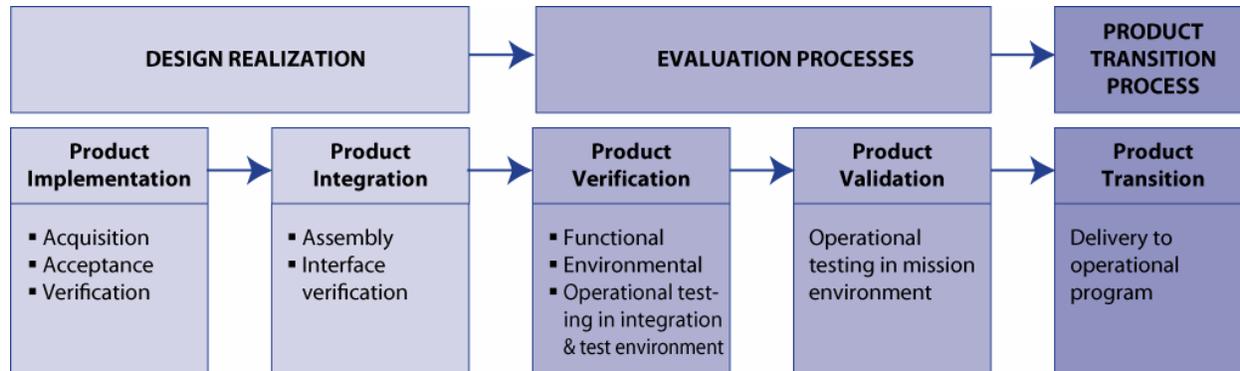
The HFE supports the systems engineering process by representing the users' and maintainers' requirements and capabilities throughout the design, production and operations stages. HFEs roles include:

1. Identify applicable requirements based on Agency standards for human-system integration during the requirements definition phase.
2. Support development of mission concepts by providing information on human performance capabilities and limitations.
3. Support task analysis and function allocation with information on human capabilities and limitations.
4. Identify system design features that enhance usability. This integrates knowledge of human performance capabilities and design features.
5. Support trade studies by providing data on effects of alternative designs on time to complete tasks, workload, and error rates.
6. Support trade studies by providing data on effects of alternative designs on skills and training required to operate the system.
7. Support design reviews to ensure compliance with human-systems integration requirements.
8. Conduct evaluations using mockups and prototypes to provide detailed data on user performance.
9. Support development of training and maintenance procedures in conjunction with hardware designers and mission planners.
10. Collect data on human-system integration issues during operations to inform future designs.

## 5.0 Product Realization

The product realization side of the SE engine is where the rubber meets the road. In this portion of the engine, five interdependent processes result in systems that meet the design specifications and stakeholder expectations. These products are produced, acquired, reused or coded; integrated into higher level assemblies; verified against design specifications; validated against stakeholder expectations; and transitioned to the next level of the system. As has been mentioned in previous sections, products can be models and simulations, paper studies or proposals, or hardware and software. The type and level of product depends on the phase of the life cycle and the products' specific objectives. But whatever the product, all must effectively use the processes to ensure the system meets the intended operational concept.

This effort starts with the technical team taking the output from the system design processes and using the appropriate crosscutting functions, such as data and configuration management, and technical assessments to make, buy, or reuse subsystems. Once these subsystems are realized, they must be integrated to the appropriate level as designated by the appropriate interface requirements. These products are then verified through the technical assessment process to ensure they are consistent with the technical data package and “the right product was built.” Once consistency is achieved, the technical team will validate the products against the Stakeholder Expectations or was the “right product built.” Upon successful completion of validation, the products are transitioned to the next level of the system. Figure 5-1 illustrates these processes.



**Figure 5-1 Product Realization**

This is an iterative and recursive process. Early in the life-cycle paper products, models and simulations are run through the five realization processes. As the system matures and progress through the life cycle, hardware and software products are run through these processes. It is important to catch errors and failures at the lowest level of integration and early in the life cycle so that changes can be made through the design processes with minimum impact to the project.

The next sections describe each of the five product realization processes and their associated products for a given NASA mission.

### Product Realization Keys

- Generate and manage requirements for Off-The-Shelf hardware/software products as for all other products.
- Understand the differences between Verification Testing and Validation Testing

**Verification Testing:** Verification testing relates back to the approved requirements set (such as a Systems Requirements Document) and can be performed at different stages in the product life cycle. Verification Testing includes: (1) any testing used to assist in the development and maturation of products, product elements, or manufacturing or support processes; and/or (2) any engineering-type test used to verify status of technical progress, to verify that design risks are minimized, to substantiate achievement of contract technical performance, and to certify readiness for initial validation testing. Verification tests use instrumentation and measurements, and are generally accomplished by engineers, technicians, or operator-maintainer test personnel in a controlled environment to facilitate failure analysis.

**Validation Testing:** Validation relates back to the Concept of Operations document. Validation testing is conducted under realistic conditions (or simulated conditions) on any end product for the purpose of determining the effectiveness and suitability of the product for use in mission operations by typical users; and the evaluation of the results of such tests. Testing is the detailed quantifying method of both verification and validation. However, testing is required to validate final end products to be destroyed.

- Consider all customer, technical, programmatic, and safety requirements when evaluating the input necessary to achieve a successful Product Transition.
- Analyze for any potential incompatibilities with interfaces as early as possible
- Completely understand and analyze all test data for trends and anomalies
- Understand the limitations of the testing and any assumptions that are made

## 5.1 Product Implementation

Product implementation is the first process encountered in the SE engine that begins the movement from the bottom of the product hierarchy up towards the Product Transition Process. This is where the plans, designs, analysis, requirement development, and drawings are realized into actual products.

Product implementation is used to generate a specified product of a project or activity through buying, making/coding, or reusing previously developed hardware, software, models or studies to generate a product appropriate for the phase of the life cycle. The product must satisfy the design solution and its specified requirements.

The product implementation process is the key activity that moves the project from plans and designs into realized products. Depending on the project and life-cycle phase within the project, the product may be hardware, software, a model, simulations, mockups, study reports or other tangible results. These products may be realized through their purchase from commercial or other vendors, generated from scratch, or through partial or complete reuse of products from other projects or activities. The decision as to which or combination of these realization strategies will be used for the products of this project will have been made early in the life cycle using the Decision Analysis process.

### 5.1.1 Process Description

Figure 5-2 provides a typical flow diagram for the Product Implementation Process and identifies typical inputs, outputs, and activities to consider in addressing product implementation.

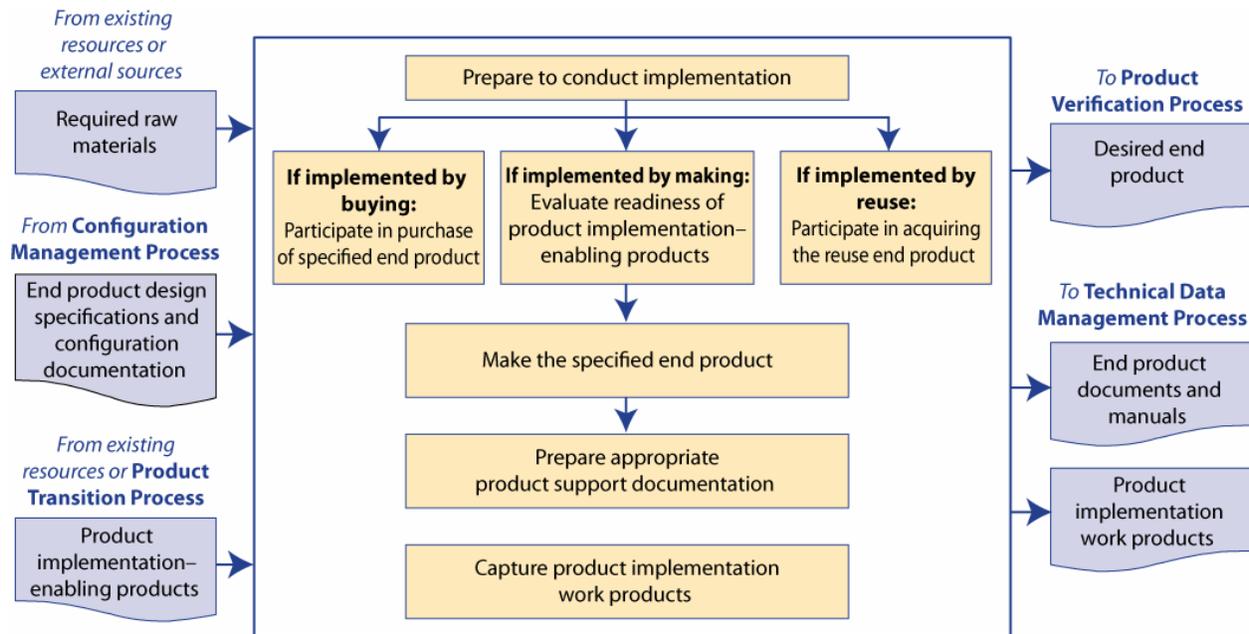


Figure 5-2 Product Implementation Process

### 5.1.1.1 Inputs

Inputs to the Product Implementation activity depend primarily on the decision as to whether the end product will be purchased, developed from scratch, or if the product will be formed by reusing part or all of products from other projects. Typical inputs are shown in Figure 5-2.

- **Inputs if Purchasing the End Product:** If the decision was made to purchase part or all of the products for this project, the end product design specifications are obtained from the configuration management system as well as other applicable documents such as the SEMP.
- **Inputs if Making/Coding the End Product:** For end products that will be made/coded by the technical team, the inputs will be the configuration controlled design specifications and raw materials as provided to or purchased by the project.
- **Inputs needed if Reusing an End Product:** For end products that will reuse part or all of products generated by other projects, the inputs may be the documentation associated with the product, as well as the product itself. Care must be taken to ensure that these products will indeed meet the specifications and environments for this project. These would have been factors involved in the decision analysis process to determine the make/buy/reuse decision.

### 5.1.1.2 Process Activities

Implementing the product can take one of three forms:

- Purchase/buy.
- Make/Code
- Reuse

These three forms will be discussed in the following sections. Figure 5-2 shows what kind of inputs, outputs, and activities that are performed during product implementation regardless of where in the product hierarchy or life cycle it is. These activities include preparing to conduct the implementation, purchasing/making/reusing the product, and capturing the product implementation work product. In some cases, implementing a product may have aspects of more than one of these forms (such as a build-to-print). In those cases, the appropriate aspects of the applicable forms are used.

#### **Prepare to Conduct Implementation**

Preparing to conduct the product implementation is a key first step regardless of what form of implementation has been selected. For complex projects, implementation strategy and detailed planning or procedures needs to be developed and documented. For less complex projects, the implementation strategy and planning will need to be discussed, approved and documented as appropriate for the complexity of the project.

The documentation, specifications, and other inputs will also need to be reviewed to ensure they are ready and at an appropriate level of detail to adequately complete the type of implementation

form being employed and for the product life-cycle phase. For example, if the “make” implementation form is being employed, the design specifications will need to be reviewed to ensure they are at a “design-to” level that will allow the product to be developed. If the product is to be bought as a pure COTS item, the specifications will need to be checked to make sure they adequately describe the vendor characteristics to narrow to a single make/model of their product line.

Finally the availability and skills of personnel needed to conduct the implementation as well as the availability of any necessary raw materials, enabling products, or special services should also be reviewed. Any special training necessary for the personnel to perform their tasks needs to be performed by this time.

## **Purchase, Make, or Reuse the Product**

### *Purchase the Product*

In the first case, the end product is to be purchased from a commercial or other vendor. Design/purchase specifications will have been generated from the Requirements Development process and provided as inputs. The technical team will need to review these specifications and ensure they are in a form adequate for the contract or purchase order. This may include the generation of contracts, Statements of Work, Requests for Proposals, purchase orders or other purchasing mechanisms. The responsibilities of the government and contractor team should have been documented in the SEMP. This will define, for example, whether the government expects the vendor to provide a fully verified and validated product, or whether the government team will be performing those duties. The team will need to work with the acquisition team to ensure the accuracy of the contract Statement of Work or purchase order, and to ensure that adequate documentation, certificates of compliance, or other specific needs are requested of the vendor.

For contracted purchases, as proposals come back from the vendors, the technical team should work with the contracting officer and participate in the review of the technical information and in the selection of the vendor that best meets the design requirements for acceptable cost and schedule.

As the purchased products arrive, the technical team should assist in the inspection of the delivered product and its accompanying documentation. The team should ensure that the requested product was indeed the one delivered, and that all necessary documentation, such as source code, operator manuals, certificates of compliance, safety information, or drawings have been received.

The technical team should also ensure that any enabling products necessary to provide test, operations, maintenance and disposal support for the product is also ready or provided as defined in the contract.

Depending on the strategy and roles/responsibilities of the vendor, as documented in the SEMP, a determination/analysis of the vendor’s verification and validation compliance may need to be reviewed. This may be done informally or formally as appropriate for the complexity of the product. For products that were verified and validated by the vendor, after ensuring that all work products from this phase have been captured, the product may be ready to enter the Transition

Process to be delivered to the next higher level or to its final end user. For products that will be verified/validated by the technical team, the product will be ready to be verified after ensuring that all work products for this phase have been captured.

#### *Make/Code the Product*

If the strategy is to make or code the product, the technical team should first ensure that the enabling products are ready to make/code the product. This may include ensuring all piece parts are available, drawings are complete and adequate, machines to cut the material are available, operators are trained and available, procedures/processes are ready and trained, software personnel are ready and available to generate code, test fixtures are developed and ready to hold products while being generated, software is ready to begin model generation, etc.

The product is then made or coded in accordance with the specified requirements, configuration documentation and applicable standards. Throughout this process the technical team should work with the quality organization to review, inspect, and discuss progress and status within the team and with higher levels of management as appropriate. Progress should be documented within the technical schedules. Peer Reviews, audits, unit testing, code inspections, simulation checkout and other techniques may be used to ensure the made or coded product is ready for the verification process.

#### *Reuse*

If the strategy is to reuse a product that already exists, care must be taken to ensure that the product is truly applicable to this project and for the intended uses and the environment in which it will be used. This should have been a factor used in the decision strategy to make/buy/reuse.

The documentation available from the reuse product should be reviewed by the technical team to become completely familiar with the product and to ensure it will meet the requirements in the intended environment. Any supporting manuals, drawing or other documentation available should also be gathered.

The availability of any supporting or enabling products or infrastructure needed to complete the fabrication, coding, testing, analysis, verification, validation or shipping of the product needs to be determined. If any of these products or services are lacking, they will need to be developed or arranged for before progressing to the next phase.

Special arrangements may need to be made or forms such as non-disclosure agreements may need to be acquired before the reuse product can be received.

#### **Capture Work Products**

Regardless of what implementation form was selected, all work products from the make/buy/reuse process should be captured including design drawings, design documentation, code listings, model descriptions, procedures used, operator manuals, maintenance manuals, or other documentation as appropriate.

### 5.1.1.3 *Outputs*

- **Delivered End Product for Verification:** Unless performed by the vendor the made/coded, bought or reused end product, in a form appropriate for the life-cycle phase is provided to the verification process. The form of the end product is a function of the life-cycle phase and the placement within the system structure (the form of the end product could be hardware, software; model, prototype, first article for test; or single operational article or multiple production article).
- **Delivered End Product Documents and Manuals:** Appropriate documentation is also delivered with the end product to the verification process and to the technical data management process. Documentation may include applicable design drawings, operation, user, maintenance, or training manuals, applicable baseline documents (configuration baseline, specifications, stakeholder expectations), certificates of compliance, or other vendor documentation.

Process is complete when the following activities have been accomplished.

- End product is fabricated, purchased or reuse modules acquired.
- End products are reviewed, checked and ready for verification
- Procedures, decisions, assumptions, anomalies, corrective actions, lessons learned, etc. resulting from the make/buy/reuse are recorded.

## 5.1.2 *Product Implementation Guidance*

### 5.1.2.1 *Buying Off-the-Shelf Products*

OTS products are hardware/software that has an existing heritage and usually originates from one of several sources which include commercial (COTS), military and NASA programs. Special care needs to be taken when purchasing OTS products for use in the space environment. Most OTS products were developed for use in the more benign environments of Earth and may not be suitable to endure the harsh space environments including vacuum, radiation, extreme temperature ranges, extreme lighting conditions, zero gravity, atomic oxygen, lack of convection cooling in zero-G, launch vibration or acceleration and shock loads.

When purchasing OTS products, requirements should still be generated and managed. A survey of available OTS is made and evaluated as to the extent they satisfy the requirements. Products that meet all the requirements are a good candidate for selection. If no product can be found to meet all the requirements, a trade study needs to be performed to determine whether the requirements can be relaxed or waived, the OTS can be modified to bring into compliance, or whether another option to build or reuse should be selected.

Several additional factors should be considered when selecting the OTS option:

- Heritage
- Critical or non-critical application

- Amount modification required and who it is performed by

**Heritage** refers to the original manufacturer's level of quality and reliability that is built into the parts which have been proven by (1) time in service, (2) number of units in service, (3) mean time between failure (MTBF) performance and (4) number of use cycles. High heritage products are from the original supplier that has maintained the great majority of the original service, design, performance and manufacturing characteristics. Low heritage products are those that (1) was not built by the original manufacturer, (2) does not have a significant history of test and usage, or (3) has had significant aspects of the original service, design, performance, and manufacturing characteristics altered. An important factor in assessing the heritage of a COTS product is to ensure that the use/application of the product is relevant to the application in which it is now intended. A product that has high heritage in a ground-based application could have a low heritage when placed in a space environment.

A critical **application** is one in which a failure could cause loss of life, vehicle or loss of mission. For use in a critical application several additional precautions should be taken including ensuring it will not be used near the boundaries of its performance or environmental envelopes. Extra scrutiny by experts should be conducted during PDR and CDRs to ensure the appropriateness of its use.

**Modification** of OTS products may be required for it to be suitable for a NASA application. This affects its heritage and therefore the modified product should be treated as a new design. If the product is modified by NASA and not the manufacturer, it would be beneficial for the supplier to have some involvement in reviewing the modification. NASA modification may also require the purchase of additional documentation from the supplier such as drawings, code or other design and test descriptions.

For additional information and suggested test and analysis requirements for OTS products see JSC EA-WI-016 or MSFC MWI 8060.1 both titled "Off the Shelf Hardware Utilization in Flight Hardware Development" and G-118-2006e "AIAA Guide for Managing the Use of Commercial Off The Shelf (COTS) Software Components for Mission-Critical Systems."

## 5.2 Product Integration

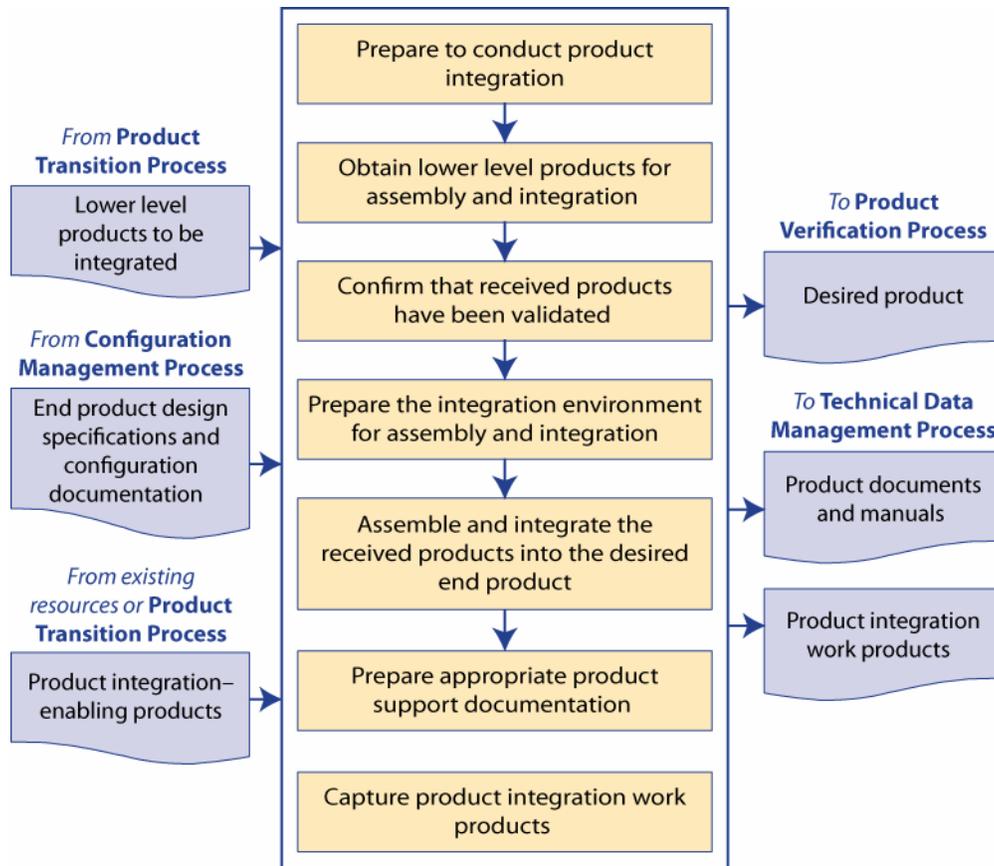
Product Integration is one of the SE engine product realization processes that make up the system structure. In this process, lower level products are assembled into higher level products and checked to make sure that the integrated product functions properly. It is the first element of the processes that lead from realized products from a level below to realized end products at a level above, between Product Implementation and the verification and validation processes.

The purpose of Product Integration is to systematically assemble the higher level product from the lower level products or subsystems (e.g., product elements, units, components, subsystems, or operator tasks); ensure that the product, as integrated, functions properly; and deliver the product. Product Integration is required at each level of the system hierarchy. The activities associated with product integrations occur throughout the entire product life cycle. This includes all of the incremental steps, including level appropriate testing, necessary to complete assembly of a product and to enable the top product level tests to be conducted. Product Integration may include and often begins with analysis and simulations (e.g., various types of prototypes) and progresses through increasingly more realistic incremental functionality until the final product is achieved. In each successive build, prototypes are constructed, evaluated, improved, and reconstructed based upon knowledge gained in the evaluation process. The degree of virtual vs. physical prototyping required depends on the functionality of the design tools, the complexity of the product, and its associated risk. There is a high probability that the product, integrated in this manner, will pass product verification and validation. For some products, the last integration phase will occur when the product is deployed at its intended operational site. If there are any problems of incompatibility which are discovered during the product verification and validation testing phase, they are resolved one at a time.

Product Integration applies not only to hardware and software systems but also to service-oriented solutions, requirements, specifications, plans, and concepts. The ultimate purpose of product integration is to ensure that the system elements will function as a whole.

### 5.2.1 Process Description

Figure 5-3 provides a typical flow diagram for the Product Integration Process and identifies typical inputs, outputs, and activities to consider in addressing product integration. The activities of the Product Integration process are truncated to indicate the action and object of the action.



**Figure 5-3 Product Integration Process**

### 5.2.1.1 Inputs

Product integration encompasses more than a one-time assembly of the lower level products and operator tasks at the end of the design and fabrication phase of the life cycle. An integration plan must be developed and documented. An example outline for an integration plan is provided in the appendix 5-2 of this handbook. Product integration is conducted incrementally, using a recursive process of assembling lower-level products and operator tasks; evaluating them through test, inspection, and/or analysis; and then assembling more lower-level products and operator tasks. Planning for product integration should be initiated during the concept formulation phase of the life cycle. The basic tasks that need to be established involve the management of internal and external interfaces of the products and lower-level products and operator tasks to support product integration and are as follows:

- Interface definitions
- Identification of the characteristics of the interfaces (physical, electrical, mechanical, etc.)
- Methods to ensure interface compatibility at all defined interfaces
- Assurance of interface compatibility at all defined interfaces

- Strict control of all of the interface processes during design, construction, operation, etc.)
- Identification of lower-level products to be assembled and integrated (from Product Transition Process).
- Identification of end product design definition specified requirements (specifications) and configuration documentation for the applicable Work Breakdown Structure (WBS) model, including interface specifications, in the form appropriate to satisfying the product-line life-cycle phase exit criteria (from Configuration Management Process).
- Identification of product integration enabling products (from existing resources or Product Transition Process for enabling product realization).

#### *5.2.1.2 Process Activities*

This section addresses the approach to the top level implementation of Product Integration, including the activities required to support Product Integration process. The Project would follow this approach throughout its life cycle.

The following are typical activities which support the Product Integration process:

- a. Prepare to conduct Product Integration to include (1) preparing a product integration strategy, detailed planning for the integration, and integration sequences and procedures; and (2) determining whether the product configuration documentation is adequately complete to conduct the type of product integration applicable for the product-line life-cycle phase, location of the product in the system structure, and management phase exit criteria.
- b. Obtain lower level products required to assemble and integrate into the desired product.
- c. Confirm that the received products that are to be assembled and integrated have been validated to demonstrate that the individual products satisfy the agreed to set of stakeholder expectations, including interfaces requirements.
- d. Prepare the integration environment in which assembly and integration will take place to include evaluating the readiness of the product integration enabling products and the assigned workforce.
- e. Assemble and integrate the received products into the desired end product in accordance with the specified requirements, configuration documentation, interface requirements, applicable standards, and integration sequencing and procedures.
- f. Conduct functional testing to ensure that assembly is ready to enter verification testing and ready to be integrated into the next level
- g. Prepare appropriate product support documentation such as special procedures for performing product verification and product validation.

- h. Capture work products and related information generated while performing the product integration process activities.

### *5.2.1.3 Outputs*

The following are typical outputs from this process and destinations for the products from this process:

- Integrated product(s) in the form appropriate to the product-line life-cycle phase and to satisfy phase exit criteria (to Product Verification Process).
- Documentation and manuals in a form appropriate for satisfying the life-cycle phase exit criteria, including “as-integrated” product descriptions and “operate-to” and maintenance manuals (to Technical Data Management Process).
- Work products, including reports, records, and non-deliverable outcomes of Product Integration process activities (to support the Technical Data Management Process).

## **5.2.2 Product Integration Guidance**

### *5.2.2.1 Relationship to Product Implementation*

As previously described, Product Implementation is where the plans, designs, analysis, requirements development, and drawings are realized into actual products. Product integration concentrates on the control of the interfaces and the verification and validation to achieve the correct product to meet the requirements. Product Integration can be thought of as release or phased deliveries. The Product Integration is the process that pulls together new and existing products and ensures that they all combine properly into a complete product without interference or complications. If there are issues, the Product Integration process documents the exceptions that can then be evaluated to determine if the product is ready for implementation/operations.

Integration occurs at every stage of a project’s life cycle. In the Formulation phase the decomposed requirements need to be integrated into a complete system to verify that nothing is missing or duplicated. In the Implementation phase the design and hardware need to be integrated into an overall system to verify it meets the requirements and that there are no duplications or omissions.

The emphasis on the recursive, iterative, and integrated nature of systems engineering highlights how the product integration activities are not only integrated across all of the phases of the entire life cycle in the initial planning stages of the project, but also used recursively across all of the life-cycle phases as the project product proceeds through the flow down and flow up conveyed by the SE engine. This ensures that when changes occur to requirements, design concepts, etc.—usually in response to updates from stakeholders and results from analysis, modeling, testing—that adequate course corrections are made to the project. This is accomplished through re-evaluation by driving through the SE engine, enabling all aspects of the product integration activities to be appropriately updated. The result is a product that meets all of the new modifications approved by the project and eliminates the opportunities for costly and time-consuming modifications in the later stages of the project.

#### *5.2.2.2 Product/Interface Integration Support*

There are several processes that support the integration of products and interfaces. Each process allows either the integration of products and interfaces or the validation that the integrated products meet the needs of the project.

The following is a list of typical example processes, but is not all inclusive that support the integration of products and interfaces and should be addressed by the project in their overall approach to Product Integration: requirements documents, requirements reviews, design reviews, design drawings and specifications, integration and test, plans, hardware configuration control documentation, quality assurance records, interface control requirement/documents, operational concept documents, verification requirement documents, verification reports/analysis, NASA and industry standards, best practices and lessons learned.

#### *5.2.2.3 Product Integration of Design Solution*

This section addresses the more specific implementation of Product Integration related to the selected design solution.

Generally, system/product designs are a culmination of subsystems and components. This is relatively obvious for complex hardware and/or software systems. The same holds true for many service oriented solutions. For example, a solution to provide a single person access to the internet involves hardware, software, and a communications interface. The purpose of product integration is to ensure that combination of these elements achieves the required result (i.e., works as expected). Consequently, internal and external interfaces must be considered in the design and evaluated prior to production.

Proof of integration is typically accomplished through testing in an operational environment. There are a variety of different testing requirements to validate product integration at all levels. Qualification testing and acceptance testing are examples of two of these test types that are performed as the product is integrated. Another type of testing which is important to the design and ultimate product integration is a planned test process in which development items are tested under actual or simulated mission profile environments to disclose design deficiencies and to provide engineering information on failure modes and mechanisms. If accomplished with development items, this provides early insight into any issues which may otherwise only be observed at the late stages of product integration where it becomes costly to incorporate corrective actions. For large, complex system/products, integration/verification efforts are accomplished using a prototype.

#### *5.2.2.4 Interface Management*

The objective of the interface management is to achieve functional and physical compatibility between all interrelated system elements. Interface management is defined in more detail in the crosscutting section 6.3. An interface is any boundary between one area and another. It may be external, internal, functional, or physical. Interfaces occur within the system (internal) as well as between the instant system and another system (external) and may be functional or physical (e.g., mechanical, electrical) in nature. Interface requirements are documented in an Interface Requirements Document (IRD). In its final form, the Interface Control Document (ICD) is a

“design” document that describes the detailed “as built” implementation of the requirements contained in the IRD. An Interface Control Plan describes the management process for IRDs and ICDs. This plan provides the means to identify and resolve interface incompatibilities and to determine the impact of interface design changes.

#### *5.2.2.5 Utilization and Value*

During the program’s life, compatibility and accessibility shall be maintained for the many diverse elements. Compatibility analysis of the interface definition demonstrates completeness of the interface and traceability records. As changes are made, an authoritative means of controlling the design of interfaces shall be managed with appropriate documentation, thereby avoiding the situation in which hardware/software, when integrated into the system, fails to function as part of the system, as intended. Ensuring that all system pieces work together is a complex task that involves teams, stakeholders, contractors, and program management from the end of the initial concept definition stage through the operations and support stage. Physical integration is accomplished during Phase D. At the finer levels of resolution, pieces must be tested, assembled and/or integrated, and tested again. The systems engineer’s role includes the performance of the delegated management duties, such as configuration control and overseeing the integration, verification, and validation process.

#### *5.2.2.6 Interface Management Tasks*

The interface management tasks begin early in the development effort, when interface requirements can be influenced by all engineering disciplines and applicable interface standards can be invoked. They continue through design and checkout. During design, emphasis is on ensuring that interface specifications are documented and communicated. During system element checkout, both prior to assembly and in the assembled configuration, emphasis is on verifying the implemented interfaces. Throughout the product integration process activities, interface baselines are controlled to ensure that changes in the design of system elements have minimal impact on other elements to which they interface. During testing, or other validation and verification activities, multiple system elements are checked out as integrated subsystems or systems.

### **Define Interfaces**

The bulk of integration problems arise from unknown or uncontrolled aspects of interfaces. Therefore, system and subsystem interfaces are specified as early as possible in the development effort. Interface specifications address logical, physical, electrical, mechanical, human, and environmental parameters as appropriate. Intra-system interfaces are the first design consideration for developers of the system’s subsystems. Interfaces are used from previous development efforts or are developed in accordance with interface standards for the given discipline or technology. Novel interfaces are constructed only for compelling reasons. Interface specifications are verified against interface requirements. Typical products include: interface descriptions, interface control documents, interface requirements specifications.

## **Verify Interfaces**

In verifying the interfaces the systems engineer is to ensure that the interfaces of each element of the system or subsystem are controlled and known to the developers. Additionally, when changes to the interfaces are needed, the changes must at least be evaluated for possible impact to other interfacing elements and then communicated to the affected developers. Although all affected developers are part of the group that makes changes, such changes need to be captured in a readily accessible place so that the current state of the interfaces can be known to all. Typical products include: interface control documents, exception reports.

The use of emulators for verifying hardware and software interfaces is acceptable where the limitations of the emulator are well-characterized and meet the operating environment characteristics and behavior requirements for interface verification. The integration plan should specifically document the scope of use for emulators.

## **Integration Strategy**

An integration strategy is developed, as well as supporting documentation, to identify optimal sequence of receipt, assembly, and activation of the various components that make up the system. This strategy should use business as well as technical factors to ensure an assembly, activation, and loading sequence that minimize cost and assembly difficulties. The larger or more complex the system or the more delicate the elements, the more critical the proper sequence becomes, as small changes can cause large impacts on project results.

The optimal sequence of assembly is built from the bottom up as components become sub-elements, elements, and subsystems, each of which must be checked prior to fitting into the next higher assembly. The sequence will encompass any effort needed to establish and equip the assembly facilities (e.g., raised floor, hoists, jigs, test equipment, Input/Output (I/O), and power connections). Once established, the sequence must be periodically reviewed to ensure that variations in production and delivery schedules have not had an adverse impact on the sequence or compromised the factors on which earlier decisions were made. To ensure the integration strategy is complete the following items should be addressed as a minimum.

Integration strategy minimum contents:

- interface management documentation
- personnel requirements
- assembly area drawings
- special handling
- system documentation for systems engineering users
- shipping schedule
- assembly sequence and rationale
- test equipment and drivers
- emulator requirements and limitations for both hardware and software

Typical products include: integration strategy document, assembly/check area drawings, system/component documentation, sequence and rationale for selected assembly.

## **System and Subsystem Receipt**

Acknowledging receipt and inspecting the condition of each system or subsystem element is required prior to assembling the system in accordance with the intended design. The elements are checked for quantity, obvious damage, and consistency between the element description and a list of element requirements. Typical products include; acceptance documents, delivery receipts, checked packing list.

## **System and Subsystem Element Validation**

System and subsystem element validation is to verify the implemented design features of developed or purchased system elements against their requirements. This is intended to ensure that each element of the system or subsystem functions in its intended environment, including those elements that are off-the-shelf for other environments. Such verifications may be by test (e.g., regression testing as a tool or subsystem/elements are combined), inspection, analysis (deficiency or compliance reports), etc., and may be executed either by the organization that will assemble the system or subsystem or by the producing organization. A method of discerning the elements that “passed” verification from those elements that “failed” needs to be in place. Typical products include verified system features, exception reports.

## **Verify Element Interfaces**

Verification of the system elements interfaces are to ensure that the element complies with the interface specification prior to assembly in the system. The intent is to ensure that the interface of each element of the system or subsystem is verified against its corresponding interface specification. Such verification may be by test, inspection, analysis, etc., and may be executed by the organization that will assemble the system or subsystem or by another organization. Typical products include verified system element interfaces, test reports, exception reports

## **Integration and Verification**

Assembly of the elements of the system should be performed in accordance with the established integration strategy. This is intended to ensure that the assembly of the system elements into larger or more complex assemblies is conducted in accordance with the planned strategy. To ensure the integration is completed a verification of the integrated system interfaces is to be performed. Typical products include: integration reports, exception reports, and an integrated system.

### 5.3 Product Verification

The Product Verification Process is the first of the verification and validation (V&V) processes conducted on a realized end product. As used in the context of systems engineering common technical processes, a realized product is one provided by either the Product Implementation Process or the Product Integration Process in a form suitable for meeting applicable life-cycle phase exit criteria. Realization is the act of verifying, validating, and transitioning the realized product for use at the next level up of the system structure or to the customer. Simply put, the Product Verification Process answers the critical question “*Was the end product realized right?*” while the Product Validation Process addresses the equally critical question of “*Was the right end product realized?*”

Verification proves that a realized product for any system model within the system structure conforms to the “design-to” requirements (for software elements) or “realize-to” specifications and design descriptive documents (for hardware elements, manual procedures, or composite products of hardware, software and manual procedures).

#### **Distinctions Between Verification and Validation**

From a process perspective, Product Verification and Product Validation may be similar in nature, but the objectives are fundamentally different.

From a customer point of view, the interest is in whether the end product provided will do what they intend within the environment of use. It is essential to confirm that the realized product is in conformance with its specifications and design description documentation in that such specifications and documents will establish the configuration baseline of that product which may have to be modified at a later time. Without a verified baseline and appropriate configuration controls such later modifications could be costly or cause major performance problems.

When cost-effective and warranted by analysis, various “combined tests” are used. The expense of validation testing alone can be mitigated by assuring that each end product in the system structure was correctly realized in accordance with its specified requirements before conducting validation.

### Differences between Verification and Validation Testing:

**Verification Testing:** Verification testing relates back to the approved requirements set (such as a Systems Requirements Document) and can be performed at different stages in the product life cycle. Verification Testing includes: (1) any testing used to assist in the development and maturation of products, product elements, or manufacturing or support processes; and/or (2) any engineering-type test used to verify status of technical progress, to verify that design risks are minimized, to substantiate achievement of contract technical performance, and to certify readiness for initial validation testing. Verification tests use instrumentation and measurements, and are generally accomplished by engineers, technicians, or operator-maintainer test personnel in a controlled environment to facilitate failure analysis.

**Validation Testing:** Validation relates back to the Concept of Operations document. Validation testing is conducted under realistic conditions (or simulated conditions) on any end product for the purpose of determining the effectiveness and suitability of the product for use in mission operations by typical users; and the evaluation of the results of such tests. Testing is the detailed quantifying method of both verification and validation. However, testing is required to validate final end products to be produced and deployed.

The outcome of the Product Verification Process is confirmation that the “as realized product,” whether achieved by Implementation or Integration, conforms to its specified requirements; i.e., verification of the end product. This section discusses the process activities, inputs, outcomes, and potential deficiencies.

#### 5.3.1 Process Description

Figure 5-4 provides a typical flow diagram for the Product Verification Process and identifies typical inputs, outputs, and activities to consider in addressing product verification.

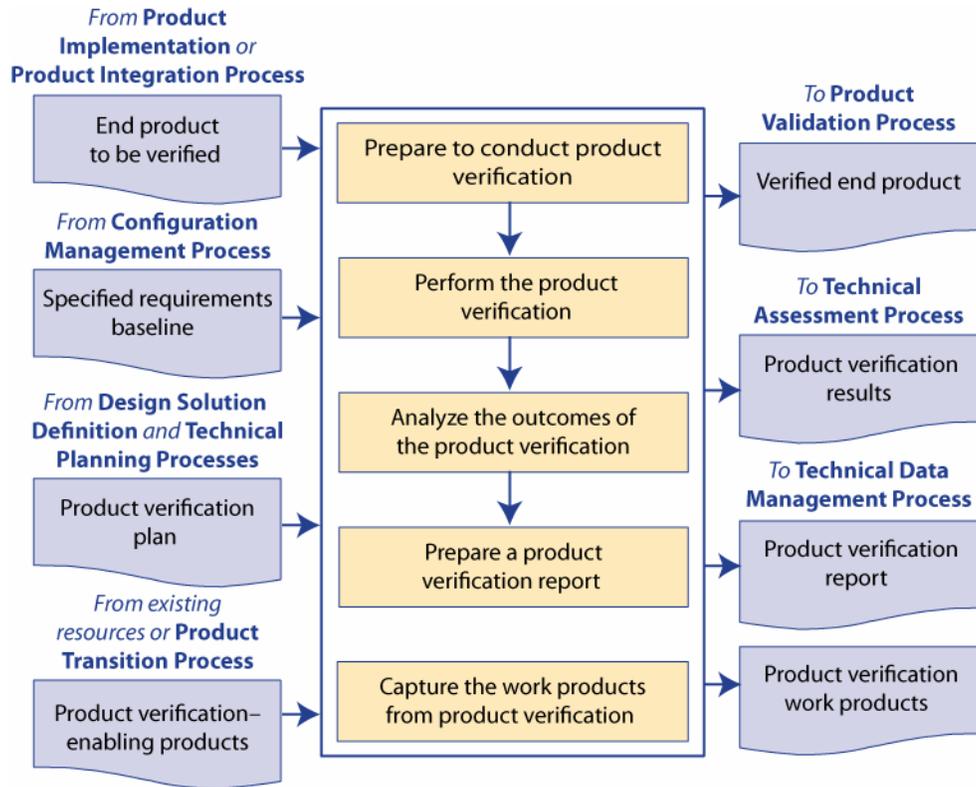


Figure 5-4 Product Verification Process

### 5.3.1.1 *Inputs*

Key inputs to the process are the product to be verified, Verification Plan, specified requirements baseline, and any enabling products needed to perform the Product Verification Process (including the Concept of Operations, mission needs and goals, requirements and specifications, interface control drawings, testing standards and policies, and agency standards and policies).

### 5.3.1.2 *Process Activities*

There are five major steps in the Verification Process: (1) Verification Planning (Prepare to Implement the Verification Plan); (2) Verification preparation (Prepare for Conducting Verification); (3) Conduct Verification (Perform Verification); (4) Analyze Verification Results; and (5) Capture the Verification Work Products.

The objectives of the Product Verification Process are:

- To generate evidence necessary to confirm that end products, from the lowest level of the system structure to the highest, conforms to the specified requirements (specifications and descriptive documents) to which it was realized whether by the Product Implementation Process or by the Product Integration Process; and
- To verify that the materials employed in end product solutions can be used in a safe and environmentally compliant manner.

Product Verification is usually performed by the developer that produced (or “realized”) the end product, with participation of the end user and customer. Product Verification confirms that the “as-realized product,” whether it was achieved by Product Implementation or Product Integration, conforms to its specified requirements (specifications and descriptive documentation) used for making or assembling and integrating the end product. Developers of the system, as well as the users, are typically involved in Verification Testing. The customer and Quality Assurance personnel are also critical in the verification planning and execution activities.

### **Product Verification Planning**

Planning to conduct the product verification is a key first step. From relevant specifications and product form, the type of verification (e.g., analysis, demonstration, inspection, or test) should be established based on the life-cycle phase, cost, schedule, resources, and position of the end product within the system structure. The Verification Plan should be reviewed (an output of the Technical Planning Process, based on design solution outputs) for any specific procedures, constraints, success criteria, or other verification requirements. (See appendix 5-4 for a sample verification plan outline.)

### **Verification Plan and Methods**

The task of preparing the Verification Plan includes establishing the type of verification to be performed, dependent on the life-cycle phase; position of the product in the system structure; the form of the product used; and related costs of verification of individual specified requirements. The Types of Verification include (1) Analyses; (2) Inspection; (3) Demonstration; and (4) Test.

The Verification Plan, typically written at a detailed technical level, plays a pivotal role in bottom-up product realization. *Close alignment of the Verification Plan with the project's Systems Engineering Management Plan (SEMP) is absolutely essential.*

### Types of Testing

There are many different types of testing that can be used in verification of an end product. Several examples are provided for consideration:

Aerodynamic	Life-time/cycling
Acceptance	Manufacturing/Random Defects
Acoustic	Nominal
Burn-in	Off-nominal
Characterization	Operational
Component	Parametric
Electromagnetic Compatibility	Performance
Electromagnetic Interference	Pressure Cycling
Environmental	Pressure Limits
Drop	Qualification
Flow	Structural
Functional	System
Go/No-Go	Thermal Cycling
High/Low Voltage Limits	Thermal Limits
Integration	Thermal Vacuum
Leak Rates	Vibration

Verification can be performed recursively throughout the project life cycle and on a wide variety of product forms. For example:

- Simulated (algorithmic models, virtual reality simulator);
- Mock-up (plywood, brassboard, breadboard);
- Concept description (paper report); and
- Prototype (product with partial functionality)
- Engineering Unit (fully functional but may not be same form/fit)
- Design Verification Test Units (form, fit and function same, but may not have flight parts);
- Qualification units (Identical to flight units but may be subjected to extreme environments)
- Flight units (end product that is flown)

Any of these types of product forms may be in any of these states

- Produced (built, fabricated, manufactured, or coded);
- Reused (modified internal non-developmental products or off-the-shelf product); and

- Assembled and integrated (a composite of lower level products).

Note: The final, official verification of the end product should be for a controlled unit. Typically, attempting to “buy-off” a ‘shall’ on a prototype is not acceptable; it is usually completed on a qualification, flight, or other more-final, controlled unit.

### Types of Verification

**Analysis:** The use of mathematical modeling and analytical techniques to predict the compliance of a design to its requirements based on calculated data or data derived from lower system structure end product verifications. Analysis is generally used when a prototype, engineering model or fabricated, assembled and integrated product is not available. Analysis includes the use of M&S as analytical tools.

**Demonstration:** Showing the use of an end product achieves the individual specified requirement. It is generally a basic confirmation of performance capability, differentiated from testing by the lack of detailed data gathering. Demonstrations can involve the use of physical models or mock-ups; for example, a requirement that all controls shall be reachable by the pilot could be verified by having a pilot perform flight related tasks in a cockpit mock-up or simulator. A demonstration could also be the actual operation of the end product by highly-qualified personnel, such as test pilots, who perform a one-time event that demonstrates a capability to operate at extreme limits of system performance, an operation not normally expected from a representative operational pilot.

**Inspection:** The visual examination of a realized end product. Inspection is generally used to verify physical design features or specific manufacturer identification. For example, if there is a requirement that the safety arming pin has a red flag with the words “Remove Before Flight” stenciled on the flag in black letters, a visual inspection of the arming pin flag can be used to determine if this requirement was met.

**Test:** The use of an end product to obtain detailed data needed to verify performance, or provide sufficient information to verify performance through further analysis. Testing can be conducted on final end products, breadboards, brassboards or prototypes. Testing produces data at discrete points for each specified requirement under controlled conditions and is the most resource-intensive verification technique. As the saying goes, “*Test as you fly, and fly as you test.*” See the “End-to-End Testing” provided in section 5.3.2.5

The conditions and environment under which the product is to be verified should be established and the verification planned based on the associated entrance/success criteria identified. The Decision Analysis Process should be used to help finalize the planning details.

**NOTE:** Verification planning is begun early in the project life cycle during the requirements development phase (See section 4.2.5). Which verification approach to be used should be included as part of the requirements development in order to plan for the future activities, establish special requirements derived from verification enabling products identified, and to ensure that the technical statement is, in fact, a requirement. Updates to verification planning continue throughout the logical decomposition and design development phases, especially as design reviews and simulations shed light on items under consideration. (See Section 6.1, Technical Planning.)

Procedures should be prepared to conduct verification based on the type (e.g., Analysis, Inspection, Demonstration, or Test) planned. These procedures are typically developed during the design phase of the project life cycle and matured as the design is matured. Operational/use scenarios are thought through so as to explore all possible verification activities to be performed.

Outcomes of Verification Planning include the following:

- The verification type that is appropriate for showing/proving the realized product conforms to its specified requirements.
- The product verification procedures are clearly defined based on: (1) the procedures for each type of verification selected; (2) the purpose and objective of each procedure; (3) any pre-verification and post-verification actions; and (4) the criteria for determining the success or failure of the procedure.
- A defined verification environment (e.g., facilities, equipment, tools, simulations, measuring devices, personnel and climatic conditions) in which the verification procedures will be implemented.
- As appropriate updated project risk items based on approved verification strategies that cannot duplicate fully integrated tests systems, configurations, and/or target operating environments. Rationale, trade space, optimization results, and implications of the approaches are documented in the new/revised risk statements as well for future references to accommodate future design, test, and operational changes to the project baseline.

### **Product Verification Preparation**

In preparation of verification, the specified requirements (outputs of the Design Solution process) are collected and confirmed. The product to be verified is obtained (output from implementation or integration), as are any enabling products and support resources that are necessary for verification (requirements identified and acquisition initiated by Design Solution Definition activities). The final element of Verification Preparation includes the preparation of the verification environment (e.g., facilities, equipment, tools, simulations, measuring devices, personnel and climatic conditions). Identification of the environmental requirements is necessary and the implications of those requirements must be carefully considered.

**NOTE:** Depending on the nature of the verification effort and the life-cycle phase the program is in, some type of review to assess readiness for verification (as well as validation later) is typically held. In earlier phases of the life cycle, these reviews may be held informally; in later phases of the life cycle, this review becomes a formal event called a Test Readiness Review (TRR). TRRs and other technical reviews are an activity of the Technical Assessment Process.

On most projects, a number of TRRs with tailored entrance/success criteria are held to assess the readiness and availability of such items as test ranges; test facilities; trained testers; instrumentation; availability of integration labs, support equipment and other enabling products; etc.

Peer Reviews are additional reviews that may be conducted formally or informally to ensure readiness for verification (as well as the results of the verification process).

Outcomes of verification preparation include the following:

- The preparations for performing the as-planned verification are completed.
- An appropriate set of specified requirements and supporting configuration documentation are available and on hand.

- Articles/Models to be used for verification are on hand, assembled and integrated with the verification environment according to verification plans and schedules.
- The resources needed to conduct the verification are available according to the verification plans and schedules.
- The verification environment is checked-out for adequacy, completeness, readiness and integration.

### **Conduct Planned Product Verification**

The actual act of Verifying the end product is conducted as spelled out in the plans and procedures and conformance established to each specified verification requirement. The responsible engineer should ensure that the procedures were followed and performed as planned, the verification enabling products were calibrated correctly, and the data was collected and recorded for required verification measures.

When poor verification design, conduct, etc. cause anomalies, the verification should be re-planned as necessary, the environment preparation anomalies corrected, and the verification conducted again.

The Decision Analysis Process should be used to help make decisions with respect to making needed changes in the verification plans, environment, and/or conduct.

Outcomes of Conducting Verification include the following:

- A verified product is established with supporting confirmation that the appropriate results were collected and evaluated to show completion of verification objectives.
- A determination as to whether the realized end product (in the appropriate form for the life-cycle phase) complies with its specified requirements.
- A determination that the verification product was appropriately integrated with the verification environment and each specified requirement was properly verified.
- A determination that product functions were verified both together and with interfacing products throughout their performance envelope.

### **Analyze Product Verification Results**

Once the verification activities have been completed, the results are collected and analyzed. The data is analyzed for quality, integrity, correctness, consistency, and validity, and any verification anomalies, variations, and out-of-compliance conditions are identified and reviewed.

System design and product realization process activities may be required to resolve anomalies *not* resulting from poor verification conduct, poor verification design, or poor verification conditions. Variations, anomalies, and out-of-compliance conditions must be recorded and reported for following action and closure. Verification results should be recorded in the

requirements compliance matrix developed during the Requirements Development process or other mechanism to trace compliance for each verification requirement.

If there are anomalies *not* resulting from the verification conduct, design, or conditions, the verification should be re-planned as necessary and the verification conducted again.

Outcomes of Analyzing the Verification Results include the following:

- End product variations, anomalies, and out-of-compliance conditions have been identified.
- Assurances that appropriate re-planning, redefinition of requirements, design and re-verification have been accomplished for resolution for anomalies, variations, or out-of-compliance conditions (for problems not caused by poor verification conduct).
- Discrepancy and Corrective Action Reports have been generated as needed.

**NOTE:** Non-conformances and discrepancy reports may be directly linked with the Technical Risk Management process. Depending on the nature of the non-conformance, approval through such bodies as a Material Review Board or Configuration Control Board (which typically includes risk management participation) may be required.

- The Verification Report is completed.

## Reengineering

Based on analysis of verification results, it could be necessary to re-realize the end product used for verification, or to reengineer the end products assembled and integrated into the product being verified, based on where and what the type of defect that was found.

## Verification Deficiencies

Verification test outcomes can be unsatisfactory for several reasons. One reason is poor conduct of the verification (e.g., procedures not followed, equipment not calibrated, improper verification environmental conditions, failure to control other variables not involved in verifying a specified requirement). A second reason could be that the realized end product used was not realized correctly. Re-engineering could require the re-application of the system design processes (*Stakeholder Expectations Definition, Technical Requirements Definition, Logical Decomposition and Design Solution Definition*) to obtain a corrected set of specified requirements. This re-application could create the need for the following:

- Reengineering products lower in the system structure that make up the product that was found to be defective (i.e., which failed to satisfy verification requirements); and/or
- Re-performing the Product Verification Process.

## **Pass Verification But Fail Validation?**

Many systems successfully complete verification but then are unsuccessful in some critical phase of the validation process, delaying development and causing extensive re-work and possible compromises with the Stakeholder. Developing a solid ConOps in early phases of the project (and refining it through the requirements development and design phases) is critical to preventing unsuccessful validation. Communications with the stakeholders helps to identify operational scenarios and key ‘needs’ that must be understood when designing and implementing the end product. Should the product ‘fail’ validation, redesign may be a necessary reality. Review of the understood requirements set, the existing design, operational scenarios, and support material may be necessary, as well as negotiations and compromises with the customer and/or end-users to determine what, if anything can be done to correct or resolve the situation. This can add time and cost to the overall project or, in some cases, cause the project to fail or be cancelled.

## **Capture Product Verification Work Products**

Verification work products (inputs to the Technical Data Management Process) take many forms and involve many sources of information. The capture and recording of verification results and related data is a very important, but often under-emphasized, step in the Verification Process.

Verification results, anomalies, and any corrective action(s) taken should be captured, as should all relevant results from the application of the Product Verification Process (related decisions, rationale for the decisions made, assumptions, lessons learned).

Outcomes of Capturing Verification Work Products include the following:

- Verification of work products are recorded; i.e., type of verification, procedures, environments, outcomes, decisions, assumptions, corrective actions, lessons learned, etc.
- Variations, anomalies and out-of-compliance conditions have been identified and documented, including the actions taken to resolve
- Proof that the realized end product satisfies the specified requirements.
- Recorded verification result data including.
  - Version of the set of specified requirements used;
  - Version of the product verified;
  - Version or standard for tools, data and equipment used;
  - Results of each verification including pass or fail declarations; and
  - Expected vs. actual discrepancies.

### *5.3.1.3 Outputs*

Key outputs from the process are:

- Discrepancy reports and Identified Corrective Actions
- Verification Report(s) and updates to requirements compliance documentation (including verification plans, verification procedures, verification matrices, verification results and analysis, and test/demonstration/inspection/analysis records)

Exit criteria includes (1) objective evidence of compliance (or waiver, as appropriate) of each system of interest “*shall*” requirements are documented; and (2) the verification process is not considered or designated as complete until all discrepancy reports are closed (i.e., all errors tracked to closure).

### **5.3.2 Product Verification Guidance**

#### *5.3.2.1 Verification Program*

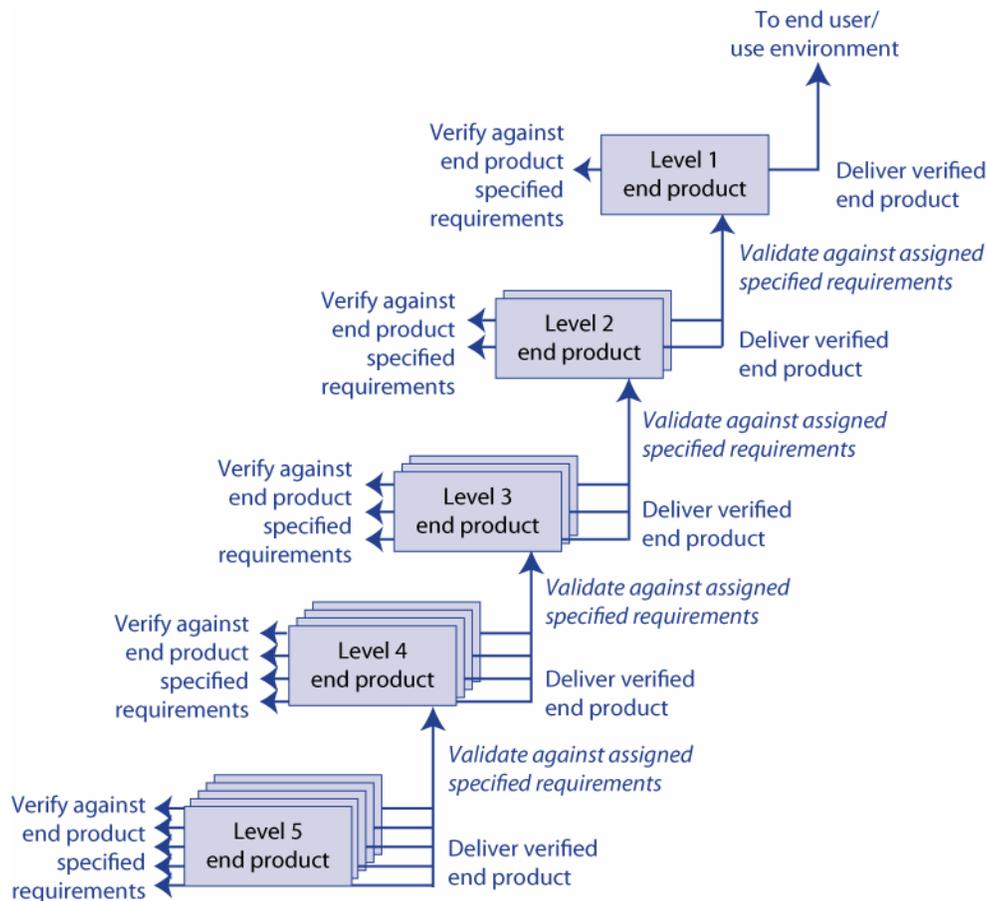
A verification program should be tailored to the project it supports. The project manager/systems engineer must work with the verification engineer to develop a verification program concept. Many factors need to be considered in developing this concept and the subsequent verification program. These factors include:

- Project type, especially for flight projects. Verification methods and timing depend on:
  - The type of flight article involved (e.g., an experiment, payload, or launch vehicle).
  - NASA payload classification (NPR 8705.4). Guidelines are intended to serve as a starting point for establishment of the formality of test programs which can be tailored to the needs of a specific project based on the “A-D” payload classification.
  - Project cost and schedule implications. Verification activities can be significant drivers of a project’s cost and schedule; these implications should be considered early in the development of the verification program. Trade studies should be performed to support decisions about verification methods and requirements and the selection of facility types and locations. For example, a trade study might be made to decide between performing a test at a centralized facility or at several decentralized locations.
  - Risk implications. Risk management must be considered in the development of the verification program. Qualitative risk assessments and quantitative risk analyses (e.g., a FMECA) often identify new concerns that can be mitigated by additional testing, thus increasing the extent of verification activities. Other risk assessments contribute to trade studies that determine the preferred methods of verification to be used and when those methods should be performed. For example, a trade might be made between performing a model test versus determining model characteristics by a less costly, but less revealing, analysis. The project manager/systems engineer must determine what risks are acceptable in terms of the project’s cost and schedule.
- Availability of verification facilities/sites and transportation assets to move an article from one location to another (when needed). This requires coordination with the ILS engineer;

- Acquisition strategy (i.e., in-house development or system contract). Often a NASA field center can shape a contractor's verification process through the project's Statement of Work (SoW); and
- Degree of design inheritance and hardware/software reuse.

### 5.3.2.2 Verification in the Life Cycle

The type of verification completed will be a function of the life-cycle phase and the position of the end product within the system structure. The end product must be verified and validated before it is transitioned to the next level up as part of the bottoms-up realization process. (See Figure 5-5).



**Figure 5-5 Bottoms-Up Realization Process**

While illustrated here as separate processes, there can be considerable overlap between some V&V events when implemented.

### Quality Assurance in Verification

Even with the best of available designs, hardware fabrication (and software coding) and testing, projects are subject to the vagaries of Nature and human beings. The systems engineer needs to

have some confidence that the system actually produced and delivered is in accordance with its functional, performance, and design requirements. Quality Assurance (QA) provides an independent assessment to the project manager/systems engineer of the items produced and processes used during the project life cycle. The quality assurance engineer typically acts as the systems engineer's eyes and ears in this context.

The quality assurance engineer typically monitors the resolution and close-out of nonconformances and Problem/Failure Reports; verifies that the physical configuration of the system conforms to the "build-to" (or "code-to") documentation approved at CDR; and collects and maintains QA data for subsequent failure analyses. The quality assurance engineer also participates in major reviews (primarily SRR, PDR, CDR, and FRR) on issues of design, materials, workmanship, fabrication and verification processes, and other characteristics that could degrade product system quality.

The project manager/systems engineer must work with the quality assurance engineer to develop a Quality Assurance Program (the extent, responsibility, and timing of QA activities) tailored to the project it supports. In part, the Quality Assurance Program ensures verification requirements are properly specified, especially with respect to test environments, test configurations, and pass/fail criteria, and monitors qualification and acceptance tests to ensure compliance with verification requirements and test procedures to ensure that test data are correct and complete.

### **Configuration Verification**

Configuration verification is the process of verifying that resulting products (e.g., hardware and software items) conform to the intentions of the designers and to the standards established by preceding approved baselines, and that baseline documentation is current and accurate. Configuration verification is accomplished by two types of control gate activity: audits and technical reviews.

### **Qualification Verification**

Qualification stage verification activities begin after completion of development of the flight hardware designs, and includes analyses and testing to ensure that the flight or flight-type hardware (and software) will meet functional and performance requirements in anticipated environmental conditions. During this stage many performance requirements are verified, while analyses and models are updated as test data is acquired. Qualification tests generally are designed to subject the hardware to worst-case loads and environmental stresses. Some of the verifications performed to ensure hardware compliance are vibration/acoustic, pressure limits, leak rates, thermal vacuum, thermal cycling, electromagnetic interference and electromagnetic compatibility (EMI/EMC), high and low voltage limits, and life time/cycling. Safety requirements, defined by hazard analysis reports, may also be satisfied by qualification testing.

Qualification usually occurs at the component or subsystem level, but could occur at the system level as well. When a project decides against building dedicated qualification hardware, using the flight hardware itself for qualification purposes, the process is termed *protoflight*.

## Acceptance Verification

Acceptance stage verification activities provide the assurance that the flight hardware and software are in compliance with all functional, performance, and design requirements, and are ready for shipment to the launch site. The acceptance stage begins with the acceptance of each individual component or piece part for assembly into the flight article, continuing through the System Acceptance Review (See Technical Review Process).

Some verifications cannot be performed after a flight article, especially a large one, has been assembled and integrated (e.g., due to inaccessibility). When this occurs, these verifications are to be performed *during* fabrication and integration, and are known as *in-process tests*. In this case, acceptance testing begins with in-process testing and continues through functional testing, environmental testing, and end-to-end compatibility testing. Functional testing normally begins at the component level and continues at the systems level, ending with all systems operating simultaneously.

When flight hardware is unavailable, or its use is inappropriate for a specific test, simulators may be used to verify interfaces. Anomalies occurring during a test are documented on the appropriate reporting system, and a proposed resolution should be defined before testing continues. Major anomalies, or those that are not easily dispositioned, may require resolution by a collaborative effort of the systems engineer, and the design, test, and other organizations. Where appropriate, analyses and models are validated and updated as test data is acquired.

## Deployment Verification

The pre-launch verification stage begins with the arrival of the flight article at the launch site and concludes at liftoff. During this stage, the flight article is processed and integrated with the launch vehicle. The launch vehicle could be the Shuttle, or some other launch vehicle; or the flight article could be part of the launch vehicle. Verifications performed during this stage ensure that no visible damage to the system has occurred during shipment and that the system continues to function properly.

If system elements are shipped separately and integrated at the launch site, testing of the system and system interfaces is generally required. If the system is integrated into a carrier, the interface to the carrier must also be verified. Other verifications include those that occur following integration into the launch vehicle and those that occur at the launch pad; these are intended to ensure that the system is functioning and in its proper launch configuration. Contingency verifications and procedures are developed for any contingencies that can be foreseen to occur during pre-launch and countdown. These contingency verifications and procedures are critical in that some contingencies may require a return of the launch vehicle or flight article from the launch pad to a processing facility.

## Operational and Disposal Verification

Operational verification provides the assurance that the system functions properly in a (near-) zero gravity and vacuum environment. These verifications are performed through system activation and operation, rather than through a verification activity. Systems that are assembled on-orbit must have each interface verified, and must function properly during end-to-end testing.

Mechanical interfaces that provide fluid and gas flow must be verified to ensure no leakage occurs, and that pressures and flow rates are within specification. Environmental systems must be verified.

Disposal verification provides the assurance that the safe deactivation and disposal of all system products and processes has occurred. The disposal stage begins in Phase E at the appropriate time (i.e., either as scheduled, or earlier in the event of premature failure or accident), and concludes when all mission data have been acquired and verifications necessary to establish compliance with disposal requirements are finished.

Both operational and disposal verification activities may also include validation assessments—that is, assessments of the degree to which the system accomplished the desired mission goals/objectives.

### *5.3.2.3 Verification Procedures*

Verification procedures provide step-by-step instructions for performing a given verification activity to be performed to satisfy a requirement. This procedure could be a test, demonstration, or any other verification-related activity. The procedure to be used is written and submitted prior to the Test Readiness Review (TRR) or the start of the verification activity in which the procedure is used. (See section on TRRs.)

Procedures are also used to verify the acceptance of facilities, electrical and mechanical ground support equipment, and special test equipment. The information generally contained in a procedure is as follows, but it may vary according to the activity and test article:

- Nomenclature and identification of the test article or material;
- Identification of test configuration and any differences from flight configuration;
- Identification of objectives and criteria established for the test by the applicable verification specification;
- Characteristics and design criteria to be inspected or tested, including values, with tolerances, for acceptance or rejection;
- Description, in sequence, of steps and operations to be taken;
- Identification of computer software required;
- Identification of measuring, test, and recording equipment to be used, specifying range, accuracy, and type;
- Certification that required computer test programs/support equipment and software have been verified prior to use with flight hardware;
- Any special instructions for operating data recording equipment or other automated test equipment as applicable;

- Layouts, schematics, or diagrams showing identification, location, and interconnection of test equipment, test articles, and measuring points;
- Identification of hazardous situations or operations;
- Precautions and safety instructions to ensure safety of personnel and prevent degradation of test articles and measuring equipment;
- Environmental and/or other conditions to be maintained with tolerances;
- Constraints on inspection or testing;
- Special instructions for non-conformances and anomalous occurrences or results; and
- Specifications for facility, equipment maintenance, housekeeping, certification inspection, and safety and handling requirements before, during, and after the total verification activity.

The written procedure may provide blank spaces in the format for recording of results and narrative comments in order that the completed procedure can serve as part of the verification report. The as-run and certified copy of the procedure is maintained as part of the project's archives.

#### 5.3.2.4 *Verification Reports*

A verification report should be provided for each analysis and, at a minimum, for each major test activity, such as functional testing, environmental testing, and end-to-end compatibility testing occurring over long periods of time or separated by other activities. Verification reports may be needed for each individual test activity, such as functional testing, acoustic testing, vibration testing, and thermal vacuum/thermal balance testing. Verification reports should be completed within a few weeks following a test, and should provide evidence of compliance with the verification requirements for which it was conducted.

The verification report should include as appropriate:

- Verification objectives and degree to which they were met;
- Description of verification activity;
- Test configuration and differences from flight configuration;
- Specific result of each test and each procedure including annotated tests;
- Specific result of each analysis;
- Test performance data tables, graphs, illustrations, and pictures;

- Descriptions of deviations from nominal results, problems/failures, approved anomaly corrective actions, and re-test activity;
- Summary of non-conformance/discrepancy reports including dispositions;
- Conclusion and recommendations relative to success of verification activity;
- Status of support equipment as affected by test;
- Copy of as-run procedure; and
- Authentication of test results and authorization of acceptability.

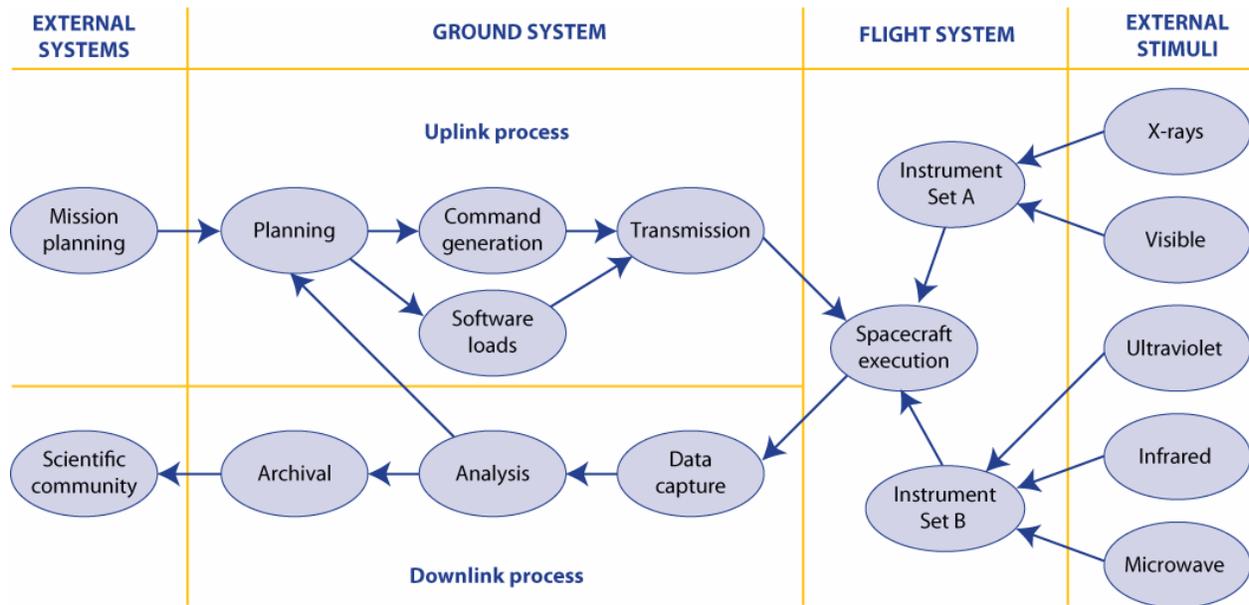
#### 5.3.2.5 *End-to-End System Testing*

The objective of end-to-end testing is to demonstrate interface compatibility and desired total functionality among different elements of a system, between systems, and within a system as a whole. End-to-end tests performed on the integrated ground and flight system include all elements of the payload, its control, stimulation, communications, and data processing to demonstrate that the entire system is operating in a manner to fulfill all mission requirements and objectives.

End-to-end testing includes executing complete threads or operational scenarios across multiple configuration items assuring that all mission and performance requirements are verified. Operational scenarios are used extensively to ensure that the system (or collections of systems) will successfully execute mission requirements. Operational scenarios are a step-by-step description of how the system should operate and interact with its users and its external interfaces (e.g., other systems). Scenarios should be described in a manner that will allow engineers to walk through them and gain an understanding of how all the various parts of the system should function and interact as well as verify that the system will satisfy the user's needs and expectations. Operational scenarios should be described for all operational modes, mission phases (e.g., installation, startup, typical examples of normal and contingency operations, shutdown, and maintenance), and critical sequences of activities for all classes of users identified. Each scenario should include events, actions, stimuli, information, and interactions as appropriate to provide a comprehensive understanding of the operational aspects of the system.

Figure 5-6 presents an example of an end-to-end data flow for a scientific satellite mission. Each arrow in the diagram represents one or more data or control flows between two hardware, software, subsystem, or system configuration items. End-to-end testing verifies that the data flows throughout the multi-system environment are correct, that the system provides the required functionality, and that the outputs at the eventual end points correspond to expected results. Since the test environment is as close an approximation as possible to the operational environment, performance requirements testing is also included. This figure is not intended to show the full extent of end-to-end testing. Each system shown would need to be broken down into a further level of granularity for completeness.

End-to-end testing is an integral part of the V&V of the total system, and is an activity that is employed during selected hardware, software, and system phases throughout the life cycle. In comparison with configuration item testing, end-to-end testing addresses each configuration item only down to the level where it interfaces externally to other configuration items, which can be either hardware or software based. Internal interfaces (e.g., software subroutine calls, analog-to-digital conversion, etc.) of a configuration item are not within the scope of end-to-end testing.



**Figure 5-6 Example of End-to-End Data Flow for a Scientific Satellite Mission**

### How to Perform End-to-End Testing

End-to-end testing is probably the most significant element of any project verification program and the test should be designed to satisfy the edict to “test the way we fly”. This means assembling the system in its realistic configuration, subjecting it to a realistic environment and then “flying” it through all of its expected operational modes. For a scientific robotic mission, targets and stimuli should be designed to provide realistic inputs to the scientific instruments. The output signals from the instruments would flow through the satellite data handling system and then transmitted to the actual ground station through the satellite communications system. If data is transferred to the ground station through one or more satellite or ground relays (e.g., the Tracking and Data Relay Satellite, TDRS) then those elements must be included as part of the test.

The end-to-end compatibility test encompasses the entire chain of operations that will occur during all mission modes in such a manner as to ensure that the system will fulfill mission requirements. The mission environment should be simulated as realistically as possible and the instruments should receive stimuli of the kind they will receive during the mission. The RF links, ground station operations, and software functions should be fully exercised. When acceptable simulation facilities are available for portions of the operational systems, they may be used for the test instead of the actual system elements. The specific environments under which the end-to-

end test is conducted and the stimuli, payload configuration, RF links, and other system elements to be used must be determined in accordance with the characteristics of the mission.

Although end-to-end testing is probably the most complex test in any system verification program the same careful preparation is necessary as for any other system-level test. For example, a test lead must be appointed and the test team selected and trained. Adequate time must be allocated for test planning and coordination with the design team. Test procedures and test software must be documented, approved and placed under configuration control.

Plans, agreements, and facilities must be put in place well in advance of the test to enable end-to-end testing between all components of the system. (Note: This is particularly important when missions are developed with international or external partners.)

Once the tests are run the test results are documented and any discrepancies carefully recorded and reported. All test data must be maintained under configuration control.

Before completing end-to-end testing, the following activities are completed for each configuration item:

- a. All requirements, interfaces, states, and state transitions of each configuration item should be tested through the exercise of comprehensive test procedures and test cases to ensure the configuration items are complete and correct.
- b. A full set of operational range checking tests should be conducted on software variables to ensure that the software performs as expected within its complete range and fails, or warns, appropriately for out-of-range values or conditions.

End-to-end testing activities include the following:

- a. Operational scenarios that span all of the following items (during nominal, off-nominal, and stressful conditions) that could occur during the mission:
  1. Mission phase, mode, and state transitions.
  2. First-time events.
  3. Operational performance limits.
  4. Fault protection routines.
  5. Failure detection, isolation, and recovery (FDIR) logic.
  6. Safety properties.
  7. Operational responses to transient or off-nominal sensor signals.
  8. Communication uplink and downlink.
- b. The operational scenarios are used to test the configuration items, interfaces, and end-to-end performance as early as possible in the configuration items' development life cycle. This typically means simulators or software stubs have to be created to implement a full scenario. It is extremely important to produce a skeleton of the actual system to run full scenarios as soon as possible with both simulated/stubbed out and actual configuration items.
- c. A complete diagram and inventory of all interfaces are documented.

- d. Test cases are executed to cover hardware/software, software/software, subsystem/subsystem interfaces and associated inputs, outputs, and modes of operation (including safing modes).
- e. Tests should cover all operational scenarios.
- f. It is strongly recommended that during end-to-end testing, an operations staff member who has not previously been involved in the testing activities be designated to exercise the system as it is intended to be used to determine if it will fail.
- g. The test environment should approximate/simulate the actual operational conditions when possible. The fidelity of the test environment should be authenticated. Differences between the test and operational environment should be documented in the Test or Validation Plan.
- h. When testing of a requirement is not possible, verification is demonstrated by other means (e.g., model checking, analysis, or simulation). If true end-to-end testing cannot be achieved then the testing must be done piecemeal and “patched” together by analysis and simulation. An example of this would be a system that is assembled on orbit where the various elements come together for the first time on orbit.
- i. When an error in the developed system is identified and fixed, regression testing of the system or component is performed to ensure that modifications have not caused unintended effects and that the system or component still complies with previously tested specified requirements.
- j. When tests are aborted or a test is known to be flawed (e.g., due to configuration, test environment), the test should be rerun after the identified problem is fixed.
- k. The operational scenarios should be used to formulate the final Operations Plan.
- l. Prior to system delivery, as part of the system qualification testing, test cases should be executed to cover all of the plans documented in the Operations Plan in the order in which they are expected to occur during the mission.

End-to-end test documentation includes the following:

- a. Inclusion of end-to-end testing plans as a part of the Test or Validation Plan.
- b. A document, matrix, or database under configuration control that traces the end-to-end system test suite to the results. Data that is typically recorded includes the test case identifier, subsystems/hardware/program sets exercised, list of requirements validated, interfaces exercised, date, and outcome of test (i.e., whether the test actual output met the expected output).
- c. End-to-end test cases and procedures (including inputs and expected outputs).
- d. A record of end-to-end problems/failures/anomalies.

End-to-end testing can be integrated with other project testing activities; however, the documentation mentioned in this section should be readily extractable for review, status, and assessment.

#### 5.3.2.6 *Modeling and Simulation*

For the Product Verification Process a model is a physical, mathematical or logical representation of an end product to be verified. M&S can be used to augment and support the Product Verification Process, and is an effective tool performing the verification whether in early life-cycle phases or later. Both the facilities and the model itself are developed using the system design and product realization processes.

**NOTE:** The development of the physical, mathematical or logical model includes evaluating whether the model to be used as representative of the system end product was realized according to its design solution specified requirements for a model, and whether it will be valid for use as a model. In some cases the model must also be accredited to certify the range of specific uses for which the model can be used. Like any other enabling product, budget and time must be planned for creating and evaluating the model to be used to verify the applicable system end product.

The model used, as well as the M&S facility, is an enabling product and must use the seventeen Technical Processes (ref: NPR 7123.1) for their development and realization (including acceptance by the operational community) that the model and simulation adequately represent the operational environment and performance of the modeled end product. Additionally, in some cases, *accreditation* is required before models and simulations can be used.

M&S assets can come from a variety of sources; for example, contractors, other government agencies, or laboratories can provide models that address specific systems attributes.

#### 5.3.2.7 *Hardware-in-the-Loop*

Fully functional end products, an actual piece of hardware, may be combined with models and simulations that simulate the inputs and outputs of other end products of the system. This is referred to as “hardware-in-the-loop” (HWIL) testing. HWIL testing links all elements (subsystems and test facilities) together within a synthetic environment to provide a high fidelity real-time operational evaluation for the real system or subsystems. The operator can be intimately involved in the testing, and HWIL resources can be connected to other facilities for distributed test and analysis applications. One of the uses of HWIL testing is to get as close to the actual concept of operation as possible to support verification and validation when the operational environment is difficult or expensive to recreate.

During development, this HWIL verification normally takes place in an integration laboratory or test facility. For example, HWIL could be a complete spacecraft in a special test chamber, with the inputs/outputs being provided as output from models that simulate the system in an operational environment. Real-time computers are used to control the spacecraft and subsystems in projected operational scenarios. Flight dynamics, responding to the commands issued by the guidance and control system hardware/software, are simulated in real-time to determine the trajectory and to calculate system flight conditions. HWIL testing verifies that the end product being evaluated meets the interface requirements, properly transforming inputs to required outputs. HWIL modeling can provide a valuable means of testing physical end products lower in

the system structure by providing simulated inputs to the end product or receiving outputs from the end product to evaluate the quality of those outputs. This tool can be used throughout the life cycle of a program or project. The space shuttle program uses a HWIL to verify software and hardware updates for the control of the shuttle main engines.

Modeling, simulation, and hardware/human-in-the-loop technology, when appropriately integrated and sequenced with testing, provide a verification method at a reasonable cost. This integrated testing process specifically 1) reduces the cost of life-cycle testing, 2) provides significantly more engineering/performance insights into each system evaluated, and 3) reduces test time and lowers project risk. This process also significantly reduces the number of destructive tests required over the life of the product. The integration of modeling and simulation into verification testing provides insights into trends and tendencies of system and subsystem performance that might not otherwise be possible due to hardware limitations.

## 5.4 Product Validation

The Product Validation Process is the second of the verification and validation (V&V) processes conducted on a realized end product. While verification proves whether “the system was done right,” validation proves whether “the right system was done.” In other words, verification provides objective evidence that every “*shall*” was met, whereas validation is performed for the benefit of the customers and users to ensure that the system functions in the expected manner when placed in the intended environment. This is achieved by examining the products of the system at every level of the structure.

Validation confirms that realized end products at any position within the system structure conform to their set of stakeholder expectations, and ensures that any anomalies discovered during validation are appropriately resolved prior to product delivery. This section discusses the process activities, types of validation, inputs and outputs, and potential deficiencies.

### **Distinctions Between Verification and Validation**

From a process perspective, Product Verification and Product Validation may be similar in nature, but the objectives are fundamentally different.

From a customer point of view, the interest is in whether the end product provided will do what they intend within the environment of use. It is essential to confirm that the realized product is in conformance with its specifications and design description documentation in that such specifications and documents will establish the configuration baseline of that product which may have to be modified at a later time. Without a verified baseline and appropriate configuration controls such later modifications could be costly or cause major performance problems.

When cost-effective and warranted by analysis, various “combined tests” are used. The expense of validation testing alone can be mitigated by assuring that each end product in the system structure was correctly realized in accordance with its specified requirements before conducting validation.

### Differences between Verification and Validation Testing:

**Verification Testing:** Verification testing relates back to the approved requirements set (such as a Systems Requirements Document) and can be performed at different stages in the product life cycle. Verification Testing includes: (1) any testing used to assist in the development and maturation of products, product elements, or manufacturing or support processes; and/or (2) any engineering-type test used to verify status of technical progress, to verify that design risks are minimized, to substantiate achievement of contract technical performance, and to certify readiness for initial validation testing. Verification tests use instrumentation and measurements, and are generally accomplished by engineers, technicians, or operator-maintainer test personnel in a controlled environment to facilitate failure analysis.

**Validation Testing:** Validation relates back to the Concept of Operations document. Validation testing is conducted under realistic conditions (or simulated conditions) on any end product for the purpose of determining the effectiveness and suitability of the product for use in mission operations by typical users; and the evaluation of the results of such tests. Testing is the detailed quantifying method of both verification and validation. However, testing is required to validate final end products to be produced and deployed.

#### 5.4.1 Process Description

Figure 5-7 provides a typical flow diagram for the Product Validation Process and identifies typical inputs, outputs, and activities to consider in addressing product validation.

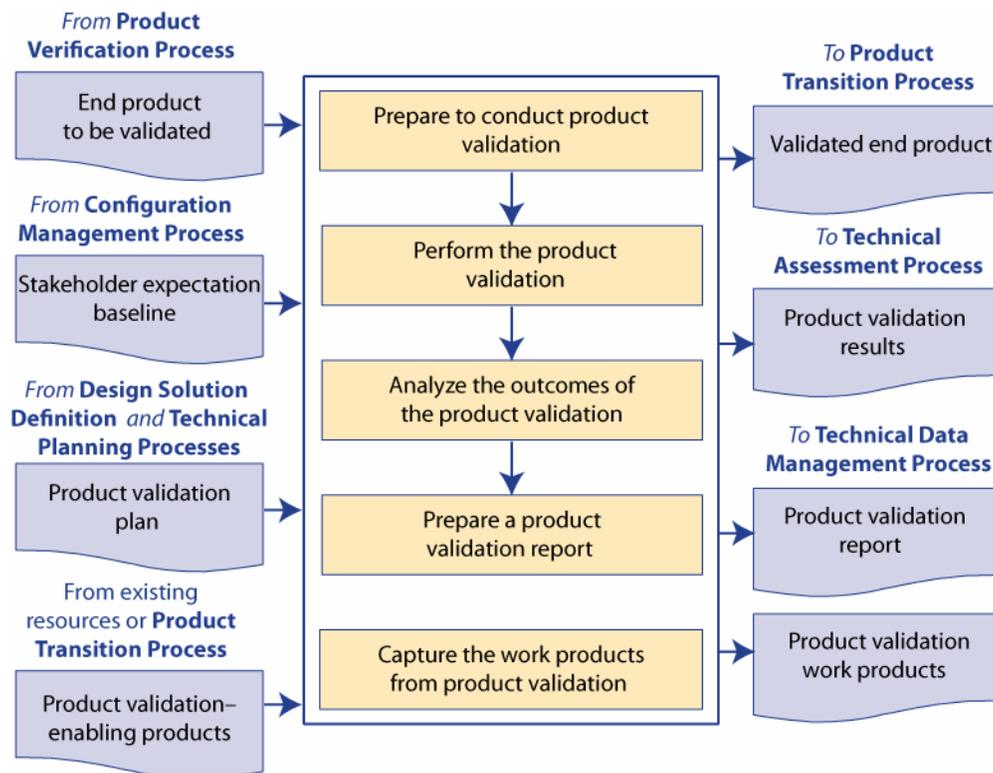


Figure 5-7 Product Validation Process

##### 5.4.1.1 Inputs

Key inputs to the process are:

- Verified product
- Verification Plan
- Baselined stakeholder expectations
- Any enabling products needed to perform the Product Validation Process.

#### 5.4.1.2 *Process Activities*

The Product Validation Process demonstrates that the realized end product satisfies its stakeholder (customer and other interested party) expectations within the intended operational environments, with validation performed by anticipated operators and/or users. The type of validation is a function of the life-cycle phase and the position of the end product within the system structure.

There are five major steps in the validation process: 1) Validation Planning (Prepare to Implement the Validation Plan); 2) Validation Preparation (Prepare for Conducting Validation); 3) Conduct Planned Validation (Perform Validation); 4) Analyze Validation Results; and 5) Capture the Validation Work Products.

The objectives of the Product Validation Process are:

- To confirm:
  - That the right product was realized—the one wanted by the customer;
  - That the realized product can be used by intended operators/users; and
  - That the MOEs are satisfied.
- To confirm that the realized product fulfills its intended use when operated in its intended environment.
  - Validation is performed for each realized (implemented or integrated) product from the lowest end product in a system structure branch up to the top WBS model End Product.
  - Evidence is generated as necessary to confirm that products at each layer of the system structure meet the capability and other operational expectations of the customer/user/operator and other interested parties.
- To ensure that any problems discovered are appropriately resolved:
  - Prior to delivery of the realized product (if validation is done by the supplier of the product); or
  - Prior to integration with other products into a higher-level assembled product (if validation is done by the receiver of the product).

V&V events are illustrated as separate processes, but, when used, there can be considerable overlap between some events. When cost-effective and warranted by analysis, various “combined tests” are used; however, while from a process perspective Verification and Validation are similar in nature, their objectives are fundamentally different.

From a customer's point of view, the interest is in whether the end product provided will supply the needed capabilities within the intended environments of use. The expense of validation testing alone can be mitigated by assuring that the each end product in the system structure was correctly realized in accordance with its specified requirements prior to validation, during verification. It is possible that the system design was not done properly and, even though the verification tests were successful (satisfying the specified requirements), the validation tests would still fail (stakeholder expectations not satisfied). Thus, it is essential that validation of lower products in the system structure be conducted as well as verification so as to catch design failures or deficiencies as early as possible.

### **Product Validation Planning**

Planning to conduct the product validation is a key first step. The type of validation to be used (e.g., analysis, demonstration, inspection, or test) should be established based on the form of the realized end product, the applicable life-cycle phase, cost, schedule, resources available, and location of the system product within the system structure. (See Appendix 5-4 for a sample validation plan outline.)

An established set or subset of requirements to be validation should be identified and the Validation Plan reviewed (an output of the Technical Planning Process, based on design solution outputs) for any specific procedures, constraints, success criteria, or other validation requirements. The conditions and environment under which the product is to be validated should be established and the validation planned based on the relevant life-cycle phase and associated success criteria identified. The Decision Analysis Process should be used to help finalize the planning details.

It is important to review the validation plans with relevant stakeholders and understand the relationship between the context of the validation and the context of use (human involvement). As part of the planning process, validation enabling products should be identified scheduling and/or acquisition initiated.

Procedures should be prepared to conduct validation based on the type (e.g., Analysis, Inspection, Demonstration, or Test) planned. These procedures are typically developed during the design phase of the project life cycle and matured as the design is matured. Operational/use scenarios are thought through so as to explore all possible validation activities to be performed.

### **Validation Plan and Methods**

The Validation Plan is one of the work products of the Technical Planning Process and generated during the design solution to validate the realized product against the baselined stakeholder expectations. This plan can take many forms. The plan describes the total test and evaluation (T&E) planning from development of lower through higher end products in the system structure and through operational T&E into production and acceptance. It may include the plan for Verification and be called the Verification and Validation Plan (see Appendix 5-4 for a sample V&V Plan Outline)

The types of validation include 1) test, 2) demonstration, 3) inspection, and 4) analysis. While the name of each method is the same as the name of the methods for verification, the purpose and intent is quite different.

#### Types of Validation

**Analysis:** The use of mathematical modeling and analytical techniques to predict the compliance of a design to its requirements based on calculated data or data derived from lower system structure end product validations. It is generally used when a prototype, engineering model or fabricated, assembled and integrated product is not available. Analysis includes the use of both modeling and simulation (M&S).

**Demonstration:** The use of a realized end product to show that a set of stakeholder expectations can be achieved. It is generally used for a basic confirmation of performance capability and is differentiated from testing by the lack of detailed data gathering. Validation is done under realistic conditions for any end product within the system structure for the purpose of determining the effectiveness and suitability of the product for use in NASA missions or mission support by typical users; and the evaluation of the results of such tests.

**Inspection:** The visual examination of a realized end product. It is generally used to validate physical design features or specific manufacturer identification.

**Test:** The use of a realized end product to obtain detailed data to validate performance or to provide sufficient information to validate performance through further analysis. Testing is the detailed quantifying method of both verification and validation but it is required in order to validate final end products to be produced and deployed.

Validation is conducted by the user/operator or by the developer, as determined by NASA Center directives or the contract with the developers. Systems-level validation (e.g., customer T&E and some other types of validation) may be performed by an acquirer testing organization. For those portions of validation performed by the developer, appropriate agreements must be negotiated to ensure that validation proof-of-documentation is delivered with the realized product.

All realized end products, regardless of the source (buy, make, reuse, assembly and integrate) and the position in the system structure, should be validated to demonstrate/confirm satisfaction of stakeholder requirements. Variations, anomalies, and out-of-compliance conditions, where such have been detected, are documented along with the actions taken to resolve the discrepancies. Validation is typically carried out in the intended operational environment under simulated or actual operational conditions; not under the controlled conditions usually employed for the Product Verification Process.

Validation can be performed recursively throughout the project life cycle and on a wide variety of product forms. For example:

- Simulated (algorithmic models, virtual reality simulator);
- Mock-up (plywood, brassboard, breadboard);
- Concept description (paper report); and
- Prototype (product with partial functionality)
- Engineering Unit (fully functional but may not be same form/fit)

- Design Validation Test Units (form, fit and function same, but may not have flight parts);
- Qualification units (Identical to flight units but may be subjected to extreme environments)
- Flight units (end product that is flown)

Any of these types of product forms may be in any of these states

- Produced (built, fabricated, manufactured, or coded);
- Reused (modified internal non-developmental products or off-the-shelf product); and
- Assembled and integrated (a composite of lower level products).

Note: the final, official validation of the end product should be for a controlled unit. Typically, attempting final validation against operational concepts on a prototype is not acceptable; it is usually completed on a qualification, flight, or other more-final, controlled unit.

Outcomes of validation planning include the following:

- The validation type that is appropriate to confirm that the realized product or products conform to stakeholder requirements (based on the form of the realized end product) has been identified.
- Defined validation procedures based on: (1) the needed procedures for each type of validation selected; (2) the purpose and objective of each procedure step; (3) any pre-test and post-test actions; and (4) the criteria for determining the success or failure of the procedure.
- A defined validation environment (e.g., facilities, equipment, tools, simulations, measuring devices, personnel and operational conditions) in which the validation procedures will be implemented.

**NOTE:** In planning for validation, consideration should be given to the extent to which validation testing will be done. In many instances, off-nominal operational scenarios AND nominal operational scenarios should be utilized. Off-nominal testing offers insight into a systems total performance characteristics and often assists in identification of design issues and human-machine interface, training, and procedural changes required to meet the mission goals and objectives. Off-nominal testing, as well as nominal testing, should be included when planning for validation.

### **Product Validation Preparation**

In preparation of performing product validation, the appropriate set of expectations against which the validation is to be made should be obtained and the product to be validated (output from implementation, or integration and verification), as well as the validation enabling products and support resources (requirements identified and acquisition initiated by design solution activities) with which validation will be conducted are collected.

### **Examples of enabling products and support resources for “Prepare for Conducting Validation”**

One of the key tasks in the Product Validation Process of “Prepare for Conducting Validation” is to obtain necessary enabling products and support resources needed to conduct validation. Examples of these include:

- Measurement tools (scopes, electronic devices, probes);
- Embedded test software;
- Test wiring, measurement devices, and telemetry equipment;
- Recording equipment (capture test results);
- End products in the loop (by software, electronics, or mechanics) for hardware-in-the-loop simulations;
- External interfacing products of other systems;
- Actual external interfacing products of other systems (aircraft, vehicles, humans); and
- Facilities and skilled operators.

The validation environment is then prepared (set up the equipments, sensors, recording devices, etc. that will be involved in the validation conduct) and the validation procedures reviewed to identify and resolve any issues impacting validation.

Outcomes of validation preparation include the following:

- Preparation for doing the planned validation is completed
- Appropriate set of stakeholder requirements or applicable subset of requirements are available and on hand.
- Articles or models to be used for validation with the validation product and enabling products integrated within the validation environment according to plans and schedules.
- Resources are available according to validation plans and schedules.
- Validation environment is checked-out for adequacy, completeness, readiness and integration.

### **Conduct Planned Product Validation**

The actual act of Validating the end product is conducted as spelled out in the Validation Plans and Procedures and conformance established to each specified validation requirement. The responsible engineer should ensure that the procedures were followed and performed as planned, the validation enabling products were calibrated correctly, and the data was collected and recorded for required validation measures.

When poor validation conduct, design, or conditions cause anomalies, the validation should be re-planned as necessary, the environment preparation anomalies corrected, and the validation conducted again with improved or correct procedures and resources. The Decision Analysis Process should be used to make decisions for issues identified that may require alternative choices to be evaluated and a selection made, or when needed changes to the validation plans, environment, and/or conduct are required.

Outcomes of conducting validation include the following:

- A validated product is established with supporting confirmation that the appropriate results were collected and evaluated to show completion of validation objectives.
- A determination as to whether the fabricated/manufactured, or assembled and integrated products (including software or firmware builds, as applicable) comply with their respective stakeholder requirements.
- A determination that the validated product was appropriately integrated with the validation environment, and the selected stakeholder requirements set was properly validated.
- A determination that the product being validated functions together with interfacing products throughout their performance envelopes.

### **Analyze Product Validation Results**

Once the validation activities have been completed, the results are collected and the data is analyzed to confirm that validation procedures were followed, and that enabling products and supporting resources functioned correctly. The data is also analyzed for quality, integrity, correctness, consistency and validity, and any unsuitable products or product attributes are identified and reported.

It is important to compare the actual validation results to the expected results and to conduct any required system design and product realization process activities to resolve deficiencies. The deficiencies, along with recommended corrective actions and resolution results, should be recorded and validation repeated, as required.

Outcomes of analyzing validation results include the following:

- Identification of product deficiencies and/or issues.
- A determination that design and re-validation have been accomplished to resolve deficiencies not caused by poor validation conduct or conditions, appropriate re-planning, redefinition of requirements, approval of changed requirements,.
- Discrepancy and Corrective Action Reports have been generated as needed.
- The Validation Report is completed.

### **Validation Notes**

The types of validation used are dependent on the life-cycle phase, the product's location in the system structure, and cost, schedule and resources available. Validation of products within a single system model may be conducted together (e.g., an end product with its relevant enabling products such as operation (control center), maintenance (required tools work with product) or logistics (launcher or transporter) or a radar with its display).

Each realized product of system structure should be validated against stakeholder expectations before being integrated into a higher-level product.

## **Reengineering**

Based on the results of the Product Validation Process, it could become necessary to re-engineer a deficient end product. Care should be taken in that correcting a deficiency, or set of deficiencies, does not generate a new issue with a part or performance that had previously operated satisfactorily. Regression testing, a formal process of rerunning previously used acceptance tests, and primarily utilized for software, is one method utilized to ensure a change did not affect function/performance that was previously accepted.

## **Validation Deficiencies**

Validation outcomes can be unsatisfactory for several reasons. One reason is poor conduct of the validation (e.g., enabling products and supporting resources missing or not functioning correctly, untrained operators, procedures not followed, equipment not calibrated, improper validation environmental conditions), and failure to control other variables not involved in validating a set of stakeholder requirements. A second reason could be that the realized end product used was not realized correctly. This could create the need for:

- Re-engineering end products lower in the system structure that make up the end product that was found to be deficient (which failed to satisfy validation requirements); and/or
- Re-performing any needed V&V processes.

**NOTE:** Care should be exercised to ensure that the corrective actions identified to remove validation deficiencies do not conflict with the baselined stakeholder requirements without first coordinating such changes with the appropriate stakeholders.

## **Capture Product Validation Work Products**

Validation work products (inputs to the Technical Data Management Process) take many forms and involved many sources of information. The capture and recording of validation-related data is a very important, but often under-emphasized, step in the Validation Process.

Validation results, deficiencies identified, and corrective actions taken should be capture, as should all relevant results from the application of the Product Validation Process (related decisions, rationale for decisions made, assumptions, and lessons learned).

Outcomes of capturing validation work products include the following:

- Work products and related information generated while doing Product Validation Process activities and tasks are recorded; i.e., type of validation conducted, the form of the end product used for validation, validation procedures used, validation environments, outcomes, decisions, assumptions, corrective actions, lessons learned, etc. (often captured in a matrix or other tool—see Appendix 4-2c).

- Deficiencies (e.g., variations and anomalies and out-of-compliance conditions) have been identified and documented, including the actions taken to resolve:
- Proof that the realized product is in conformance with the stakeholder requirement set used in the validation.
- Recorded validation result data that includes the following:
  - Version of the set of stakeholder requirements used;
  - Version and form of the end product validated;
  - Version or standard for tools and equipment used, together with applicable calibration data;
  - Outcome of each validation including pass or fail declarations; and
  - Discrepancy between expected and actual results.

#### *5.4.1.3 Outputs*

Key outputs of validation are:

- Validated product
- Discrepancy reports and Identified Corrective Actions
- Validation reports

Exit criteria for this process includes: (1) Objective evidence of performance and the results of each system-of-interest validation activity are documented, and (2) The validation process should not be considered or designated as complete until all issues and actions are resolved.

### *5.4.2 Product Validation Guidance*

The following is some generic guidance for the Validation Process.

#### *5.4.2.1 Modeling and Simulation*

As stressed in the verification process material, Modeling and Simulation (M&S) is also an important validation tool. M&S usage considerations involve model verification, validation and accreditation (VV&A).

## VV&A

**Model Verification:** Degree to which a model accurately meets its specifications. Answers: “Is it what I intended?”

**Model Validation:** Accurate real-world fidelity. It encompasses the model, data and even the operators and analysts who will use the model or its results. Answers: “How well does the model represent what it claims to represent?”

**Model Accreditation:** Certification for use for a specific purpose. Answers: “Should I endorse this model?”

### 5.4.2.2 Software

The purpose of software V&V is the same as discussed in this and the previous lesson for system products. Complete V&V of complex software is particularly difficult given real world schedule and resource constraints; however, lack of systems and software engineering rigor in software development is often a contributing factor in failure of mission-critical or safety-critical software intensive systems.

Historically, for many mission-critical or safety-critical software intensive systems, the need to *independently* evaluate the state and progress of software development arose, leading to the growth of disciplines such as software quality assurance (SQA) and independent verification and validation (IV&V). IV&V is verification and validation performed by an individual or organization that is technically, managerially, and financially *independent* of the development organization. IV&V adds the independent evaluation to the standard software V&V in order to ensure a quality software product. IV&V is sometimes mandated by the agency for specific projects, and knowledge of this can be used in risk trades.

A project must balance its risk posture in software product quality between its V&V, SQA, and IV&V activities.

## 5.5 Product Transition

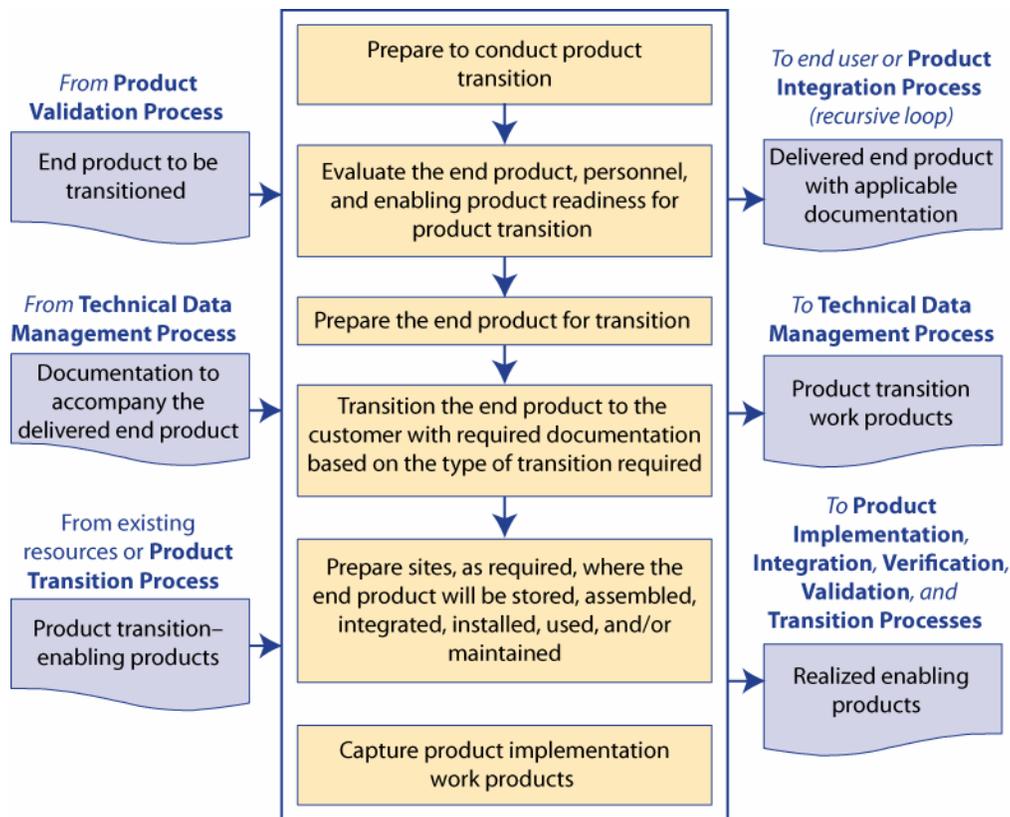
Product transition occurs during all phases of the life cycle. During the early phases the technical teams' products are documents, models, studies and reports. As the project moves through the life cycle these "paper" or "soft products" are transformed through implementation and integration processes into hardware and software solutions to meet the stakeholders' expectations. They are repeated with different degrees of rigor throughout the life cycle. The Transition Process includes product transitions from one level of the system architecture upward. The Transition Process is the last of the product realization processes, and it is a bridge from one level of the system to the next higher level.

The product transition process is used to transition a verified and validated end product that has been generated by product implementation or product integration to the customer at the next level in the system structure for integration into an end product or, for the top level end product, transitioned to the intended end user. The form of the product transitioned will be a function of the product-line life-cycle phase exit criteria and the location within the system structure of the WBS model in which the end product exits.

The product transition process is the key to bridge from one activity, subsystem, or element to the overall engineered system. As the system development nears completion, the End Product transition process is again applied, but with much more rigor since now the transition objective is delivery of the system-level end product to the actual end-user. Depending on the kind or category of system developed, this may involve a Center or the Agency and impact thousands of individuals storing, handling, and transporting multiple end products, in user site preparation, training operators and maintenance personnel, and installation and sustainability, as applicable. Examples are transitioning the external tank, SRBs, and orbiter to KSC for integration and flight.

### 5.5.1 Process Description

Figure 5-8 provides a typical flow diagram for the Product Transition Process and identifies typical inputs, outputs, and activities to consider in addressing product transition.



**Figure 5-8 Product Transition Process**

### 5.5.1.1 Inputs

Inputs to the Product Transition Process depend primarily on the transition requirements, the product that is being transitioned, the form of the product transition that is taking place, and where the product is transitioning to. Typical inputs are shown in Figure 5-8.

- The End Product or Products to Be Transitioned (From Product Validation Process):** The product to be transitioned can take several forms. It can be a subsystem component, system assembly, or top level end product. It can be hardware or software. It can be newly built, purchased, or reused. A product can transition from a lower system product to a higher one by being integrated with other transitioned products. This process may be repeated until the final end-product is achieved. Each succeeding transition requires unique input considerations when preparing for the validated product for transition to the next level.

Early phase products can take the form of information or data generated from basic or applied research using analytical or physical models and often are in paper or electronic form. In fact the end-product for many NASA research projects or science activities is a report, paper, or even an oral presentation. In a sense, the dissemination of information gathered through NASA research and development is an important form of product transition.

- **Documentation Including Manuals, Procedures And Processes That Are to Accompany the End Product (From Technical Data Management Process):** The documentation required for Product Transition depends on the specific end product, its current location within system structure, and the requirements identified in various agreements, plans or requirement documents. Typically a product has a unique identification (i.e., serial number) and may have a pedigree (documentation) that specifies the product's heritage and current state. Pertinent information may be documented through a configuration management system or work order system as well as in design drawings and test reports. Documentation often includes proof of verification and validation conformance. A COTS product would typically contain a manufacturer's specification or fact sheet. Documentation may include operations manuals, installation instructions, and other information.

The documentation level of detail is dependent upon where the product is within the product hierarchy and the life cycle. Early in the life cycle, this documentation may be preliminary in nature. Later in the life cycle, the documentation may be detailed design documents, user manuals, drawings, or other work products. Documentation that is gathered during the input process for the transition phase may require editing, assembling, or repackaging to ensure it is in the required condition for acceptance by the customer.

Special consideration to safety must be considered, including clearly identifiable tags and markings that identify the use of hazardous materials, special handling instructions, and storage requirements.

- **Product Transition Enabling Products to Include Packaging Materials, Containers, Handling Equipment, and Storage, Receiving and Shipping Facilities (From Existing Resources or Product Transition Process for Enabling Product Realization):** Product transition enabling products may be required to facilitate the implementation, integration, evaluation, transition, training, operations, support, and/or retirement of the transition product at its next higher level or for the transition of the final end-product. Some or all of the enabling products may be defined in transition related agreements, systems requirement documents or project plans. In some cases, product transition enabling products are developed during the realization of the product itself or may be required to be developed during the transition stage.

As a product is developed, special containers, holders or other devices may also be developed to aid in the storing and transporting of the product through development and realization. These may be temporary accommodations that do not satisfy all the transition requirements but allow the product to be initiated into the transition process. In such case the temporary accommodations will have to be modified or new accommodations will need to be designed and built or procured to meet specific transportation, handling, storage, and shipping requirements.

Sensitive or hazardous products may require special enabling products; such as monitoring equipment, inspection devices, safety devices, and personnel training to ensure adequate safety and environmental requirements are achieved and maintained.

### 5.5.1.2 Process Activities

Transitioning the product can take one of two forms:

- The delivery of lower system end products to higher ones for integration into another end product
- The delivery of the final end product to the customer or user that will use the end product in its operational environment.

In the first case, the end product is one of perhaps several other pieces that will ultimately be integrated together to form the item in the second case for final delivery to the customer. For example, the end product might be one of several circuit cards that will be integrated together to form the final unit that is delivered. Or that unit might also be one of several units that have to be integrated together to form the final product.

The form of the product transitioned is not only a function of the location of that product within the system product hierarchy (i.e., WBS model), but also a function of the life-cycle phase. Early life-cycle phase products may be in the form of paper, electronic files, physical models, or technology demonstration prototypes. Later phase products may be pre-production prototypes (engineering models), the final study report, or the flight units.

Figure 5-8 shows what kind of inputs, outputs, and activities that are performed during product transition regardless of where in the product hierarchy or life cycle it is. These activities include preparing to conduct the transition, making sure the end product, all personnel and any enabling products are ready for transitioning, preparing the site, and performing the transition including capturing/documenting all work products.

How these activities are performed and what form the documentation takes will depend on where the end items are in the product hierarchy (WBS model) and its life-cycle phase.

#### **Prepare to Implement Transition**

The first task is to identify which of the two forms of transition is needed: 1) The delivery of lower system end products to higher ones for integration into another end product or 2) the delivery of the final end product to the customer or user that will use the end product in its operational environment. The form of the product being transitioned will affect transition planning and the kind of packaging, handling, storage, and transportation that will be required. The customer and other stakeholder expectations, as well as the specific design solution may indicate special transition procedures or enabling product needs for packaging, storage, handling, shipping/transporting, site preparation, installation and/or sustainability. These requirements need to be reviewed during the preparation stage.

Another task in preparing to transition a product is to make sure the end product, personnel and any enabling products are ready for that transition. This includes the availability of the documentation that will be sent with the end product, including proof of verification and validation conformance. The appropriateness of detail for that documentation is dependent upon where the product is within the product hierarchy and the life cycle. Early in the life cycle, this

documentation may be preliminary in nature. Later in the life cycle, the documentation may be detailed design documents, user manuals, drawings or other work products. Procedures necessary for conducting the transition should be reviewed and approved by this time. This includes all necessary approvals by management, legal, safety, quality, property, or other organizations as identified in the SEMP.

Finally the availability and skills of personnel needed to conduct the transition as well as the availability of any necessary packaging materials/containers, handling equipment, storage facilities, and shipping/transporter services should also be reviewed. Any special training necessary for the personnel to perform their tasks needs to be performed by this time.

### **Prepare the Product for Transition**

Whether transitioning a product to the next room for integration into the next higher assembly, or for final transportation across the country to the customer, care must be taken to ensure the safe transportation of the product. The requirements for packaging, handling, storage, and transportation should have been identified during system design. Preparing for the packaging for protection, security, and prevention of deterioration is critical for products placed in storage or when it is necessary to transport or ship between and within organizational facilities or between organizations by land, air and/or water vehicles. Particular emphasis needs to be on protecting surfaces from physical damage; preventing corrosion, eliminating damage to electronic wiring or cabling, shock or stress damage, heat warping or cold fractures, moisture, and other particulate intrusion that could damage moving parts.

The design requirements should have already addressed the ease of handling or transporting the product such as component staking, additional of transportation hooks, crating, etc. The ease and safety of packing and unpacking the product should also have been addressed. Additional measures may also need to be implemented to show accountability and to securely track the product during transportation. In cases where hazardous materials are involved, special labeling or handling needs including transportation routes need to be in place.

### **Prepare the Site to Receive Product**

For either of the forms of product transition, the receiving site needs to be prepared to receive the product. Here the end product will be stored, assembled, integrated, installed, used, and/or maintained, as appropriate for the life-cycle phase, position of the end product in the system structure, and customer agreement.

A vast number of key complex activities many of them outside direct control of the technical team, have to be synchronized to ensure smooth transition to the end user. If transition activities are not carefully controlled, there can be impacts on schedule, cost, and safety of the end product.

A site survey may need to be performed to determine the issues and needs. They should address the adequacy of existing facilities to accept, store, and operate the new end product and identify any logistical support enabling products and services required but not planned for. Additionally, any modifications to existing facilities must be planned well in advance of fielding; therefore the site survey should be made during an early phase in the product life cycle. These may include

logistical enabling products and services to provide support for end product use, operations, maintenance, and disposal. Training for users, operators, maintainers and other support personnel may need to be conducted. Environmental impact studies or approvals may need to be obtained prior to the receipt of the end product.

### **Transition the Product**

Prior to shipment or after receipt, the end product may need to be stored in suitable storage conditions to protect, secure, and prevent damage or deterioration of the product. These conditions should have been identified early in the design life cycle and the receiving site prepared for during the “Prepare the site” activity.

Transition the end product (i.e., move, transport, or ship) with required documentation to the customer based on the type of transition required, e.g., to the next higher level WBS model for product integration or to the end user. Documentation may include operations manuals, installation instructions and other information.

Install the end product into the next higher assembly or into the customer/user site using the pre-approved installation procedures.

### **Confirm Ready to Support**

After installation, whether into the next higher assemble or into the final customer site, functional and acceptance testing of the end product should be conducted. This ensures no damage from the shipping/handling process has occurred and that the product is ready for support. Any final transitional work products should be captured as well as documentation of product acceptance.

#### *5.5.1.3 Outputs*

- **Delivered End Product for Integration to Next Level up in System Structure:** This includes the appropriate documentation, and the form of the end product and applicable documentation is a function of the life-cycle phase and the placement within the system structure (the form of the end product could be hardware, software; model, prototype, first article for test; or single operational article or multiple production article). Documentation includes applicable draft installation, operation, user, maintenance, or training manuals, applicable baseline documents (configuration baseline, specifications, and stakeholder expectations) and test results that reflect completion of verification and validation of the end product.
- **Delivered Operational End Product for End Users:** The appropriate documentation is to be delivered with the delivered end product as well as the operational end product appropriately packaged. Documentation includes applicable final installation, operation, user, maintenance, or training manuals, applicable baseline documents (configuration baseline, specifications, stakeholder expectations) and test results that reflect completion of verification and validation of the end product. If the end user will perform end product validation, sufficient documentation to support end user validation activities is delivered with the end product.

- **Work Products from Transition Activities to Technical Data Management:** Work products could include the transition plan, site surveys, measures, training modules, procedures, decisions, lessons learned, corrective actions, etc.
- **Realized Enabling End Products to Appropriate Life-Cycle Support Organization:** Some of the enabling products that were developed during the various phases could include fabrication or integration specialized machines, tools, jigs, fabrication processes and manuals, integration processes and manuals specialized inspection, analysis, or test equipments, tools, test stands, manuals specialized packaging materials and containers, handling equipments, storage site environments, shipping or transportation vehicles or equipments, specialized courseware, instructional site environments, manuals, and delivery of the training instruction. For the latter life-cycle phases, enabling products that are to be delivered may include specialized mission control equipments, data collection equipments, data analysis equipments, operations manuals, specialized maintenance equipment, tools, manuals, and spare parts specialized recovery equipment, disposal equipment, and readying recovery or disposal site environments.

Process is complete when the following activities have been accomplished.

- End product is validated against stakeholder expectations unless the validation is to be done by the integrator before integration is accomplished
- For deliveries to the integration path, end product is delivered to intended usage sites in a condition suitable for integration with other end products or composites of end products. Procedures, decisions, assumptions, anomalies, corrective actions, lessons learned, etc. resulting from transition for integration are recorded.
- For delivery to the end user path, also ensure that the end products are installed at the appropriate sites, appropriate acceptance and certification tests are completed, training of users, operators, maintainers and other necessary personnel is completed and that delivery is closed out with appropriate acceptance documentation.
- Any realized enabling end products are also delivered as appropriate including procedures, decisions, assumptions, anomalies, corrective actions, lessons learned, etc., actions, lessons learned, etc. resulting from transition enabling products are recorded.

## **5.5.2 Product Transition Guidance**

### *5.5.2.1 Additional Product Transition Input Considerations*

It is important to consider all customer, technical, programmatic, and safety requirements when evaluating the input necessary to achieve a successful Product Transition Process. This includes the following:

- **Transportability Requirements.** If applicable, requirements in this section define the required configuration of the system of interest for transport. Further, this section details

the external systems (and the interfaces to those systems) required for transport of the system of interest.

- **Environmental Requirements.** Requirements in this section define the environmental conditions under which the system of interest is required to be in during transition (including storage and transportation).
- **Maintainability Requirements.** Requirements in this section detail how frequently, by whom, and by what means the system of interest will require maintenance (also any “care and feeding” if required).
- **Safety Requirements.** Requirements in this section define the life-cycle safety requirements for the system of interest and associated equipment, facilities, and personnel.
- **Security Requirements.** Requirements in this section define the required IT, ITAR, and physical security for the system of interest.
- **Programmatic Requirements.** Requirements in this section define cost and schedule requirements.

#### *5.5.2.2 After Product Transition to the End User What Next?*

As mentioned in chapter 2, there is a relationship between the SE engine and the activities performed after the product is transitioned to the end user. As shown in Figure 2-10, after the final deployment to the end user, the end product is operated, managed and maintained through sustaining engineering functions. The technical management processes described in section 6.0 are used during these activities. If at any time a new capability, upgrade or new enabling product is needed, the developmental processes of the engine are re-engaged. When the end product’s use is completed, the plans developed early in the life cycle to dispose, retire, or phase out the product are enacted.

## 6.0 Crosscutting Technical Management

The technical management processes are the bridges between project management and the technical team. In this portion of the engine, eight crosscutting processes provide the integration of the crosscutting functions that allow the design solution to be realized. Even though every technical team member might not be directly involved with these eight processes, they are indirectly effected by these key functions. Every member of the technical team relies on technical planning; management of requirements, interfaces, technical risk, configuration, and data; technical assessment; and decision analysis to meet the project's objectives. Without these crosscutting processes, individual members and tasks cannot be integrated into a functioning system that meets the concept of operation within cost and schedule. The project management team also uses these crosscutting functions to execute project control on the apportioned tasks.

This effort starts with the technical team conducting extensive planning early in Pre-Phase A. With this early, detailed baseline plan, technical team members will understand the roles and responsibilities of each team member, and the project can establish its program cost and schedule goals and objectives. From this effort, the SEMP is developed and baselined. Once a SEMP has been established, it must be synchronized with the project master plans and schedule. In addition the plans for establishing and executing all technical contracting efforts are identified.

This is a recursive and iterative process. Early in the life cycle the plans are established and synchronized to run the design and realization processes. As the system matures and progresses through the life cycle, these plans must be updated as necessary to reflect the current environment and resources to control the projects performance, cost and schedule. At a minimum these updates will occur at every KDP. However, if there is a significant change in the project, such as new stakeholder expectations, resources adjustments or other constraints, all plans must be analyzed for the impact of these changes to the baselined project.

The next sections describe each of the eight technical management processes and their associated products for a given NASA mission.

### **Crosscutting Technical Management Keys**

- Thoroughly understand and plan the scope of the technical effort by investing time up-front to develop the technical product breakdown structure, the technical schedule and workflow diagrams, and the technical resource requirements and constraints (funding, budget, facilities and long lead items) which will be the technical planning infrastructure.
- Define all interfaces and assign interface authorities and responsibilities to each, both intra- and inter organizational. This includes understanding potential incompatibilities and defining the transition processes.
- Control of the configuration is critical to understanding how changes will impact the system. For example, changes in design and environment, could invalidate previous analysis results.
- Conduct milestone reviews to enable a critical and valuable assessment to be performed. These reviews are not to be used to meet contractual or scheduling incentives. These reviews have specific entrance criteria and should be conducted when these are met.
- Understand any biases, assumptions and constraints that impact the analysis results.
- Place all analysis under configuration control to be able to track the impact of changes and understand when the analysis needs to be re-evaluated.

## 6.1 Technical Planning

The Technical Planning Process, the first of the eight technical management processes contained in the SE engine, establishes a plan for applying and managing each of the common technical processes that will be used to drive the development of system products and associated work products. This process also establishes a plan for identifying and defining the technical effort required to satisfy the project objectives and life-cycle-phase exit criteria within the cost, schedule and risk constraints of the project.

### 6.1.1 Process Description

Figure 6-1 provides a typical flow diagram for the Technical Planning Process and identifies typical inputs, outputs, and activities to consider in addressing technical planning.

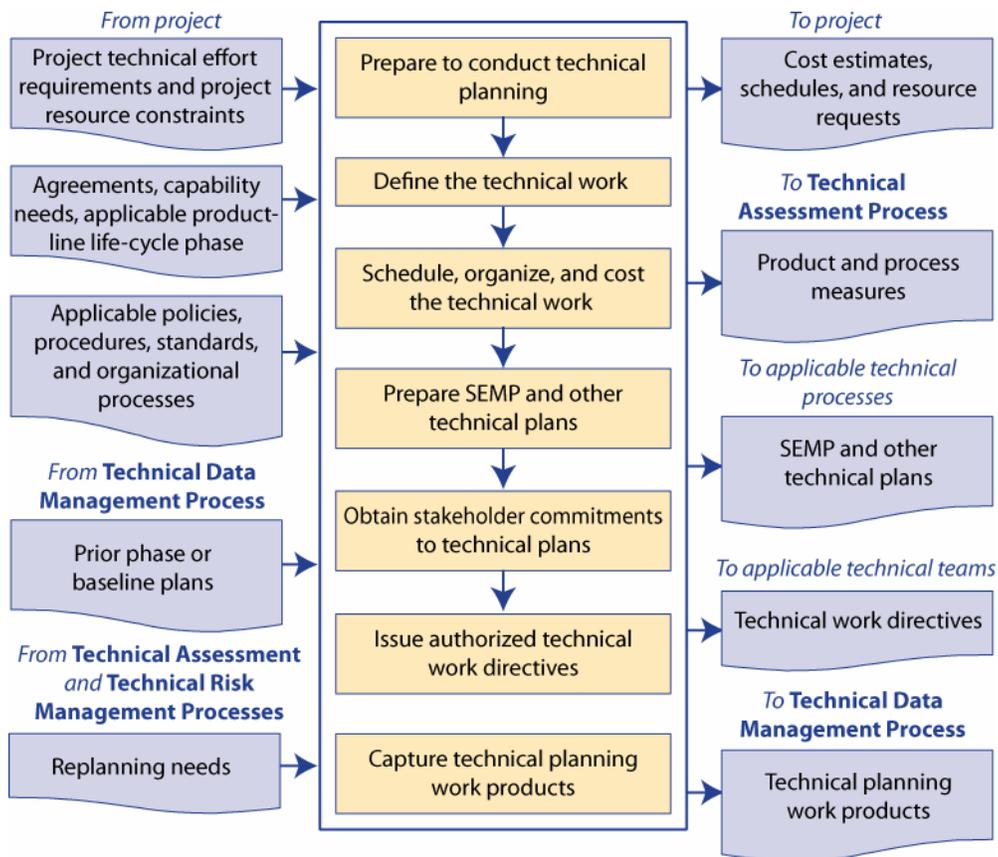


Figure 6-1 Technical Planning Process

#### 6.1.1.1 Inputs

Input to the Technical Planning process comes from both the project management and technical teams as outputs from the other common technical processes. Initial planning utilizing external inputs from the project to determine the general scope and framework of the technical effort will be based on known programmatic requirements, constraints, policies and processes. Throughout

the project's life cycle, the technical team continually incorporates results into the technical planning strategy and documentation and any internal changes based on decisions and assessments generated by the other processes of the SE engine or from requirements and constraints mandated by the project.

- **External Inputs from the Project:** The Project Plan provides the project's top-level technical requirements, the available budget allocated to the project from the program, and the desired schedule for the project to support overall program needs. Although the budget and schedule allocated to the project will serve as constraints on the project, the technical team will generate a technical cost estimate and schedule based on the actual work required to satisfy the project's technical requirements. Discrepancies between the project's allocated budget and schedule and the technical team's actual cost estimate and schedule must be reconciled continuously throughout the project's life cycle.

The Project Plan also defines the applicable project life-cycle phases and milestones, as well any internal and external agreements or capability needs required for successful project execution. The project's life-cycle phases and programmatic milestones will provide the general framework for establishing the technical planning effort and for generating the detailed technical activities and products required to meet the overall project milestones in each of the life-cycle phases.

Finally, the Project Plan will include all programmatic policies, procedures, standards, and organizational processes that must be adhered to during execution of the technical effort. The technical team must develop a technical approach that ensures the project requirements will be satisfied and that any technical procedures, processes, and standards to be used in developing the intermediate and final products comply with the policies and processes mandated in the Project Plan.

- **Internal Inputs from Other Common Technical Processes:** As the project progresses through the life-cycle phases, technical planning for each subsequent phase must be assessed and continually updated. When a project transitions from one life-cycle phase to the next, the technical planning for the upcoming phase must be assessed and updated to reflect the most recent project data. The latest technical plans (either baselined or from the previous life-cycle phase) from the Data Management or Configuration Management processes should be used in updating the technical planning for the upcoming life-cycle phase.

Technical planning updates may be required based on results from technical reviews conducted in the Technical Assessment process, issues identified during the Risk Management process, or from decisions made during the Decision Analysis process.

#### *6.1.1.2 Process Activities*

Technical planning as it relates to systems engineering at NASA is intended to identify, define, and plan how the 17 common technical process in NPR 7123.1 will be applied in each life-cycle phase for all levels of the WBS model (ref. section 6.1.4.1) within the system structure to meet

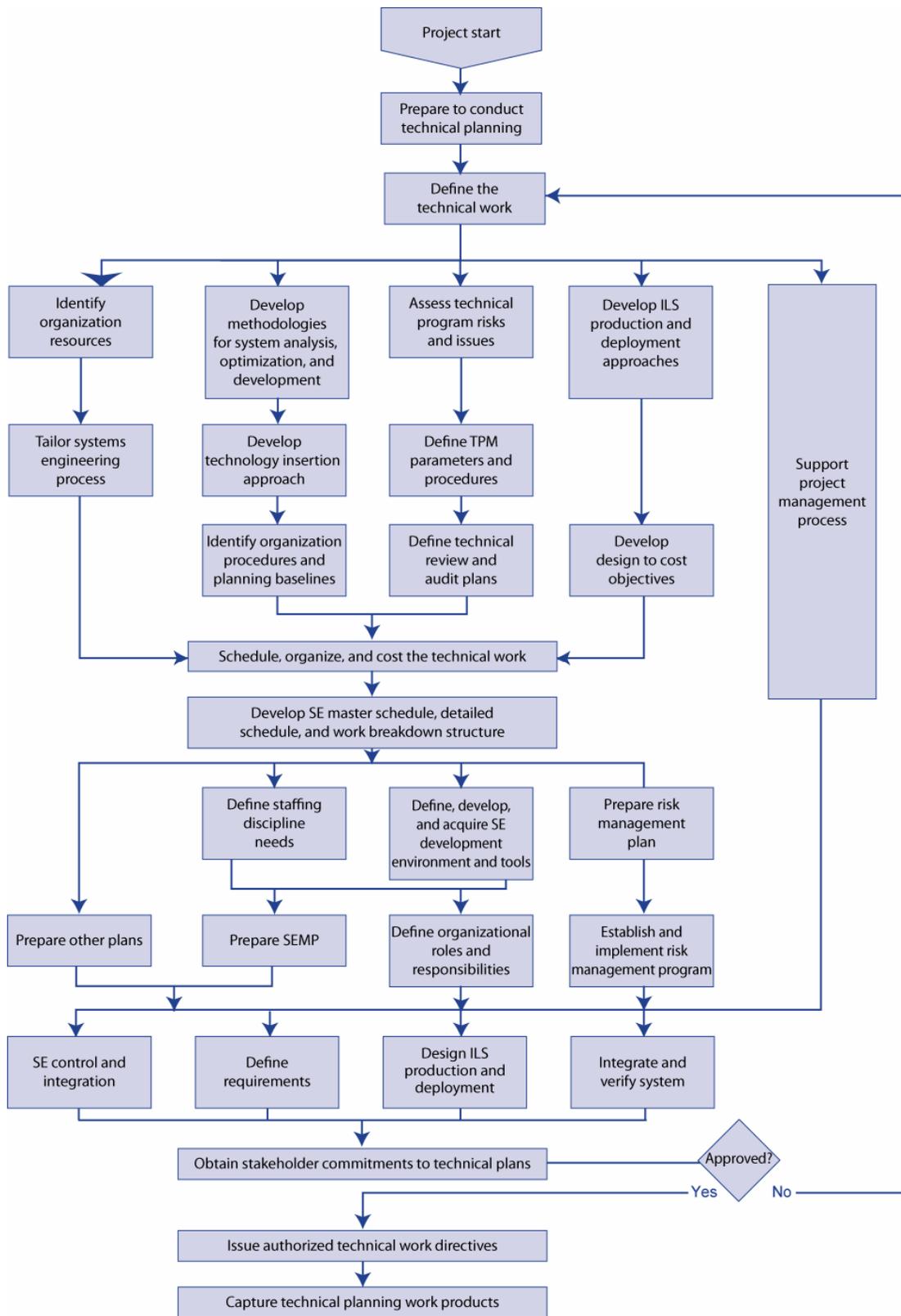
product-line life-cycle-phase exit criteria. A key document generated by this process is the Systems Engineering Management Plan (SEMP).

The SEMP is a subordinate document to the Project Plan. While the SEMP defines to all project participants how the project will be technically managed within the constraints established by the Project, the Project Plan defines how the project will be managed to achieve its goals and objectives within defined programmatic constraints. The SEMP also communicates how the systems engineering management techniques will be applied throughout all phases of the project life cycle.

Technical Planning should be tightly integrated with the Technical Risk Management process (see section 5.4) and the Technical Assessment process (see section 5.7) to ensure corrective action for future activities will be incorporated based on current issues identified within the project.

Technical planning, as opposed to program or project planning, addresses the scope of the technical effort required to develop the system products. While the project manager concentrates on managing the overall project life cycle, the technical team, led by the systems engineer, concentrates on managing the technical aspects of the project. The technical team identifies, defines and develops plans for performing decomposition, definition, integration, verification, and validation of the system while orchestrating and incorporating the appropriate concurrent engineering. Additional planning will include defining and planning for the appropriate technical reviews, audits, assessments and status reports; and determining any specialty engineering and/or design certification requirements.

This section describes how to perform the activities contained in the Technical Planning process shown in Figure 6-1. The initial technical planning at the beginning of the project will establish the technical team members, their roles and responsibilities, and the tools, processes, and resources that will be utilized in executing the technical effort. In addition, the expected activities the technical team will perform and the products that the technical team will produce will be identified, defined and scheduled. Technical planning will continue to evolve as actual data from completed tasks is received and details of near-term and future activities are known. Figure 6-2 shows a generic process flow for technical planning tasks.



**Figure 6-2 Typical Technical Planning Process Flow Tasks**

(adapted from *Systems Engineering Handbook* by J.N. Martin)

## Technical Planning Preparation

For technical planning to be conducted properly, the processes and procedures to conduct technical planning should be identified, defined and communicated. As participants are identified, their roles and responsibilities and any training and/or certification requirements should be clearly defined and communicated.

Once the processes, people, and roles and responsibilities are in place, a planning strategy may be formulated for the technical effort. A basic technical planning strategy should address the following:

- a. the level of planning documentation required for the SEMP and all other technical planning documents.
- b. the sequence of technical work to be conducted, including inputs and outputs
- c. the deliverable products from the technical work
- d. how to capture the work products of technical activities
- e. how technical risks will be identified and managed
- f. the tools, methods, and training needed to conduct the technical effort
- g. the involvement of stakeholders in each facet of the technical effort
- h. how the NASA technical team will be involved with the technical efforts of external contractors
- i. the entry and exit criteria for milestones such as technical reviews and life-cycle phases
- j. the identification, definition and control of internal and external interfaces
- k. the identification and incorporation of relevant lessons learned into the technical planning
- l. the approach for technology development and how the resulting technology will be incorporated into the project
- m. the identification and definition of the technical metrics for measuring and tracking progress to the realized product
- n. the decision criteria for make, buy or reuse decisions and incorporation criteria for Commercial Off-The-Shelf (COTS) software and hardware
- o. the plan to identify and mitigate off-nominal performance
- p. the “how-to’s” for contingency planning and replanning
- q. the plan for status assessment and reporting

- r. the approach to decision analysis. Include materials needed, required skills, and expectations in terms of accuracy.

By addressing these items and others unique to the project, the technical team will have a basis for understanding and defining the scope of the technical effort, including the deliverable products that the overall technical effort will produce, the schedule and key milestones for the project which the technical team must support, and the resources required by the technical team to perform the work.

A key element in defining the technical planning effort is understanding the amount of work associated with performing the identified activities. Once the scope of the technical effort begins to coalesce, the technical team may begin to define specific planning activities and to estimate the amount of effort and resources required to perform each task. Historically, many projects have underestimated the resources required to perform proper planning activities and have been forced into a position of continuous crisis management in order to keep up with changes in the project.

### **Define the Technical Work**

The technical effort must be thoroughly defined. When performing the technical planning, realistic values for cost, schedule and labor resources should be used. Whether extrapolated from historical data bases or from interactive planning sessions with the project and stakeholders, realistic values must be calculated and provided to the project team. Contingency should be included in any estimate and based on complexity and criticality of the effort. Contingency planning must be conducted. The following are examples of contingency planning:

- a. Additional, unplanned for, software engineering resources are typically needed during hardware and systems development and testing to aid in trouble shooting errors/anomalies. Frequently, software engineers are called upon to help troubleshoot problems and pinpoint the source of errors in hardware and systems development and testing (e.g., for writing addition test drivers to debug hardware problems). Additional software staff should be planned into the project contingencies to accommodate inevitable component and system debugging and avoid cost and schedule overruns.
- b. Hardware-in-the-Loop (HWIL) must be accounted for in the technical planning contingencies. HWIL testing is typically accomplished as a debugging exercise where the hardware and software are brought together for the first time in the costly environment of a HWIL. If upfront work is not done to understand the messages and errors arising during this test, additional time in the HWIL facility may result in significant cost and schedule impacts. Impacts may be mitigated through upfront planning, such as making appropriate debugging software available to the technical team prior to the test, etc.

### **Schedule, Organize, and Cost the Technical Effort**

Once the technical team has defined the technical work to be done, efforts can focus on producing a schedule and cost estimate for the technical portion of the project. The technical team must organize the technical tasks according to the Project WBS in a logical sequence of

events, taking into consideration the major project milestones, phasing of available funding, and timing of availability of supporting resources.

### *Scheduling*

Products described in the WBS are the result of activities that take time to complete. These activities have time precedence relationships among them which may be used to create a network schedule explicitly defining the dependencies of each activity on other activities, the availability of resources, and the receipt of receivables from outside sources.

Scheduling is an essential component of planning and managing the activities of a project. The process of creating a network schedule provides a standard method for defining and communicating what needs to be done, how long it will take, and how each element of the project WBS might affect other elements. A complete network schedule may be used to calculate how long it will take to complete a project, which activities determine that duration (i.e., critical path activities), and how much spare time (i.e., float) exists for all the other activities of the project.

Network scheduling systems help managers accurately assess the impact of both technical and resource changes on the cost and schedule of a project. Cost and technical problems often show up first as schedule problems. Understanding the project's schedule is a prerequisite for determining an accurate project budget and for tracking performance and progress. Because network schedules show how each activity affects other activities, they assist in assessing and predicting the consequences of schedule slips or accelerations of an activity on the entire project.

### *Network Schedule Data and Graphical Formats*

Network schedule data consists of:

- Activities and associated tasks
- Dependencies between activities (e.g., where an activity depends upon another activity for a receivable)
- Products or milestones that occur as a result of one or more activities
- Duration of each activity.

A network schedule contains all four of the above data items. When creating a network schedule, creating graphical formats of the above data elements may be a useful first step in planning and organizing schedule data.

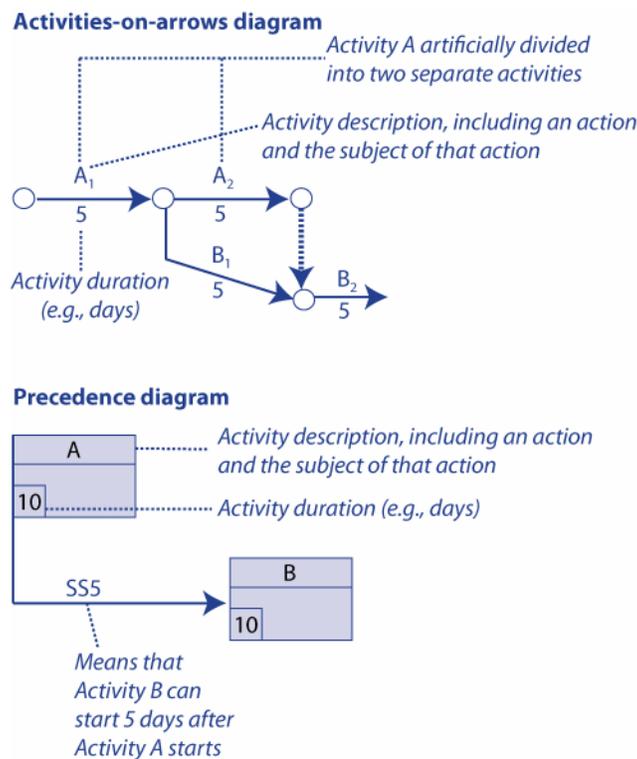
### *Work Flow Diagrams*

A work flow diagram (WFD) is a graphical display of the first three data items. Two general types of graphical formats are used as shown in Figure 6-3. One places activities-on-arrows, with products and dependencies at the beginning and end of the arrow. This is the typical format of the Program Evaluation and Review Technique (PERT) chart.

The second format, called Precedence Diagrams, uses boxes to represent activities; dependencies are then shown by arrows. The precedence diagram format allows for simple depiction of the following logical relationships:

- Activity B begins when Activity A begins (Start-Start, or SS)
- Activity B begins only after Activity A ends (Finish-Start, or FS)
- Activity B ends when Activity A ends (Finish-Finish, or FF).

Each of these three activity relationships may be modified by attaching a lag (+ or -) to the relationship, as shown in Figure 6-3. It is possible to summarize a number of low-level activities in a precedence diagram with a single activity. This is commonly referred to as *hammocking*. One takes the initial low-level activity, and attaches a summary activity to it using the first relationship described above. The summary activity is then attached to the final low-level activity using the third relationship described above. Unless one is *hammocking*, the most common relationship used in precedence diagrams is the second one mentioned above. The activity-on-arrow format can represent the identical time-precedence logic as a precedence diagram by creating artificial events and activities as needed.



**Figure 6-3 Activity-on-Arrow and Precedence Diagrams for Network Schedules**

### *Establishing a Network Schedule*

Scheduling begins with project-level schedule objectives for delivering the products described in the upper levels of the WBS. To develop network schedules that are consistent with the project's objectives, the following six steps are applied to each element at the lowest available level of the WBS.

*Step 1:* Identify activities and dependencies needed to complete each WBS element. Enough activities should be identified to show exact schedule dependencies between activities and other WBS elements. This first step is most easily accomplished by:

- Ensuring that the WBS model is extended downward to describe all significant products including documents, reports, hardware and software items.
- For each product, listing the steps required for its generation and drawing the process as a work flow diagram
- Indicating the dependencies among the products, and any integration and verification steps within the work package.

*Step 2:* Identify and negotiate external dependencies. External dependencies are any receivables from outside of, and any deliverables that go outside of, the WBS element. Negotiations should occur to ensure that there is agreement with respect to the content, format, and labeling of products that move across WBS elements so that lower level schedules can be integrated.

*Step 3:* Estimate durations of all activities. Assumptions behind these estimates (workforce, availability of facilities, etc.) should be written down for future reference.

*Step 4:* Enter the data for each WBS element into a scheduling program to obtain a network schedule and an estimate of the critical path for that element. It is not unusual at this point for some iteration of steps 1 to 4 to obtain a satisfactory schedule. Reserve is often added to critical path activities to ensure that schedule commitments can be met.

*Step 5:* Integrate schedules of lower level WBS elements so that all dependencies between elements are correctly included in a project network. It is important to include the impacts of holidays, weekends, etc. by this point. The critical path for the project is discovered at this step in the process.

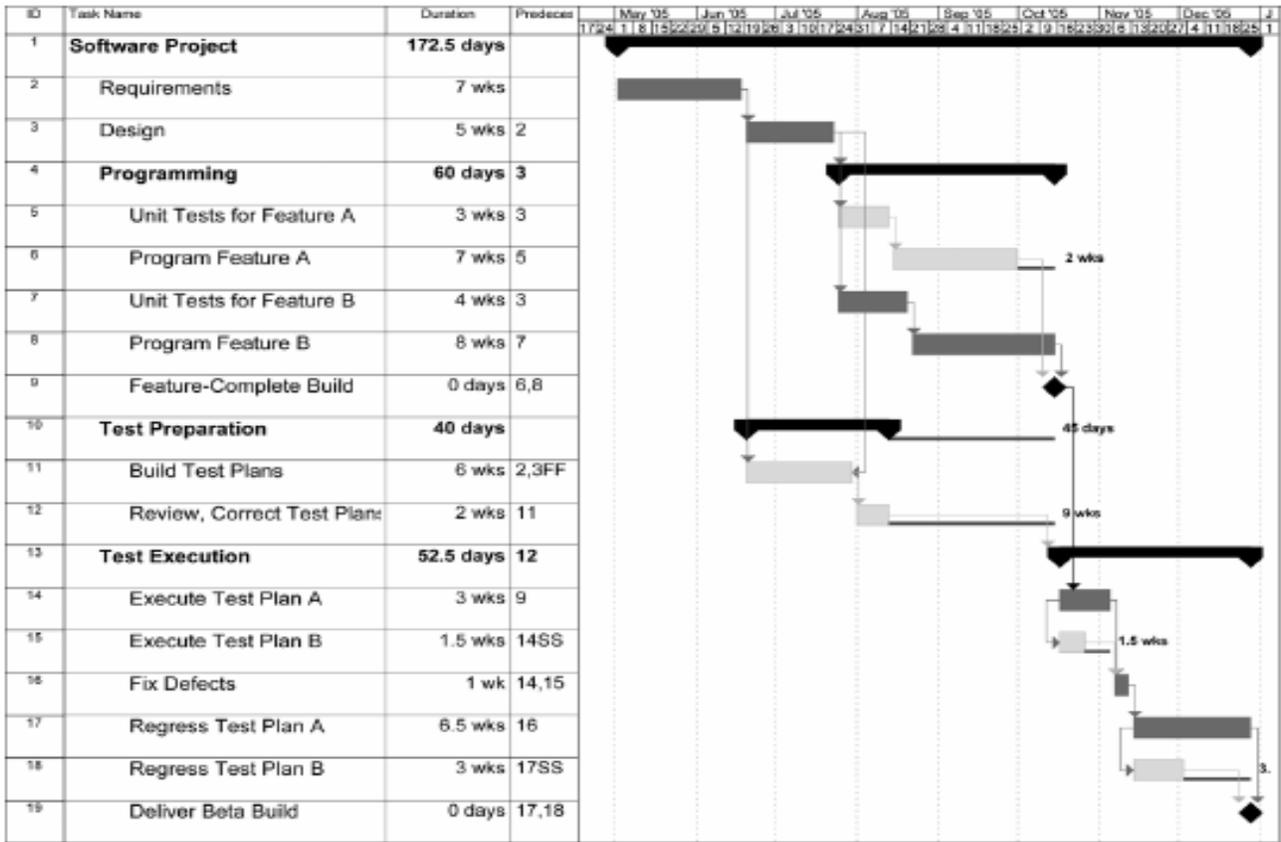
*Step 6:* Review the workforce level and funding profile over time, and make a final set of adjustments to logic and durations so that workforce levels and funding levels are within project constraints. Adjustments to the logic and the durations of activities may be needed to converge to the schedule targets established at the project level. Adjustments may include adding more activities to some WBS elements, deleting redundant activities, increasing the workforce for some activities that are on the critical path, or finding ways to do more activities in parallel, rather than in series.

Again, it is good practice to have some schedule reserve, or float, as part of a risk mitigation strategy. The product of these last steps is a feasible baseline schedule for each WBS element that is consistent with the activities of all other WBS elements. The sum of all of these schedules should be consistent with both the technical scope and the schedule goals of the project. There should be enough float in this integrated master schedule so that schedule and associated cost risk are acceptable to the project and to the project's customer. Even when this is done, time estimates for many WBS elements will have been underestimated, or work on some WBS

elements will not start as early as had been originally assumed due to late arrival of receivables. Consequently, replanning is almost always needed to meet the project's goals.

### Reporting Techniques

Summary data about a schedule is usually described in charts. A Gantt chart is a bar chart that depicts a project schedule using start and finish dates of the appropriate product elements tied to the project WBS of a project. Some Gantt charts also show the dependency (i.e., precedence and critical path) relationships between activities and also current status. A good example of a Gantt chart is shown in Figure 6-4. (See sidebar on Gantt chart features.)



**Figure 6-4 Gantt chart created in Microsoft Project**

(Chart Reference: Applied Software Project Management by Andrew Stellman and Jennifer Greene, 1st Edition, November 2005)

### Desirable Features in Gantt Charts

The Gantt chart shown in Figure 6-4 illustrates the following desirable features:

- A heading that describes the WBS element, the responsible manager, the date of the baseline used, and the date that status was reported.
- A milestone section in the main body (lines 1 and 2)
- An activity section in the main body. Activity data shown includes:
  - a. WBS elements (lines 3, 5, 8, 12, 16, and 20)
  - b. Activities (indented from WBS elements)
  - c. Current plan (shown as thick bars)
  - d. Baseline plan (same as current plan, or if different, represented by thin bars under the thick bars)
  - e. Status line at the appropriate date
  - f. Slack for each activity (dashed lines above the current plan bars)
  - g. Schedule slips from the baseline (dashed lines below the milestone on line 12)
- A note section, where the symbols in the main body can be explained.

This Gantt chart shows only 23 lines, which is a summary of the activities currently being worked for this WBS element. It is appropriate to tailor the amount of detail reported to those items most pertinent at the time of status reporting.

Another type of output format is a table that shows the float and recent changes in float of key activities. For example, a project manager may wish to know precisely how much schedule reserve has been consumed by critical path activities, and whether reserves are being consumed or are being preserved in the latest reporting period. This table provides information on the rate of change of schedule reserve.

#### *Resource Leveling*

Good scheduling systems provide capabilities to show resource requirements over time, and to make adjustments so that the schedule is feasible with respect to resource constraints over time. Resources may include workforce level, funding profiles, important facilities, etc. The objective is to move the start dates of tasks that have float to points where the resource profile is feasible. If that is not sufficient, then the assumed task durations for resource-intensive activities should be reexamined and, accordingly, the resource levels changed.

#### *Budgeting*

Budgeting and resource planning involves the establishment of a reasonable project baseline budget and the capability to analyze changes to that baseline resulting from technical and/or schedule changes. The project's WBS, baseline schedule, and budget should be viewed as mutually dependent, reflecting the technical content, time, and cost of meeting the project's goals and objectives. The budgeting process needs to take into account whether a fixed cost cap or cost profile exists. When no such cap or profile exists, a baseline budget is developed from the WBS and network schedule. This specifically involves combining the project's workforce and other resource needs with the appropriate workforce rates and other financial and programmatic factors to obtain cost element estimates. These elements of cost include:

- Direct labor costs
- Overhead costs
- Other direct costs (travel, data processing, etc.)
- Subcontract costs
- Material costs
- General and administrative costs
- Cost of money (i.e., interest payments, if applicable)
- Fee (if applicable)
- Contingency

When there is a cost cap or a fixed cost profile, there are additional logic gates that must be satisfied before completing the budgeting and planning process. A determination needs to be made whether the WBS and network schedule are feasible with respect to mandated cost caps and/or cost profiles. If not, it will be necessary to consider stretching out a project (usually at an increase in the total cost), or descoping the project's goals and objectives, requirements, design, and/or implementation approach.

If a cost cap or fixed cost profile exists, it is important to control costs after they have been baselined. An important aspect of cost control is project cost and schedule status reporting and assessment, methods for which are discussed in Section 6.7, Technical Assessment, of this handbook. Another is cost and schedule risk planning, such as developing risk avoidance and work-around strategies. At the project level, budgeting and resource planning must ensure that an adequate level of contingency funds is included to deal with unforeseen events.

The maturity of the Life-Cycle Cost Estimate (LCCE) should progress as follows:

- Pre-Phase A—Initial LCCE; 70 percent confidence level, however much uncertainty is expected
- Phase A—Preliminary commitment to LCCE
- Phase B—Approve LCCE: 70 percent confidence level at PDR commitment
- Phase C, D, and E report variances to LCCE baseline using EVM and LCCE updates.

Credibility of the cost estimate is suspect if:

- WBS cost estimates are expressed only in dollars with no other identifiable units, indicating that requirements are not sufficiently defined for processes and resources to be identified.
- Basis of Estimates do not contain sufficient detail for independent verification that work scope and estimated cost (and schedule) are reasonable.
- Actual costs vary significantly from the LCCE
- Work is performed that was not originally planned, causing cost or schedule variance

- Schedule and Cost Earned Value performance trends readily indicate unfavorable performance.

### **Prepare the SEMP and Other Technical Plans**

The SEMP is the primary, top-level technical management document for the project. It documents the integration methodologies for all of the technical and engineering activities within the scope of the Project Plan.

The systems engineer and project manager must identify additional required technical plans based on the project scope and type. If plans are not included in the SEMP, they should be referenced and coordinated in the development of the SEMP. Other plans such as system safety and the PRA also need to be planned for and coordinated with the SEMP. The list of plans below are not exhaustive. If a technical plan is a stand-alone, it should be referenced in the SEMP. Depending on the size and complexity of the project, these may be separate plans or may be included within the SEMP. Once identified, the plans can be developed, training on these plans established, and the plans implemented. Examples of technical plans in addition to the SEMP are:

- Risk Management Plan
- Requirements Management Plan
- Verification and Validation Plan
- Technical Review Plan
- Interface Management Plan
- Configuration and Data Management Plan
- Technology Development Plan
- Disposal Plan
- Safety and Mission Assurance Plan
- Software Development Plan
- Maintenance Plan
- Launch Site Operations Plan
- Environmental Management Plan

Since the SEMP describes the project's technical management approach, which is driven by the type of project, the phase in the project life cycle, and the technical development risks, it must be specifically written for each project, or project element. While the specific content of the SEMP is tailored to the project, the recommended content is discussed in Appendix 6-1, SEMP Outline.

The SEMP must be developed concurrently with the Project Plan. In developing the SEMP, the technical approach to the project, and hence the technical aspect of the project life cycle, are developed. This determines the project's length and cost. The development of the programmatic and technical management approaches requires that the key project personnel develop an understanding of the work to be performed and the relationships among the various parts of that work. Refer to section 6.1.4.1 and section 6.1.2.3.4 on Work Breakdown Structures and Network Scheduling, respectively.

The SEMP's development requires contributions from knowledgeable programmatic and technical experts from all areas of the project that can significantly influence the project's outcome. The involvement of recognized experts is needed to establish a SEMP that is credible to the project manager and to secure the full commitment of the project team.

The SEMP development process activities are driven by the critical or key events during any phase of a life cycle (from formulation through implementation, including operations, decommissioning and disposal) that set the objectives and work product outputs of the processes and how the processes are integrated.

#### *Systems Engineering Management Plan*

The SEMP is used to document the technical content of the engineering work early in the Formulation Phase for each project and is updated throughout the project life cycle. The SEMP provides the specifics of the technical effort and describes what technical processes will be used, how the processes will be applied using appropriate activities, how the project will be organized to accomplish the activities, and the cost and schedule associated with accomplishing the activities. The SEMP is driven by the technical complexity of the project, the phase in the project life cycle and the technical development risks.

The physical length of a SEMP is not what is important. This will vary from project to project. The plan needs to be adequate enough to address the specific technical needs of the project. It is a 'living' document which is updated as often as necessary to incorporate new information as it becomes available and as the project develops through implementation. The SEMP should not duplicate other project documents, however, the SEMP should reference and summarize the content of other technical plans.

The technical team, working under the overall Program Plan, develops and updates the SEMP as necessary. The technical team works with the project manager to review the content and obtain concurrence. This allows for thorough discussion and coordination of how the proposed technical activities would impact the programmatic, cost, and schedule aspects of the project.

#### *Role of the SEMP*

The SEMP is the rule book that describes to all participants how the project will be technically managed. The NASA field center responsible for the project should have a SEMP to describe how it will conduct its technical management, and each contractor should have a SEMP to describe how it will manage in accordance with both its contract and NASA's technical management practices. Each center that is involved with the project should also have a SEMP for their part of the project which would interface with the project SEMP of the responsible NASA center, but this lower tier SEMP specifically will address that center's technical effort and how it interfaces with the overall project. Since the SEMP is project- and contract-unique, it must be updated for each significant programmatic change or it will become outmoded and unused, and the project could slide into an uncontrolled state. The lead NASA field center should have its SEMP developed before attempting to prepare an initial cost estimate, since activities that incur cost, such as technical risk reduction, need to be identified and described beforehand. The contractor should have its SEMP developed during the proposal process (prior to costing and pricing) because the SEMP describes the technical content of the project, the potentially costly

risk management activities, and the verification and validation techniques to be used, all of which must be included in the preparation of project cost estimates. The SEMP from the supporting centers should be developed along with the primary project SEMP. The project SEMP is the senior technical management document for the project; all other technical control documents, such as the Interface Control Plan, Change Control Plan, Make-or-Buy Control Plan, Design Review Plan, Technical Audit Plan, depend on the SEMP and must comply with it. The SEMP should be comprehensive and describe how a fully integrated engineering effort will be managed and conducted.

### **Obtain Stakeholder Commitments to Technical Plans**

In order to obtain commitments to the technical plans by the stakeholders, the technical team should ensure that the appropriate stakeholders have a method to provide inputs and to review the project planning for implementation of stakeholder interests. During formulation, the roles of the stakeholders should be defined in the Project Plan and the SEMP. Review of these plans and the agreement from the stakeholders of the content of these plans will constitute buy-in from the stakeholders in the technical approach. Later in the project life cycle, stakeholders may be responsible for delivery of products to the project. Initial agreements regarding the responsibilities of the stakeholders are key to assuring that the project technical team obtains the appropriate deliveries from stakeholders.

The identification of stakeholders is one of the early steps in the Systems Engineering process. As the project progresses, stakeholder expectations are flowed down through the logical decomposition process and specific stakeholders are identified for all of the primary and derived requirements. A critical part of the stakeholder's involvement is in the definition of the technical requirements. As requirements and ConOps are developed, the stakeholders will be required to agree to these products. Inadequate stakeholder involvement will lead to inadequate requirements and a resultant product that does not meet the stakeholder expectations. Status on relevant stakeholder involvement should be tracked and corrective action taken if stakeholders are not participating as planned.

Throughout the project life cycle, communication with the stakeholders and commitment from the stakeholders may be accomplished through the use of agreements. Organizations may use an Internal Task Agreement (ITA), a Memorandum of Understanding (MOU), or other similar documentation to establish the relationship between the project and the stakeholder. These agreements are also used to document the customer and provider responsibilities for definition of products to be delivered. These agreements should establish the Measures of Effectiveness (MOE) or Measures of Performance (MOP) which will be used to monitor the progress of activities. Reporting requirements and schedule requirements should be established in these agreements. Preparation of these agreements will ensure that the stakeholder's roles and responsibilities support the project goals and that the project has a method to address risks and issues as they are identified.

During development of the Project Plan and the Systems Engineering Management Plan, forums are established to facilitate communication and document decisions during the life cycle of the project. These forums include meetings, working groups, decision panels and control boards. Each of these forums should establish a charter to define the scope and authority of the forum

and identify necessary voting or non-voting participants. Ad Hoc members may be identified when the expertise or input of specific stakeholders is needed when specific topics are addressed. Mandatory reviewers should be established to ensure that stakeholders have been identified to support forum.

More details on the SEMP and a general SEMP Planning Checklist can be found in section 5.1.4.2.

### **Issue Technical Work Directives**

The systems engineering manager provides technical work directives to Cost Account Managers (CAMs) within the technical team. This enables the CAMs to prepare detailed plans which are mutually consistent and collectively address all of the work to be performed. These plans include the detailed schedules and budgets for cost accounts that are needed for cost management and Earned Value Management (EVM).

Issuing technical work directives is an essential activity during Phase B of a project, when a detailed planning baseline is required. If this activity is not implemented, then the CAMs are often left with insufficient guidance for detailed planning. The schedules and budgets that are needed for EVM will then be based on assumptions and on local interpretations of project level information. If this is the case, it is highly likely that substantial variances will occur between the baseline plan and the work performed. Providing technical work directives to CAMs produces a more organized technical team. This activity may be repeated when replanning occurs.

This activity is not limited to systems engineering. This is a normal part of project planning wherever there is a need for an accurate planning baseline.

The SEM will provide technical directives to CAMs for every cost account within the SE element of the WBS. These directives may be in any format, but should clearly communicate the following information for each account:

- Technical products expected
- Documents and technical reporting requirements for each cost account
- Critical events, and specific products expected from a particular CAM in support of this event (e.g., this cost account is expected to deliver a presentation on specific topics at the PDR)
- References to applicable requirements, policies and standards
- Identification of particular tools that should be used
- Instructions on how the SEM wants to coordinate and review cost account plans before they go to Project management
- Decisions that have been made on how work is to be performed and who is to perform it

CAMs receive these technical directives, along with the project planning guidelines, and prepare cost account plans. These plans may be in any format and may have various names at different Centers, but minimally they will include:

- a. Scope of the cost account, which includes:
  - Technical products delivered
  - Other products developed that will be needed to complete deliverables (e.g., a Configuration Management System may need development in order to deliver the product of a “managed configuration”)
  - A brief description of the procedures that will be followed to complete work on these products, such as:
    - Product X will be prepared in-house, using the local procedure A, which is commonly used in Organization ABC
    - Product X will be verified/validated in the following manner ...
    - Product X will be delivered to the Project in the following manner
    - Product X delivery will include the following reports (e.g., delivery of a CM system to the project would include regular reports on the status of the configuration, etc.)
    - Product Y will be procured in accordance with procurement procedure B.
- b. A schedule would be prepared and attached to this plan which would be in a format compatible with Project guidelines for schedules. This schedule would contain each of the procedures and deliverables mentioned above, and provides additional information on the activity steps of each procedure
- c. A budget would be attached to this plan which would be in a system compatible with Project guidelines for budgets. This budget would be consistent with the resources needed to accomplish the scheduled activities
- d. Any necessary agreements and approvals.

If the Project is going to use EVM, then the scope of a cost account needs to further identify a number of “work packages”, which are units of work that can be scheduled and given cost estimates. Work packages should be based on completed products to the greatest extent possible, but may also be based on completed procedures (e.g., completion of validation). Each work package will have its own schedule and a budget. The budget for this work package becomes part of the “Budgeted Cost of Work Scheduled” (BCWS) in the EVM system. When this unit of work is completed, the Project’s Earned Value (EV) will increase by this amount. There may be future work in this cost account that is not well enough defined to be described as a set of work packages. For example, launch operations will be supported by the technical team, but the details of what will be done often are not been worked out during Phase B. In this case, this future work is called a “planning package”, which has a high level schedule and an overall budget. When this

work is understood better, the planning package will be broken up into work packages, so that the EVM system can continue to operate during launch operations.

Cost account plans should be reviewed and approved by the SEM and by the line manager of the cost account manager's home organization. Planning guidelines may identify additional review and approval requirements.

The planning process described above is not limited to systems engineering. This is the expected process for all elements of a flight project. One role that the systems engineer may have in planning is to verify that scope of work described in cost account plans across the Project is consistent with the project WBS Dictionary, and that the WBS Dictionary is consistent with the architecture of the Project.

### **Capture Technical Planning Work Products**

The work products from the Technical Planning process should be managed using either the Technical Data Management process or the Configuration Management process as required. Some of the more important products of technical planning (i.e., the WBS, the SEMP and the schedule, etc.) are kept under configuration control and captured using the CM process. The technical data management process is used to capture trade studies, cost estimates, technical analyses, reports, and other important documents not under formal configuration control. Work products such as meeting minutes and correspondence (including email) containing decisions or agreements with stakeholders should also be retained and stored in project files for later reference.

#### *6.1.1.3 Outputs*

- Technical work cost estimates, schedules, and resource needs, e.g., funds, workforce, facilities, and equipment (to project), within the project resources.
- Product and process measures needed to assess progress of the technical effort and the effectiveness of processes (to Technical Assessment Process).
- Technical Planning strategy, WBS, SEMP, and other technical plans that support implementation of the technical effort (to all processes; applicable plans to Technical Processes).
- Technical work directives, e.g., work packages or task orders with work authorization (to applicable technical teams).
- Technical planning work products needed to provide reports, records, and non-deliverable outcomes of process activities (to Technical Data Management Process).

The resulting technical planning strategy would constitute an outline, or rough draft, of the SEMP. This would serve as a starting part for the overall Technical Planning process after initial preparation is complete. When preparations for technical planning are complete, the technical team should have a cost estimate and schedule for the Technical Planning effort. The budget and schedule to support the defined Technical Planning effort can then be negotiated with the PM to

resolve any discrepancies between what is needed and what is available. The SEMP baseline needs to be completed. Planning for the update of the SEMP based on programmatic changes needs to be developed and implemented. The SEMP needs to be approved by the appropriate level of authority.

This “technical work directives” step produces: (1) planning directives to cost account managers which result in (2) a consistent set of cost account plans. Where EVM is called for, it produces (3) an EVM planning baseline, including a BCWS.

Some typical example, but not all inclusive, products of the output of the Technical Planning process are the following which could be identified and captured for all of the project personnel and stakeholders to understand:

- a. Stakeholder expectations
- b. Stakeholder requirements
- c. Stakeholder roles and responsibilities
- d. Stakeholder concept of operations and implementation
- e. Stakeholder agreement
- f. Distribution lists for review items
- g. Charters for project forums

## **6.1.2 Technical Planning Guidance**

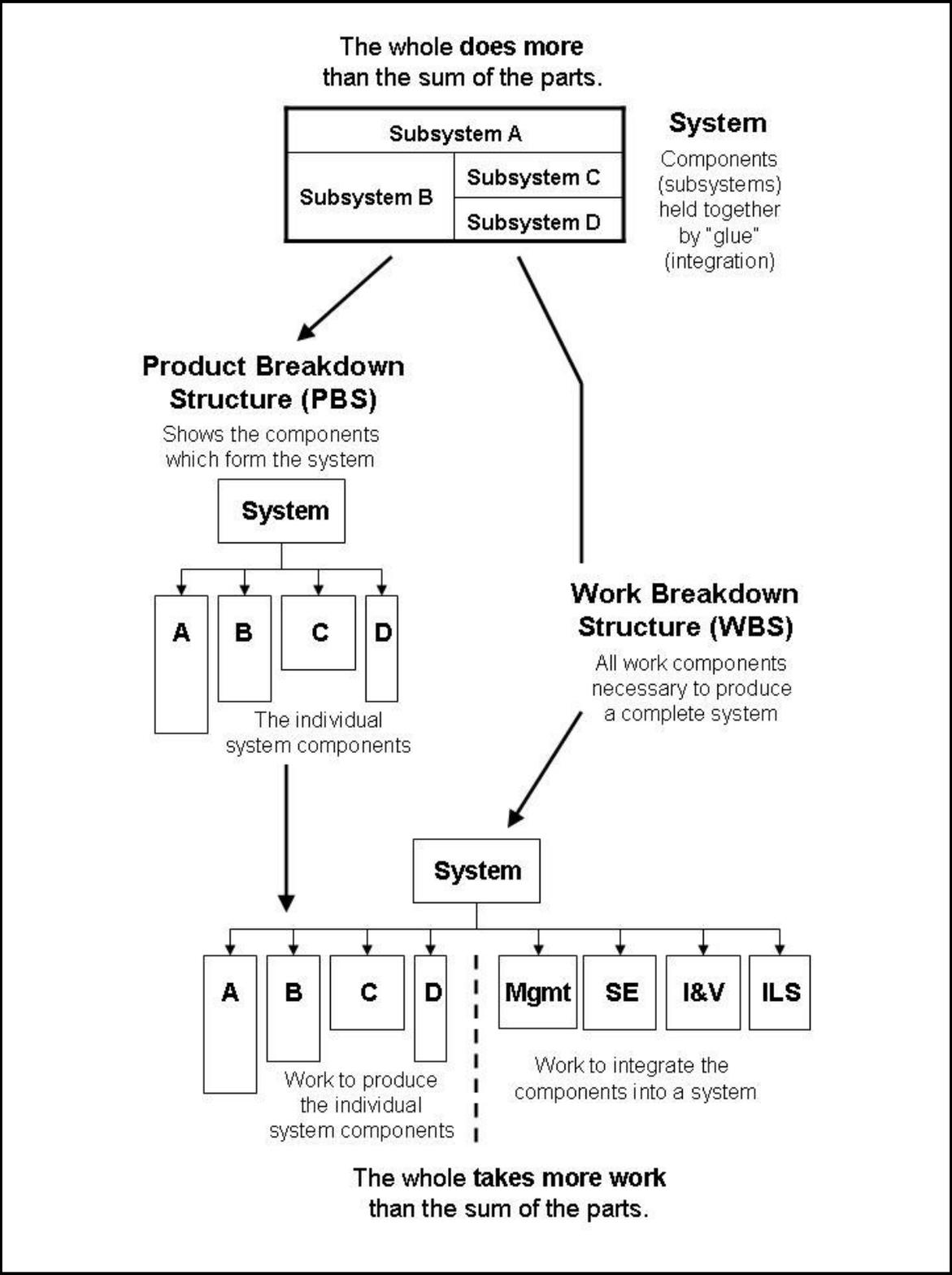
### **6.1.2.1 Work Breakdown Structure**

A Work Breakdown Structure (WBS) is a hierarchical breakdown of the work necessary to complete a project. The WBS should be a product-based, hierarchical division of deliverable items and associated services. As such, it should contain the project’s Product Breakdown Structure (PBS), with the specified prime product(s) at the top, and the systems, segments, subsystems, etc. at successive lower levels. At the lowest level are products such as hardware items, software items, and information items (documents, databases, etc.) for which there is a cognizant engineer or manager. Branch points in the hierarchy should show how the PBS elements are to be integrated. The WBS is built from the PBS by adding, at each branch point of the PBS, any necessary service elements such as management, systems engineering, integration and verification (I&V), and integrated logistics support (ILS). If several WBS elements require similar equipment or software, then a higher level WBS element might be defined to perform a block buy or a development activity (e.g., “System Support Equipment”). Figure 6-5 shows the relationship between a system, a PBS, and a WBS.

A project WBS should be carried down to the cost account level appropriate to the risks to be managed. The appropriate level of detail for a cost account is determined by management’s desire to have visibility into costs, balanced against the cost of planning and reporting. Contractors may have a Contract WBS (CWBS), which is appropriate to the contractor’s needs to control costs. A summary CWBS, consisting of the upper levels of the full CWBS, is usually included in the project WBS to report costs to the contracting organization. WBS elements should be identified by title and by a numbering system that performs the following functions:

- Identifies the level of the WBS element
- Identifies the higher level element into which the WBS element will be integrated
- Shows the cost account number of the element.

A WBS should also have a companion WBS dictionary that contains each element's title, identification number, objective, description, and any dependencies (e.g., receivables) on other WBS elements. This dictionary provides a structured project description that is valuable for orienting project members and other interested parties. It fully describes the products and/or services expected from each WBS element. This section provides some techniques for developing a WBS, and points out some mistakes to avoid.



**Figure 6-5 Relationship Between a System, a Product Breakdown Structure, and a Work Breakdown Structure**

## **Role of the WBS**

The technical team should receive planning guidelines from the Project office. The technical team should provide the Project office with any appropriate tailoring or expansion of the systems engineering WBS element, and have Project level concurrence on the WBS and WBS dictionary before issuing technical work directives.

A product-based WBS is the organizing structure for:

- Project and technical planning and scheduling
- Cost estimation and budget formulation. (In particular, costs collected in a product-based WBS can be compared to historical data. This is identified as a primary objective by DoD standards for WBSs.)
- Defining the scope of statements of work and specifications for contract efforts
- Project status reporting, including schedule, cost, workforce, technical performance, and integrated cost/schedule data (such as Earned Value and estimated cost at completion)
- Plans, such as the SEMP, and other documentation products, such as specifications and drawings.

It provides a logical outline and vocabulary that describes the entire project, and integrates information in a consistent way. If there is a schedule slip in one element of a WBS, an observer can determine which other WBS elements are most likely to be affected. Cost impacts are more accurately estimated. If there is a design change in one element of the WBS, an observer can determine which other WBS elements will most likely be affected, and these elements can be consulted for potential adverse impacts.

## **Techniques for Developing the WBS**

Developing a successful project WBS is likely to require several iterations through the project life cycle since it is not always obvious at the outset what the full extent of the work may be. Prior to developing a preliminary WBS, there should be some development of the system architecture to the point where a preliminary PBS can be created. The PBS and associated WBS can then be developed level by level from the top down. In this approach, a project-level systems engineer finalizes the PBS at the project level, and provides a draft PBS for the next lower level. The WBS is then derived by adding appropriate services such as management and systems engineering to that lower level. This process is repeated recursively until a WBS exists down to the desired cost account level. An alternative approach is to define all levels of a complete PBS in one design activity, and then develop the complete WBS. When this approach is taken, it is necessary to take great care to develop the PBS so that all products are included, and all assembly/integration and verification branches are correct. The involvement of people who will be responsible for the lower level WBS elements is recommended.

## Common Errors in Developing a WBS

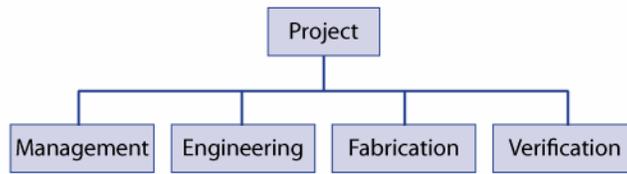
There are three common errors found in Work Breakdown Structures.

**Error 1:** The WBS describes functions, not products. This makes the project manager the only one formally responsible for products.

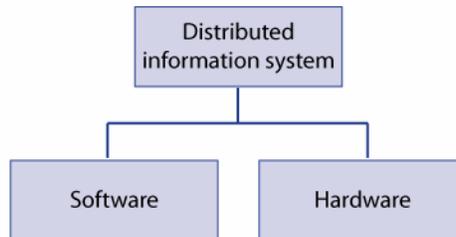
**Error 2:** The WBS has branch points that are not consistent with how the WBS elements will be integrated. For instance, in a flight operations system with a distributed architecture, there is typically software associated with hardware items that will be integrated and verified at lower levels of a WBS. It would then be inappropriate to separate hardware and software as if they were separate systems to be integrated at the system level. This would make it difficult to assign accountability for integration and to identify the costs of integrating and testing components of a system.

**Error 3:** The WBS is inconsistent with the PBS. This makes it possible that the PBS will not be fully implemented, and generally complicates the management process. Some examples of these errors are shown in Figure 6-6. Each one prevents the WBS from successfully performing its roles in project planning and organizing. These errors are avoided by using the WBS development techniques described above.

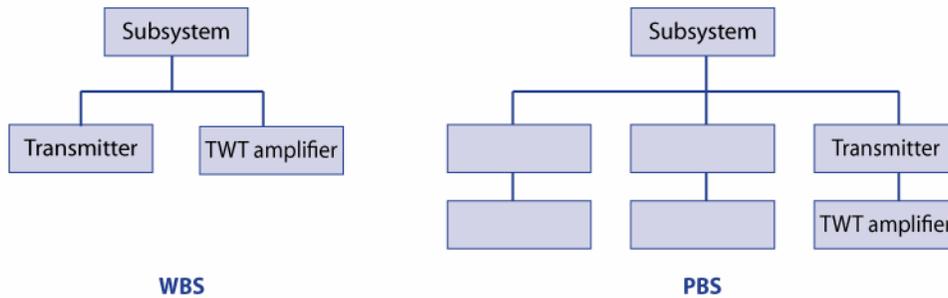
**Error 1: Functions without products**



**Error 2: Inappropriate branches**



**Error 3: Inconsistency with PBS**



**Figure 6-6 Examples of WBS Development Errors**

Common to both the Project Management and Systems Engineering disciplines is the requirement for organizing and managing a system throughout its life cycle within a systematic and structured framework, reflective of the work to be performed and the associated cost, schedule, technical and risk data to be accumulated, summarized and reported. (NPR 7120.5, Appendix A).

A key element of this framework is a hierarchical, product-oriented Work Breakdown Structure (WBS). Derived from both the physical and system architectures, the WBS provides a systematic, logical approach for defining and translating initial mission goals and technical concepts into tangible project goals, system products and life-cycle support (or enabling) functions.

When appropriately structured and used in conjunction with sound engineering principles, the WBS supplies a common framework for subdividing the total project into clearly-defined, product-oriented, work components, logically related and sequenced according to hierarchy, schedule and responsibility assignment.

The composition and level-of-detail required in the WBS hierarchy is determined by the project management and technical teams based on careful consideration of the project's size and the complexity, constraints and risk associated with the technical effort. The initial WBS will provide a structured framework for conceptualizing and defining the program/project objectives, and for translating the initial concepts into the major systems, component products, and services to be developed, produced and/or obtained. As successive levels of detail are defined, the WBS hierarchy will evolve to reflect a comprehensive, complete view of both the total project effort and each system or end-product to be realized throughout the project's life cycle.

Decomposition of the major deliverables into unique, tangible product or service elements should continue to a level representative of how each WBS element will be planned and managed. Whether assigned to in-house and contractor organizations, these lower WBS elements will be subdivided into subordinate tasks and activities and aggregated into the work packages and control accounts utilized to populate the project's cost plans, schedules and performance metrics.

At a minimum, the WBS should reflect the major system products and services to be developed and/or procured, the enabling (support) products and services, and any high cost and/or risk product-elements residing at lower levels in the hierarchy.<sup>1</sup> The baseline WBS configuration will be documented as part of the Program Plan and utilized to structure the Systems Engineering Management Plan. The cost estimates in the WBS Dictionary and maintained throughout the project's life cycle to reflect the project's current scope.

The preparation and approval of three key program/project documents, the Formulation Authorization Document (FAD), the Program Commitment Agreement (PCA), and the Program/Project Plans are significant contributors to early WBS development.<sup>2</sup>

---

<sup>1</sup> Reference IEEE Standard 1220, Section C.3 "The system products and life cycle enabling products should be jointly engineered and once the enabling products and services are identified, should be treated as systems in the overall system hierarchy.

<sup>2</sup> It is important to note that while product-oriented in nature, the standard WBS mandated for NASA Space Flight projects in NPR 7120.5D, Appendix G, approaches WBS development from a Project and not a system perspective. The WBS mandated in Appendix G is reflective of the scope of a major Agency Project; and therefore structured to include the development, operation and disposal of more than one major System-of-Interest during the Project's normal life cycle.

WBS hierarchies for NASA's Space Flight projects will include high-level system products such as Payload, Spacecraft and Ground Systems, and enabling products and services such as Project Management, Systems Engineering, and Education. These standard-product elements have been established to facilitate alignment with the Agency's accounting, acquisition and reporting systems.

Unlike the Project-view WBS approach described in Appendix G of NPR 7120.5D, creation of a technical WBS focuses on the development and realization of both the overall end-product and each, individual sub-product included as a lower-level element in the overall system structure.

NPR 7123.1, NASA Systems Engineering Processes and Requirements, mandates a standard, systematic technical approach to system or end-product development and realization. Utilizing a building-block or product-hierarchy approach, the system architecture is successively defined and decomposed into subsystems (elements performing the operational functions of the system) and associated and interrelated sub-elements (assemblies, components, parts and enabling life-cycle products).

The resulting hierarchy or family-product tree depicts the entire system architecture in a format commonly referred to as a Product Breakdown Structure (PBS). Recognized by government and industry as a "best practice," utilization of the PBS and its building-block configuration facilitates both the application of NPR 7123.1's 17

The initial contents of these documents will establish the purpose, scope, objectives and applicable agreements for the program-of-interest and will include a list of approved projects, control plans, management approaches and any commitments and constraints identified.

The technical team selects the appropriate system design processes to be employed in the “top-down” definition of each product in the system structure. Subdivision of the project and system’s architecture into smaller, more manageable components will provide logical summary points for assessing the overall project’s accomplishments and for measuring cost and schedule performance.

Once the initial mission goals and objectives have evolved into the “build-to” or final design, the WBS will be refined and updated to reflect the evolving scope and architecture of the project and the bottom-up realization of each product in the system structure.

Throughout the applicable life-cycle phases, the WBS and WBS Dictionary will be updated to reflect the project’s current scope and to ensure control of high-risk and cost/schedule performance problems.

#### *6.1.2.2 Cost Definition and Modeling*

This section deals with the role of costs in the systems analysis and engineering process, how to measure it, how to control it, and how to obtain estimates of it. The reason costs and their estimates are of great importance in systems engineering goes back to the principal objective of systems engineering: fulfilling the system’s goals in the most cost-effective manner. The cost of each alternative should be one of the most important outcome variables in trade studies performed during the systems engineering process.

One role, then, for cost estimates is in helping to choose rationally among alternatives. Another is as a control mechanism during the project life cycle. Cost measures produced for project life-cycle reviews are important in determining whether the system goals and objectives are still deemed valid and achievable, and whether constraints and boundaries are worth maintaining. These measures are also useful in determining whether system goals and objectives have properly flowed down through to the various subsystems.

As system designs and operational concepts mature, cost estimates should mature as well. At each review, cost estimates need to be presented and compared to the funds likely to be available to complete the project. The cost estimates presented at early reviews must be given special attention since they usually form the basis for the initial cost commitment for the project. The systems engineer must be able to provide realistic cost estimates to the project manager. In the

---

common technical processes at all levels of the PBS structure and the definition and realization of successively lower-level elements of the system’s hierarchy.

Definition and application of the work effort to the PBS structure yields a series of functional sub-products or “children” WBS models. The overall parent or system WBS Model is realized through the rollup of successive levels of these product-based, sub-element WBS models.

Each WBS model represents one unique unit or functional end-product in the overall system configuration; and when related by the PBS into a hierarchy of individual models represents one functional, system end-product or “parent” WBS Model

(See NPR 7120.5D, Appendix G, Space Flight Projects’ Work Breakdown Structure)

absence of such estimates, overruns are likely to occur, and the credibility of the entire system development process, both internal and external, is threatened.

### **Life-Cycle Cost and Other Cost Measures**

A number of questions need to be addressed so that costs are properly treated in systems analysis and engineering. These questions include:

- What costs should be counted?
- How should costs occurring at different times be treated?
- What about costs that cannot easily be measured in dollars?

### **What Costs Should Be Counted**

The most comprehensive measure of the cost of an alternative is its life-cycle cost. According to NPR 7120.5, a system's life-cycle cost is "the total of the direct, indirect, recurring, nonrecurring, and other related expenses incurred, or estimated to be incurred, in the design, development, verification, production, operation, maintenance, support, and disposal of a project. The LCC of a project or system can also be defined as the total cost of ownership over the project or system's life cycle from formulation through implementation. It includes all design, development, deployment, operation and maintenance, and disposal costs."

### **Costs Occurring Over Time**

The life-cycle cost combines costs that typically occur over a period of several years. In order to facilitate engineering trades and comparison of system costs, these real year costs are de-escalated to constant year values. This removes the impact of inflation from all estimates and allows ready comparison of alternative approaches. In those instances where major portfolio architectural trades are being conducted, it may be necessary to perform formal cost benefit analyses or evaluate leasing versus purchase alternatives. In those trades, engineers and cost analysts should follow the guidance provided in OMB Circular A-94 on rate of return and net present value calculation in comparing alternatives.

### **Difficult-to-Measure Costs**

In practice, estimating some costs pose special problems. These special problems, which are not unique to NASA systems, usually occur in two areas: (a) when alternatives have differences in the irreducible chances of loss of life and (b) when externalities are present. Two examples of externalities that impose costs are pollution caused by some launch systems and the creation of orbital debris. Because it is difficult to place a dollar figure on these resource uses, they are generally called *incommensurable costs*. The general treatment of these types of costs in trade studies is not to ignore them, but instead to keep track of them along with other costs. If these elements are part of the trade space, it is generally advisable to apply Circular A-94 approaches to those trades.

## Controlling Life-Cycle Costs

The project manager/systems engineer must ensure that the probabilistic life-cycle cost estimate is compatible with NASA's budget and strategic priorities. The current policy is that projects are to submit budgets sufficient to ensure a 70 percent probability of achieving the objectives within the proposed resources. Project managers and systems engineers must establish processes to estimate, assess, monitor, and control the project's life-cycle cost through every phase of the project.

*Early decisions in the systems engineering process tend to have the greatest effect on the resultant system life-cycle cost.* Typically, by the time the preferred system architecture is selected, between 50 and 70 percent of the system's life-cycle cost has been "locked in." By the time a preliminary system design is selected, this figure may be as high as 90 percent. This presents a major dilemma to the systems engineer, who must lead this selection process. Just at the time when decisions are most critical, the state of information about the alternatives is least certain. Uncertainty about costs is a fact of systems engineering and that uncertainty must be accommodated by complete and careful analysis of the project risks and provision of sufficient margins (cost, technical, and schedule) to ensure success. There are a number of estimating techniques to assist the systems engineer and project manager in providing for uncertainty and unknown requirements. Additional information on these techniques can be found in the NASA Cost Estimating Handbook ([www.ceh.nasa.gov](http://www.ceh.nasa.gov)).

This suggests that efforts to acquire better information about the life-cycle cost of each alternative early in the project life cycle (Phases A and B) potentially have very high payoffs. The systems engineer needs to identify the principal life-cycle cost drivers and the risks associated with the system design, manufacturing and operations. Consequently, it is particularly important with such a system to bring in the specialty engineering disciplines such as reliability, maintainability, supportability, and operations engineering early in the systems engineering process, as they are essential to proper life-cycle cost estimation.

One mechanism for controlling life-cycle cost is to establish a *life-cycle cost management program* as part of the project's management approach. (Life-cycle cost management has sometimes been called *design-to-life-cycle cost*.) Such a program establishes life-cycle cost as a design goal, perhaps with sub-goals for acquisition costs or operations and support costs. More specifically, the objectives of a life-cycle cost management program are to:

- Identify a common set of ground rules and assumptions for life-cycle cost estimation
- Ensure that best-practice methods, tools, and models are used for life-cycle cost analysis
- Track the estimated life -cycle cost throughout the project life cycle, *and, most important*
- Integrate life-cycle cost considerations into the design and development process via trade studies and formal change request assessments.

Trade studies and formal change request assessments provide the means to balance the effectiveness and life-cycle cost of the system. The complexity of integrating life-cycle cost considerations into the design and development process should not be underestimated, but neither should the benefits, which can be measured in terms of greater cost-effectiveness. The existence of a rich set of potential life-cycle cost trades makes this complexity even greater.

## **Cost Estimating Methods**

There are various cost estimating methodologies utilized throughout a program's life cycle. These include parametric, analogous and engineering (Grassroots).

**Parametric**—parametric cost models are used in the early stages of project development when there is limited program and technical definition. It involves collecting relevant historical data at an aggregated level of detail and relating it to the area to be estimated through the use of mathematical techniques to create Cost Estimating Relationships (CERs). Normally less detail is required for this approach than other methods.

**Analogous**—based on most new programs originated or evolved from existing programs or simply represent a new combination of existing components. It uses actual costs of similar existing or past programs and adjusts for complexity, technical, or physical differences to derive the new system estimate. This method would be used when there is insufficient actual cost data to use as a basis for a detailed approach but there is a sufficient amount of program and technical definition

**Engineering (Grassroots)**—or “bottoms up” estimates are the result of rolling up the costs estimated by each organization performing work described in the WBS. Properly done, grass-roots estimates can be quite accurate, but each time a “what if” question is raised; a new estimate needs to be made. Each change of assumptions voids at least part of the old estimate. Because the process of obtaining grassroots estimates is typically time-consuming and labor-intensive, the number of such estimates that can be prepared during trade studies is in reality severely limited.

The type of cost estimating method used will depend on the adequacy of program definition, level of detailed required, availability of data and time constraints. For example, during the early stages of program, a conceptual study considering several would dictate an estimating method requiring no actual cost data and limited program definition on the systems being estimated. A parametric model would be a sound approach at this point. Once a design is baselined and the program is more adequately defined, an analogy approach becomes appropriate. As detailed actual cost data is accumulated, a grass roots methodology is used.

More information on cost estimating methods and the development of cost estimates can be found in the NASA Cost Estimator's Handbook.

## **Integrating Cost Model Results for a Complete Life-Cycle Cost Estimate**

A number of parametric cost models are available for costing NASA systems. A listing of the models currently in use may be found in an Appendix to the NASA Cost Estimating Handbook. Unfortunately, none alone is sufficient to estimate life-cycle cost. Assembling an estimate of life-cycle cost often requires that several different models (along with the other two techniques) be used together. Whether generated by parametric models, analog or grassroots methods, the estimated cost of the hardware element must frequently be “wrapped” or have factors applied to estimate the costs associated with management, systems engineering, test, etc. of the systems being estimated. The NASA full cost factors must also be applied separately.

To integrate the costs being estimated by these different models, the systems engineer should ensure that the inputs to and assumptions of the models are consistent, that all relevant life-cycle cost components are covered, and that the phasing of costs is correct. Estimates from different sources are often expressed in different year *constant* dollars which must be combined.

Appropriate inflation factors must be applied to enable construction of a total life-cycle cost estimate in real year dollars. Guidance on the use of inflation rates for new projects and for budget submissions for on-going projects can be found in the annual NASA Strategic Guidance.

Cost models frequently produce a cost estimate for the first unit of a hardware item, but where the project requires multiple units a learning curve can be applied to the first unit cost to obtain the required multiple-unit estimate. Learning curves are based on the concept that resources required to produce each additional unit decline as the total number of units produced increases. The learning curve concept is used primarily for uninterrupted manufacturing and assembly tasks, which are highly repetitive and labor intensive. The major premise of learning curves is that each time the product quantity doubles the resources (labor hours) required to produce the product will reduce by a determined percentage of the prior quantity resource requirements. The two types of learning curve approaches are unit curve and cumulative average curve. The systems engineer can learn more about the calculation and use of learning curves in the Cost Estimator's Handbook.

Models frequently provide a cost estimate of the total acquisition effort without providing a recommended phasing of costs over the life cycle. The systems engineer can use a set of phasing algorithms based on the typical ramping-up and subsequent ramping-down of acquisition costs for that type of project, if a detailed project schedule is not available to form a basis for the phasing of the effort. One example of this type of spreading function is the beta curve, also known as the normal distribution curve, which was developed at JSC in the 1960s. It is used for spreading parametrically derived cost estimates and for R&D type contracts whereby costs build up slowly during the initial phases, and then escalate as the midpoint of the contract approaches. A beta curve is a combination of percent spent against percent time elapsed between two points in time. More about beta curves can be found in an appendix of the Cost Estimator's Handbook.

Although parametric cost models for space systems are already available, their proper use usually requires a considerable investment in learning how to appropriately utilize the models. For projects outside of the domains of these existing cost models, new cost models may be needed to support trade studies. Efforts to develop these models need to begin early in the project life cycle to ensure their timely application during the systems engineering process. Whether existing models or newly created ones are used, the SEMP and its associated life-cycle cost management plan should identify which (and how) models are to be used during each phase of the project life cycle.

### 6.1.2.3 *Lessons Learned*

No section on Technical Planning Guidance would be complete without the effective integration and incorporation of the Lessons Learned relevant to the project.

## Systems Engineering Role in Lessons Learned

Systems engineers are the main users and contributors to lessons learned systems. A lesson learned is knowledge or understanding gained by experience—either a successful test or mission or a mishap or failure. Systems engineers compile lessons learned to serve as historical documents, requirements’ rationales, and other supporting data analysis. Systems engineering practitioners collect lessons learned during program and project plans, key decision points, life-cycle phases, systems engineering processes and technical reviews. Systems engineers’ responsibilities include knowing how to utilize, manage, create and store lessons learned and knowledge management best practices.

### Utilization of LL Best Practice

Lessons learned are important to future programs, projects, and processes because they show hypotheses and conclusive insights from previous projects or processes. Practitioners determine how previous lessons from processes or tasks impact risks to current projects and implement those lessons learned that improve design and/or performance.

To pull LL in at the start of a project or task:

- Search the NASA Lessons Learned Information System (LLIS) data base using keywords of interest to the new program or project. The process for recording lessons learned is explained in *NPR 7120.6, Lessons Learned Process*. In addition, other organizations doing similar work may have lessons learned data bases publicly available. For example, the Chemical Safety Board has a good series of case study reports on mishaps.
- Supporting lessons from each engineering discipline should be reflected in the program and project plans. Even if little information was found, the search for lessons learned can be documented.
- Compile lessons by topic and/or discipline.
- Review and select knowledge gained from particular lessons learned.
- Determine how these lessons learned may represent potential risk to the current program or project.
- Incorporate knowledge gained into the project data base for risk management, cost estimate, and any other supporting data analysis.

As an example, a systems engineer working on the concept for an instrument for a spacecraft might search the lessons learned data base using the keywords “environment,” “mishap,” or “configuration management.” One of the lessons learned that search would bring up is #1514. The lesson was from Chandra. A rebaseline of the program in 1992 removed two instruments, changed Chandra’s orbit from low Earth to high elliptical, and simplified the thermal control concept from the active control required by one of the descoped instruments to passive “cold biased” surface plus heaters. This change in thermal control concept mandated silver Teflon thermal control surfaces. The event driving the lesson was a severe spacecraft charging and an

electrostatic discharge environment. The event necessitated an aggressive electrostatic discharge test and circuit protection effort that cost over one million dollars, according to the data base. The Teflon thermal control surfaces plus the high elliptical orbit created the electrostatic problem. Design solutions for one environment were inappropriate in another environment. The lesson learned was that any orbit modifications should trigger a complete new iteration of the systems engineering processes starting from requirements definition. Rebaselining a program should take into account change in the natural environment before new design decisions are made. This lesson would be valuable to keep in mind when changes occur to baselines on the program currently being worked on.

### **Management of LL Best Practice**

Capturing lessons learned is a function of good management practice and discipline. Too often lessons learned are missed because they should have been developed and managed within, across, or between life-cycle phases. There is a tendency to wait until resolution of a situation to document a lesson learned, but the unfolding of a problem at the beginning is valuable information and hard to recreate later. It is important to document a lesson learned as it unfolds, particularly as resolution may not be reached until a later phase. Since detailed lessons are often hard for the human mind to recover, waiting until a technical review or the end of a project to collect the lessons learned hinders the use of lessons and the evolution of practice. A mechanism for managing and leveraging lessons as they occur, such as monthly lessons learned briefings or some periodic sharing forums, facilitates incorporating lessons into practice and carrying lessons into the next phase.

At the end of each life-cycle phase, practitioners should use systems engineering processes and procedural tasks as control gate cues. All information passed across control gates must be managed in order to successfully enter the next phase, process or task.

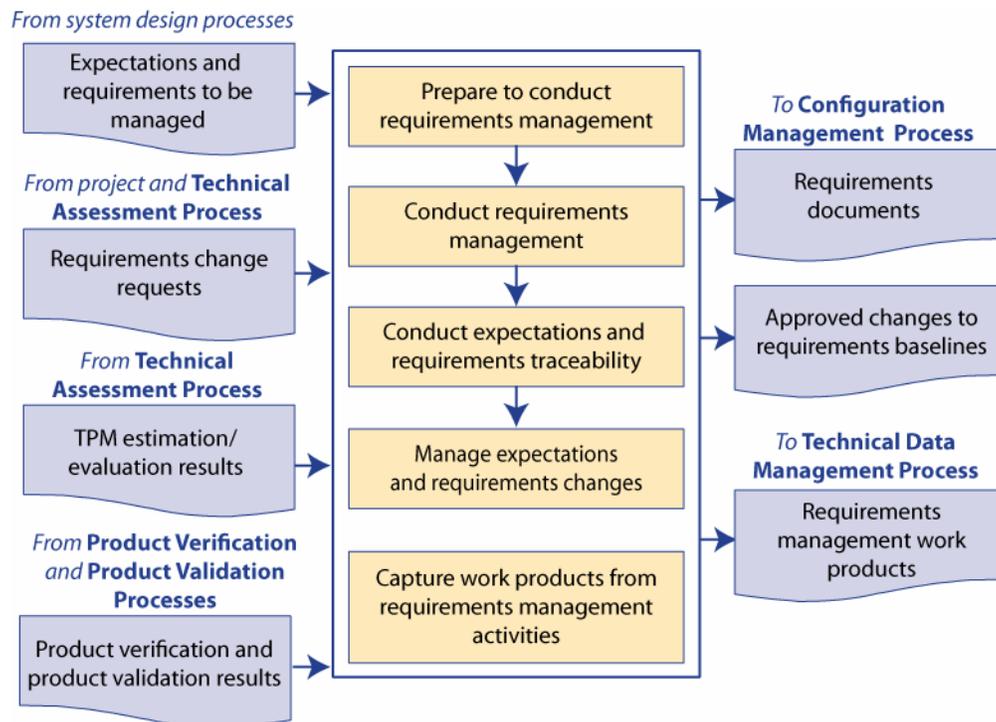
The systems engineering practitioner should make sure all lessons learned in the present phase are concise and conclusive. Conclusive lessons learned contain series of events that formulate abstracts and driving events. Irresolute lessons learned may be rolled into the next phase to await proper supporting evidence. Project managers and the project technical team are to make sure lessons learned are recorded into the Agency data base at the end of all life-cycle phases, major systems engineering processes, key decision points, and technical reviews.

## 6.2 Requirements Management

Requirements management activities apply to the management of all stakeholder expectations, customer requirements, and technical product requirements down to the lowest level product component requirements (hereafter referred to as expectations and requirements).

### 6.2.1 Process Description

Figure 6-7 provides a typical flow diagram for the Requirements Management Process and identifies typical inputs, outputs, and activities to consider in addressing requirements management.



**Figure 6-7 Requirements Management Process**

#### 6.2.1.1 Inputs

There are several fundamental inputs to the requirements management process.

- Requirements and stakeholder expectations are identified during the system design processes, primarily from the stakeholder expectation definition process and the technical requirements definition process.
- The requirements management process must be prepared to deal with requirement change requests that can be generated at any time during the project life cycle or as a result of reviews and assessments as part of the technical assessment process.

- TPM estimation/evaluation results from the technical assessment process provide an early warning of the adequacy of a design in satisfying selected critical technical parameter requirements. Variances from expected values of product performance may trigger changes to requirements.
- Product verification and product validation results from the product verification and validation processes are mapped into the requirements database with the goal of verifying and validating all requirements.

### 6.2.1.2 Process Activities

The requirements management process involves managing all changes to expectations and requirements baselines over the life of the products and maintaining bidirectional traceability between stakeholder expectations, customer requirements, technical product requirements, product component requirements, design documents, and test plans and procedures. The successful management of requirements involves several key activities:

- Establish a plan for executing requirements management
- Receive requirements from the system design processes and organize them in a hierarchical tree structure
- Establish bidirectional traceability between requirements
- Validate requirements against the stakeholder expectations, the mission objectives and constraints, the operational objectives and the mission success criteria
- Define a verification method for each requirement
- Baseline requirements
- Evaluate all change requests to the requirements baseline over the life of the project and make changes if approved by change board
- Maintain consistency between the requirements, the ConOps and the architecture/design and initiate corrective actions to eliminate inconsistencies

### Requirements Traceability

As each requirement is documented, its bidirectional traceability should be recorded. Each requirement should be traced back to a parent/source requirement or expectation in a baselined document or identify the requirement as *self-derived* and seek concurrence on it from the next higher level requirements sources. Examples of self-derived requirements are requirements that are locally adopted as good practices or are the result of design decisions made while performing Logical Decomposition and Design Solution activities.

The requirements should be evaluated, independently if possible, to ensure that the requirements trace is correct and that it fully addresses its parent requirements. If it does not, some other

requirement(s) must complete fulfillment of the parent requirement and be included in the traceability matrix. In addition, ensure that all top-level parent document requirements have been allocated to the lower level requirements. If there is no parent for a particular requirement and it is not an acceptable self-derived requirement, it should be assumed either that the traceability process is flawed and should be redone or that the requirement is “gold plating” and should be eliminated. Duplication between levels must be resolved. If a requirement is simply repeated at a lower level and it is not an externally imposed constraint, perhaps the requirement does not belong at the higher level. Requirements traceability is usually recorded in a requirements matrix (see the discussion on requirements matrix in the Requirements Definition section 4.2).

#### Definitions

**Traceability**—A discernable association among two or more logical entities such as requirements, system elements, verifications, or tasks.

**Bidirectional traceability**—An association among two or more logical entities that is discernable in either direction (i.e., to and from an entity).

### Requirements Validation

An important part of requirements management is the validation of the requirements against the stakeholder expectations, the mission objectives and constraints, the operational objectives and the mission success criteria. Validating requirements can be broken into three steps:

1. **Are Requirements Written Correctly:** identify and correct requirements “shall” statement format errors and editorial errors.
2. **Are Requirements Technically Correct:** a few trained reviewers from the technical team identify and remove as many technical errors as possible before having all the relevant stakeholders review the requirements. The reviewers should check that the requirement statements (1) have bidirectional traceability to the baselined stakeholder expectations; (2) were formed using valid assumptions; and (3) are essential to, and consistent with, designing and realizing the appropriate product solution form that will satisfy the applicable product-line life-cycle phase exit criteria.
3. **Do Requirements Satisfy Stakeholders:** all relevant stakeholder groups identify and remove defects.

Requirement validation results are often a deciding factor whether to proceed with the next process of Logical Decomposition or Design Solution. The project team should be prepared to: (1) demonstrate that the project requirements are complete and understandable; (2) demonstrate that prioritized evaluation criteria are consistent with requirements and the operations and logistics concepts; (3) confirm that requirements and evaluation criteria are consistent with customer needs; (4) demonstrate that operations and architecture concepts support mission needs, goals, objectives, assumptions, guidelines, and constraints; and (5) demonstrate that the process for managing change in requirements is established, documented in the project information repository, and communicated to stakeholders.

## **Managing Requirement Changes**

Throughout Phases A and B, changes in requirements and constraints will occur. It is imperative that all changes be thoroughly evaluated to determine the impacts on the architecture, design, interfaces, ConOps and higher and lower level requirements. Performing functional and sensitivity analyses will ensure that the requirements are realistic and evenly allocated. Rigorous requirements verification and validation ensure that the requirements can be satisfied and conform to mission objectives. All changes must be subjected to a review and approval cycle to maintain traceability and to ensure that the impacts are fully assessed for all parts of the system.

Once the requirements have been validated and reviewed in the System Requirements Review they are placed under formal configuration control. Thereafter, any changes to the requirements must be approved by the Configuration Control Board (CCB). The systems engineer, project manager, and other key engineers usually participate in the CCB approval processes to assess the impact of the change including cost, performance, programmatic and safety.

The technical team should also ensure that the approved requirements are communicated in a timely manner to all relevant people. Each project should have already established the mechanism to track and disseminate the latest project information. Further information on configuration management can be found in section 6.5.

## **Key Issues for Requirements Management**

### *Requirements Changes*

Effective management of requirements changes requires a process that assesses the impact of the proposed changes prior to approval and implementation of the change. This is normally accomplished through the use of the Configuration Management (CM) process. In order for CM to perform this function, a baseline configuration must be documented and tools used to assess impacts to the baseline. Typical tools used to analyze the change impacts are as follows:

1. **Performance Margins:** This tool is a list of key performance margins for the system and the current status of the margin. For example, the propellant performance margin will provide the necessary propellant available vs. the propellant necessary to complete the mission. Changes should be assessed for their impact to performance margins.
2. **CM Topic Evaluators List:** This list is developed by the project office to ensure that the appropriate persons are evaluating the changes and providing impacts to the change. All changes need to be routed to the appropriate individuals to ensure that the change has had all impacts identified. This list will need to be updated periodically.
3. **Risk System and Threats List:** The risk system can be used to identify risks to the project and the cost, schedule, and technical aspects of the risk. Changes to the baseline can affect the consequences and likelihood of identified risk or can introduce new risk to the project. A threats list is normally used to identify the costs associated with all the risks for the project. Project reserves are used to mitigate the appropriate risk. Analyses, of the reserves available vs. the needs identified by the threats list, assist in the prioritization for reserve use.

The process for managing requirements changes need to take into account the distribution of information related to the decisions made during the change process. The CM process needs to communicate the requirements change decisions to the affected organizations. Actions to update appropriate documentation—During the board meeting to approve a change, actions to update documentation need to be included as part of the change package. These actions should be tracked to ensure that affected documentation is updated in a timely manner.

#### *Feedback to the Requirements Baseline*

During development of the system components it will be necessary to provide feedback to the requirements. This feedback is usually generated during the product design, validation and verification process. The feedback to the project will include design implementation issues which impact the interfaces or operations of the system. In many cases, the design may introduce constraints on how the component can be operated, maintained or stored. This information needs to be communicated to the project team to evaluate the impact to the affected system operation or architecture. Each system component will optimize the component design and operation. It is the systems engineering function to evaluate the impact of this optimization at the component level to the optimization of the entire system.

#### *Requirements Creep*

Requirements creep is the term used to describe the subtle way that requirements grow imperceptibly during the course of a project. It is the tendency for the set of requirements to relentlessly increase in size during the course of development, resulting in a system that is more expensive and complex than originally intended. Often the changes are quite innocent and what appear to be changes to a system are really enhancements in disguise.

However, some of the requirements creep involves truly new requirements that did not exist, and could not have been anticipated, during the requirements definition process. These new requirements are the result of evolution, and if we are to build a relevant system, we cannot ignore them.

There are several techniques for avoiding or at least minimizing requirements creep:

1. In the early requirements definition phase flush out the conscious, unconscious, and undreamt-of requirements that might otherwise not be stated.
2. Establish a strict process for assessing requirement changes as part of the configuration management process.
3. Establish official channels for submitting change requests. This will determine who has the authority to generate requirement changes and submit them formally to the configuration control board (e.g., the contractor designated representative, project technical leads, customer/science team lead or user).
4. Measure the functionality of each requirement change request and assess its impact on the rest of the system. Compare this impact with the consequences of not approving the change. What is the risk if the change is not approved?

5. Determine if the proposed change can be accommodated within the fiscal and technical resource budgets. If it cannot be accommodated within the established resource margins then the change most likely should be denied.

### 6.2.1.3 *Outputs*

Typical outputs from the requirements management activities are:

- **Requirement Documents:** Requirement documents are submitted to the configuration management process when the requirements are baselined. The official controlled versions of these documents are generally maintained in electronic format within the requirements management tool that has been selected by the project. In this way they are linked to the requirements matrix with all of its traceable relationships.
- **Approved Changes to the Requirement Baselines:** Approved changes to the requirement baselines are issued as an output of the requirements management process after careful assessment of all the impacts of the requirements change across the entire product or system. A single change can have a far-reaching ripple effect which may result in several requirement changes in a number of documents.
- **Various Requirements Management Work Products:** Requirements management work products are any reports, records and undeliverable outcomes of the requirements management process. For example, the bidirectional traceability status would be one of the work products that would be used in the verification and validation reports.

## 6.2.2 *Requirements Management Guidance*

### 6.2.2.1 *Requirements Management Plan*

The technical team should prepare a plan for performing the requirements management activities. This plan is normally part of the Systems Engineering Management Plan but can also stand alone. The plan should:

- Identify the relevant stakeholders that will be involved in the requirements management process (e.g., those who may be affected by, or may affect, the product as well as the processes).
- Provide a schedule for performing the requirements management procedures and activities.
- Assign responsibility, authority and adequate resources for performing the requirements management activities, developing the requirements management work products, and providing the requirements management services defined in the activities. (e.g., staff, requirements management database tool, etc.)
- Define the level of configuration management/data management control for all requirements management work products.

- Identify the training for those who will be performing the requirements management activities.

#### 6.2.2.2 *Requirement Management Tools*

For small projects and products the requirements can usually be managed using a spread sheet program. However, the larger programs and projects require the use of one of the available requirement management tools. In selecting a tool it is important to define the project's procedure for specifying how the requirements will be organized in the requirements management database tool and how the tool will be used. It is possible, given modern requirements management tools, to create a requirements management database that can store and sort requirements data in multiple ways according to the particular needs of the technical team. The organization of the database is not a trivial exercise and has consequences on how the requirements data can be viewed for the life of the project. Organize the database so that it has all the views into the requirements information that the technical team is likely to need. Careful consideration should be given to how flowdown of requirements and bidirectional traceability will be represented in the database. Sophisticated requirements management database tools also have the ability to capture numerous requirement attributes in the tools' requirements matrix, including the requirements traceability and allocation links. For each requirement in the requirements matrix, the verification method(s), level, and phase are documented in the Verification Requirements Matrix housed in the requirements management database tool (e.g., the tool associates the attributes of method, level, and phase with each requirement). It is important to make sure that the requirements management database tool is compatible with the verification and validation tools chosen for the project.

## 6.3 Interface Management

The management and control of interfaces is crucial to successful programs or projects. Interface management is a process to assist in controlling product development when efforts are divided between parties (e.g., Government, contractors, geographically diverse technical teams, etc.) and/or to define and maintain compliance between the products that must interoperate.

### 6.3.1 Process Description

Figure 6-8 provides a typical flow diagram for the Interface Management Process and identifies typical inputs, outputs, and activities to consider in addressing interface management.

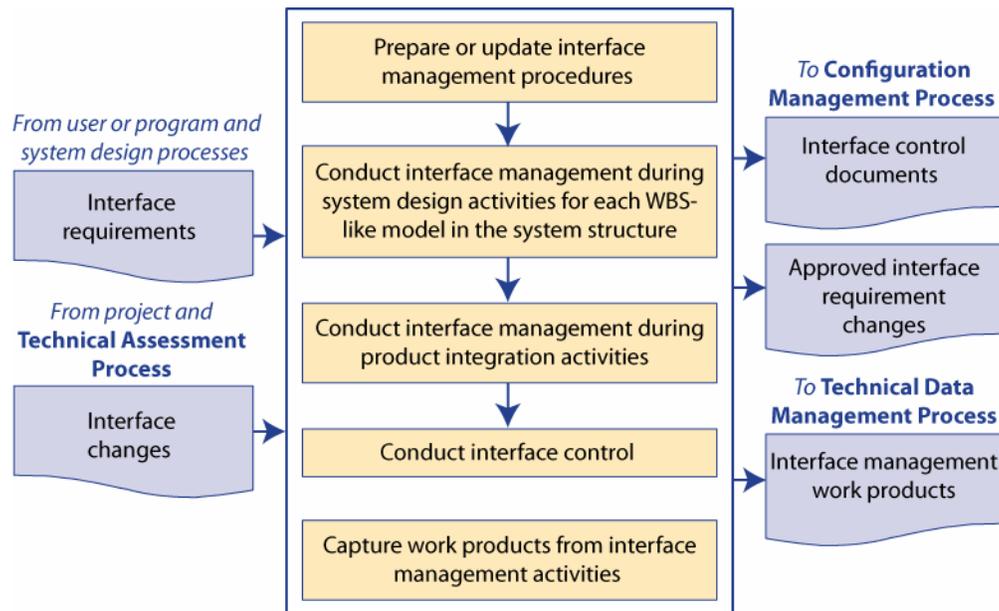


Figure 6-8 Interface Management Process

#### 6.3.1.1 Inputs

Typical inputs needed to understand and address interface management would include the following:

- **System Description:** This allows the design of the system to be explored and examined to determine where system interfaces exist. Contractor arrangements will also dictate where interfaces are needed.
- **System Boundaries:** Document physical boundaries, component and or subsystems, which are all drivers for determining where interfaces exist.
- **Organizational Structure:** Decide which organization will dictate interfaces, particularly when there is the need to jointly agree on shared interface parameters of a

system. The Program and Project Work Breakdown Structure will also provide Interface Boundaries.

- **Boards Structure:** The Systems Engineering and Management Plan (SEMP) should provide insight into organizational interface responsibilities and drive out interface locations.
- **Interface Requirements:** The internal and external functional and physical interface requirements developed as part of the requirements definition process for the product(s).
- **Interface Change Requests:** These include changes resulting from program or project agreements or changes on the part of the technical team as part of the technical assessment process

#### *6.3.1.2 Process Activities*

During project Formulation, the concept of operations of the product is analyzed to identify both external and internal interfaces. This analysis will establish the origin, destination, stimuli, and special characteristics of the interfaces that need to be documented and maintained. As the system structure and architecture emerges, interfaces will be added and existing interfaces will be changed and must be maintained. Thus, the interface management process has a close relationship to other areas, such as requirements definition and configuration management during this period. Typically, an Interface Working Group (IWG) establishes communication links between those responsible for interfacing systems, end products, enabling products and subsystems. The IWG has the responsibility to ensure accomplishment of the planning, scheduling, and execution of all interface activities. IWGs are typically a technical team with appropriate technical membership from the interfacing parties (e.g., the project, the contractor, etc.).

During product integration, interface management activities would support the review of integration and assembly procedures to ensure interfaces are properly marked and compatible with specifications and interface control documents. The interface management process has a close relationship to verification/validation. Interface control documentation and approved interface requirement changes are used as inputs to the verification and validation process, particularly where verification test constraints and interface parameters are needed to set the test objectives and test plans. Interface requirement verification is a critical aspect of the overall system verification.

#### *6.3.1.3 Outputs*

Typical outputs needed to capture interface management would include the following:

- **Interface Control Documentation:** This is the documentation that identifies and captures the interface information and the approved interface change requests (e.g., Interface Requirements Document (IRD), Interface Control Document/Drawing (ICD), Interface Definition Document (IDD), Interface Control Plan (ICP)). These outputs with

then be maintained and approved using the configuration management process and become a part of the overall technical data package for the project.

### **6.3.2 Interface Management Guidance**

#### **6.3.2.1 Interface Requirements Document**

An interface requirement defines the functional, performance, electrical, environmental, human, and physical requirements and constraints that exist at a common boundary between two or more functions, system elements, configuration items, or systems. Interface requirements include both logical and physical interfaces. They include, as necessary, physical measurements, definitions of sequences of energy or information transfer, and all other significant interactions between items. For example, communication interfaces involve the movement and transfer of data and information within the system, and between the system and its environment. Proper evaluation of communications requirements involves definition of both the structural components of communications (e.g., bandwidth, data rate, distribution, etc.) and content requirements (what data/information is being communicated, what is being moved among the system components, and the criticality of this information to system functionality). Interface requirements can be derived from the functional allocation if function inputs and outputs have been defined. For example:

- If Function F1 outputs item A to Function F2, and
- Function F1 is allocated to component C1, and
- Function F2 is allocated to component C2,
- Then there is an implicit requirement that the Interface between Components C1 and C2 to pass item A, whether item A is a liquid, a solid, or a message containing data.

The Interface Requirements Document (IRD) is a document that defines all physical, functional, and procedural interface requirements between two or more end items, elements, or components of a system and ensures project hardware and software compatibility. An IRD is composed of physical and functional requirements and constraints imposed on Hardware Configuration Items (HWCI) and/or Computer Software Configuration Items (CSCI). The purpose of the IRDs is to control the interfaces between interrelated components of the system under development, as well as between the system under development and any external systems (either existing or under development) that comprise a total architecture. Interface requirements may be contained in the system requirements document (SRD) until the point in the development process where the individual interfaces are determined. IRDs are useful when separate organizations are developing components of the system or when the system must levy requirements on other systems outside program/project control. During both Phase A and B, multiple IRDs are drafted for different levels of interfaces. By SRR, draft IRDs would be complete for system to external system interfaces (e.g., the Space Shuttle to ISS), and segment-to-segment interfaces (e.g., the Space Shuttle to the launch pad). An IRD generic outline is described in Appendix 6-3

### 6.3.2.2 *Interface Control Document or Interface Control Drawing*

An Interface Control Document or Drawing details the physical interface between two system elements, including the number and types of connectors, electrical parameters, mechanical properties, and environmental constraints. The ICD identifies the design solution to the interface requirement. ICDs are useful when separate organizations are developing design solutions to be adhered to at a particular interface.

### 6.3.2.3 *Interface Definition Document*

An Interface Definition Document is a unilateral document controlled by the end item provider and it basically provides the details of the interface for a design solution that is already established. This document is sometimes referred to as a “one-sided ICD”. The user of the IDD is provided connectors, electrical parameters, mechanical properties, environmental constraints, etc. of the existing design. The user must then design the interface of their system to be compatible with the already existing design interface.

### 6.3.2.4 *Interface Control Plan*

An Interface Control Plan (ICP) should be developed to address the process for controlling identified interfaces and the related interface documentation. Key content for the ICP is the list of interfaces by category and who owns the interface. The ICP should also address the configuration control forum and mechanisms to implement the change process (e.g., Preliminary Interface Revision Notice (PIRN)/Interface Revision Notice (IRN)) for the documents.

#### **Typical Interface Management Check List**

- Use the generic outline provided when developing the IRD. Define a ‘reserved’ placeholder if a paragraph or section is not applicable
- Ensure that there are two or more specifications that are being used to serve as the parent for the IRD specific requirements
- Ensure that ‘shall’ statements are used to define specific requirements
- Each organization must approve and sign the IRD
- A control process must be established to manage changes to the IRD
- Corresponding Interface Control Documents (ICDs) are developed based upon the requirements in the IRD
- Confirm connectivity between the Interfaces Requirements and the Verification and Validation process
- Define the SEMP content to address Interface Management
- Each major Program or Project should include a Interface Control Plan to describe the how and what of Interface Management Products

## 6.4 Technical Risk Management

In the crosscutting technical management processes, the Technical Risk Management Process is one of the control processes to support NASA’s missions. The Technical Risk Management Process is measuring or assessing risk and developing strategies to manage it, an important component of managing NASA programs under its charter to explore and expand knowledge. Critical to this process is the proactive identification and control of departures from the baseline program, project, or activity. This provides technical risk management a focus on project objectives, an analytical basis for risk management decisions and the ensuing activities, and a framework for dealing with uncertainty. Traditional risk management focuses on risks stemming from technical, cost, and schedule activities.

### 6.4.1 Process Description

Figure 6-9 provides a typical flow diagram for the Technical Risk Management Process and identifies typical inputs, outputs, and activities to consider in addressing technical risk management.

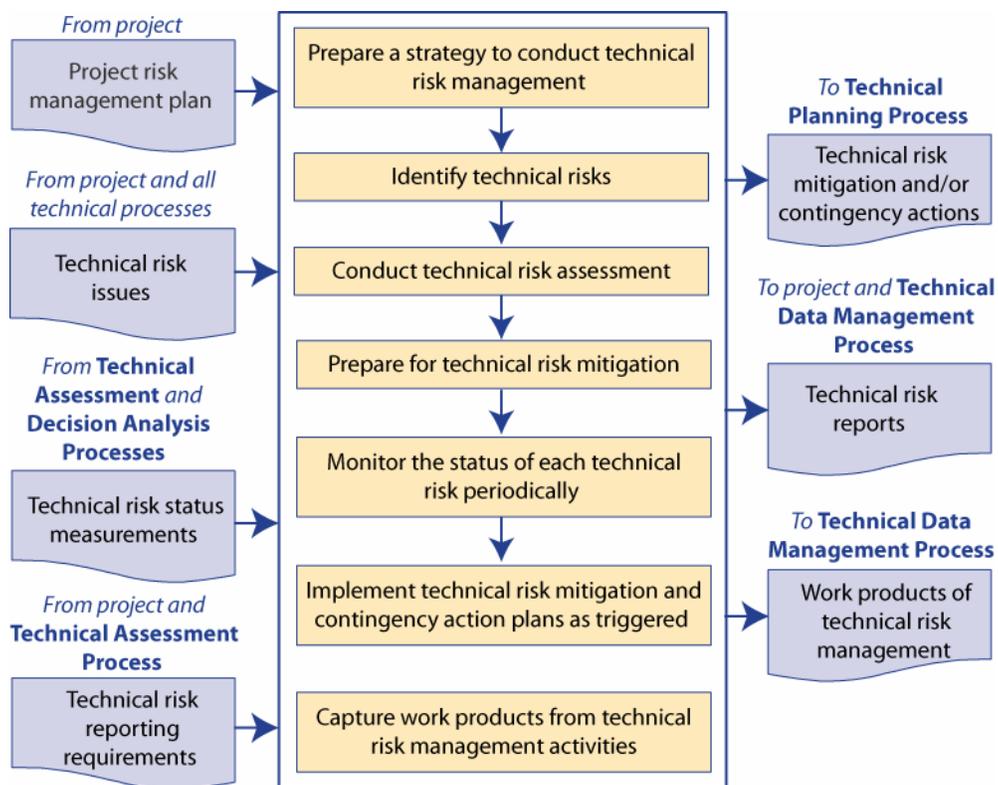


Figure 6-9 Technical Risk Management Process

#### 6.4.1.1 Inputs

Technical risk management is embedded in an analysis framework. Once the program baseline is established, the risk is identified and trade-offs are needed to determine the best course of action to manage the risk. Following are inputs to technical risk management:

- **Program Plans and Policies:** Risk management plan, risk reporting requirements, systems engineering management plan, form of technical data products, and policy input to metrics and thresholds
- **Program Technical Inputs:** technical performance measures, program alternatives to be assessed, technical issues, and current program baseline
- **Inputs Needed for Risk Analysis of Alternatives:** design information and relevant experience data
- **Other Inputs**

#### 6.4.1.2 Process Activities

Technical risk management is an iterative process to:

- Identify and assess the risks associated with implementation of technical alternatives,
- Analyze this risk and provide input to prioritizing risk to resolve uncertainty, if warranted,
- Provide input to determining the preferred alternative in light of activity requirements, constraints and priorities,
- Plan, track, and control the risk and the implementation of the selected alternative,
- Implement contingency action plans as triggered,
- Communicate, deliberate, and document work products and the risk, and
- Iterate with previous steps in light of new information throughout the life cycle.

Recognizing that the scope of technical risk management is broad, this chapter covers the high-level concepts. Strategies include transferring the risk, avoiding the risk, reducing the negative effect of the risk, and accepting some or all of the consequences of a particular risk. Once a strategy is selected, technical risk management ensures its successful implementation through planning and implementation of the risk tracking and controlling activities.

Technical risk management focuses on risk that relates to technical performance. However, management of technical risk has an impact on the non-technical risk, by affecting budget, schedule, and other stakeholder expectations. This discussion of technical risk management is

applicable to technical and non-technical risk issues, but the focus of this section is on technical risk issues.

### 6.4.1.3 Outputs

Following are key technical risk outputs from activities:

- **Program Plans and Policies:** baseline-specific plan for tracking and controlling risk
- **Program Technical Outputs:** technical risk mitigation or contingency actions; and tracking results, status findings, and emergent issues
- **Outputs from Risk Analysis of Alternatives:** identified, analyzed, prioritized, and assigned risk; and risk analysis updates

### 6.4.2 Technical Risk Management Guidance

A widely used conceptualization of risk is the scenarios, likelihoods, and consequences concept as shown below in figures 6-10 and 6-11. [Kaplan].

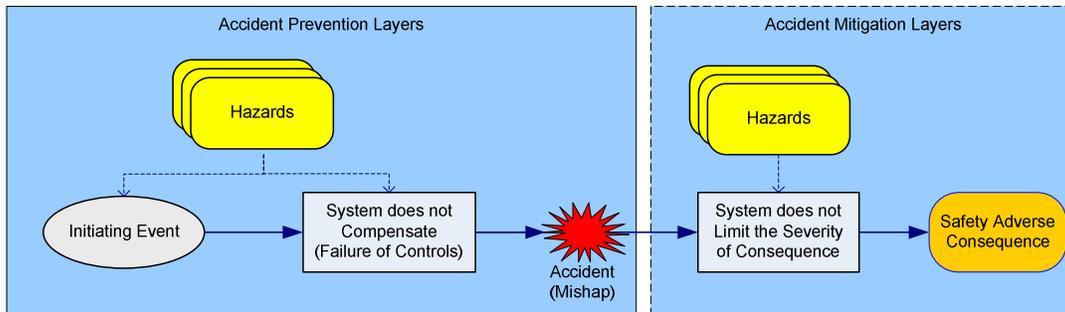


Figure 6-10 Scenario-Based Modeling of Hazards

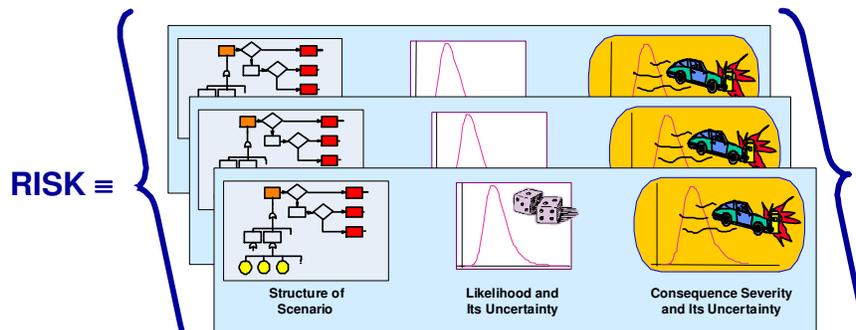


Figure 6-11 Risk as a Set of Triplets

The scenarios, along with consequences, likelihoods, and associated uncertainties, make up the complete risk triplet (risk as a Set of Triplets (Scenarios, Likelihoods, Consequences)). The triplet concept applies in principle to all risk types, and includes the information needed for

quantifying simpler measures, such as expected consequences. Estimates of expected consequences (probability or frequency multiplied by consequences) alone do not adequately inform technical decisions. Scenario-based analyses provide more of the information that risk-informed decisions need. For example, a rare but severe risk contributor may warrant a response different from that warranted by a frequent, less severe contributor, even though both have the same expected contributions to risk. In all but the simplest systems, this requires the use of detailed models to capture the important scenarios, to assess consequences, and to systematically quantify scenario likelihoods. The application of these models is to identify key and candidate models, simulation models, performance models, logic models, etc. For additional information on PRAs, refer *PRA Procedures Guide for NASA Managers and Practitioners (2002)*.

## Key Concepts in Technical Risk Management

**Risk.** Risk is a measure of the inability to achieve overall program objectives within defined cost, schedule, and technical constraints and has two components: (1) the probability of failing to achieve a particular outcome and (2) the consequences/impacts of failing to achieve that outcome. For processes, risk is a measure of the difference between actual performance of a process and the known best practice for performing that process.

**Cost Risk.** This is the risk associated with the ability of the program/project to achieve its life-cycle cost objectives and secure appropriate funding. Two risk areas bearing on cost are (1) the risk that the cost estimates and objectives are accurate and reasonable and (2) the risk that program execution will not meet the cost objectives as a result of a failure to handle cost, schedule, and performance risks.

**Schedule Risk.** These risks are those associated with the adequacy of the time estimated and allocated for the development, production, and implementation and operation of the system. Two risk areas bearing on schedule risk are (1) the risk that the schedule estimates and objectives are realistic and reasonable and (2) the risk that program execution will fall short of the schedule objectives as a result of failure to handle cost, schedule, or performance risks.

**Technical Risk.** This is the risk associated with the evolution of the design and the production of the system of interest affecting the level of performance necessary to meet the stakeholder expectations and technical requirements. The design, test, and production processes (process risk) influence the technical risk and the nature of the product as depicted in the various levels of the Product Breakdown Structure (product risk).

**Programmatic Risk.** This is the risk associated with action or inaction from outside the project, over which the project manager has no control, but which may have significant impact on the project. These impacts may manifest themselves in terms of technical, cost, and/or schedule. This includes such activities as: ITAR requirements, import/export control, partner agreements with other domestic or foreign organizations (MOU or LOA), congressional direction or earmarks, OMB direction, industrial contractor restructuring, external organizational changes, etc.

**Hazard vs. Risk.** Hazard is distinguished from risk. A hazard represents a potential for harm, while risk includes consideration of not only the potential for harm, but also the scenarios leading to adverse outcomes and the likelihood of these outcomes. In the context of safety, "risk" considers the likelihood of undesired consequences occurring.

**Probabilistic Risk Assessment (PRA).** PRA is a scenario-based risk assessment technique that quantifies the likelihoods of various possible undesired scenarios and their consequences, as well as the uncertainties in the likelihoods and consequences. Traditionally, design organizations have relied on surrogate criteria such as system redundancy or system-level reliability measures, partly because the difficulties of directly quantifying actual safety impacts, as opposed to simpler surrogates, seemed insurmountable. Depending on the detailed formulation of the objectives hierarchy, PRA can be applied to quantify Technical Performance Measures (TPMs) that are very closely related to fundamental objectives (e.g., probability of loss of crew). PRA focuses on development of a comprehensive scenario set, which has immediate include reliability models, system safety.

### 6.4.2.1 Role of Continuous Risk Management in Technical Risk Management

CRM is a widely used technique within NASA. CRM is an iterative and adaptive process that promotes the successful execution of program risk. As information is gained during program implementation, more is learned about the program's risk. Improved designs or processes may be identified, calling for updates to the risk management plan and to the baseline risk. This adaptive feature of CRM is suggested in Figure 6-12. CRM is a process in which each step of the paradigm builds on the previous step, initiated at the beginning of the program life cycle and continuing throughout the program. As information is generated or developed and a design or plan evolves, the additional information is fed back into the risk evaluation process. Further

changes may result, resulting in further evaluation, a process illustrated in Figure 6-12. CRM is initiated at the beginning of the program and is used continuously so as to avoid losing things at any time they get off track.



**Figure 6-12 Continuous Risk Management**

A brief overview of CRM is provided below for reference

**Identify**—Identify program risk by identifying scenarios having adverse consequences (deviations from program intent). CRM addresses risk related to safety, technical performance, cost, schedule, and other risk that is specific to the program.

**Analyze**—Estimate the likelihood and consequence components of the risk through Analysis, including uncertainty in the likelihoods and consequences, and the timeframes in which risk mitigation actions must be taken.

**Plan**—Plan the Track and Control actions. Decide what will be tracked, decision thresholds for corrective action, and proposed risk control actions.

**Track**—Track program observables relating to TPMs (performance data, schedule variances, etc.), measuring how close the program performance is compared to its plan.

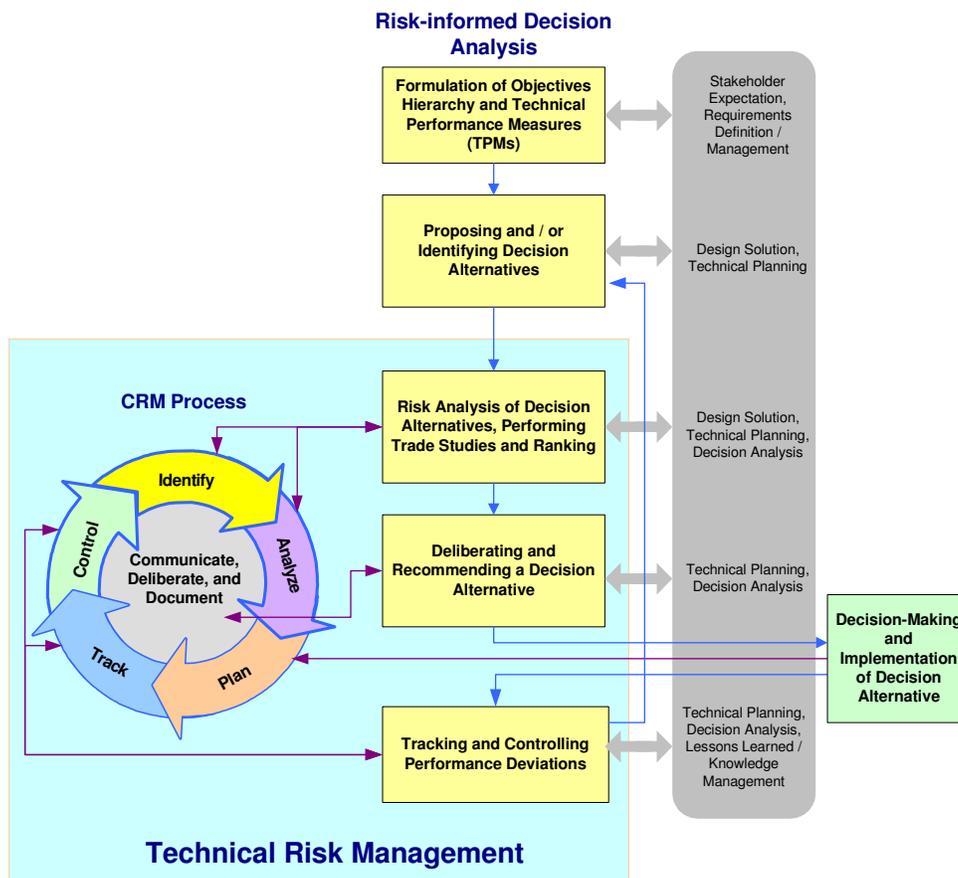
**Control**—Given an emergent risk issue, execute the appropriate control action, and verify its effectiveness.

**Communicate, Deliberate, and Document**—This is an element of each of the previous steps. Focus on understanding and communicating all risk information throughout each program phase. Document the risk, risk control plans, and closure/acceptance rationale. Deliberate on decisions throughout the CRM process.

### 6.4.2.2 The Interface Between CRM and Risk-Informed Decision Analysis

Figure 6-13 shows the interface between CRM and risk-informed decision analysis (see section 6.8.2.3). The top of Figure 6-13 shows the following inputs from risk-informed decision Analysis relating to the Objectives Hierarchy, TPMs, and Decision Alternatives:

- A selection of program-specific TPMs that both scope and prioritize much of technical risk management (top box), and
- Program alternatives that were formulated in light of these TPMs (second box).



**Figure 6-13 Risk-Informed Decision Analysis**

The risk analysis of decision alternatives (third box) not only guides selection of a preferred alternative, it also carries out the “Identify” and “Analyze” steps of CRM. Selection of a preferred alternative is based in part on an understanding of the risks associated with that alternative. Alternative selection is followed immediately by a planning activity in which key implementation aspects are addressed, namely, risk tracking and control, including risk mitigation if necessary. Also shown conceptually on Figure 6-13 is the interface between risk management and other technical and programmatic processes.

## Risk Analysis, Performing Trade Studies and Ranking (CRM Steps: “Identify,” “Analyze”)

The goal of this step is to carry out the kinds and amounts of analysis needed to characterize the risk for two purposes: ranking risk alternatives, and performing the “Identify” and “Analyze” steps of CRM.

To support ranking, trade studies may be performed. TPMs that can affect the decision outcome are quantified including uncertainty as appropriate.

To support the “Identify” and “Analyze” steps of CRM, the risk associated with the preferred alternative is analyzed in detail. Refer to Figure 6-12. Risk analysis can take many forms, ranging from qualitative risk identification (essentially scenarios and consequences, without performing detailed quantification of likelihood using techniques such as FMEA and fault trees), to highly quantitative methods such as PRA. The analysis stops when the technical case is made; if simpler, more qualitative methods suffice, then more detailed methods need not be applied. The process is then identified, planned for and continuously checked. Selection and application of appropriate methods is discussed as follows.

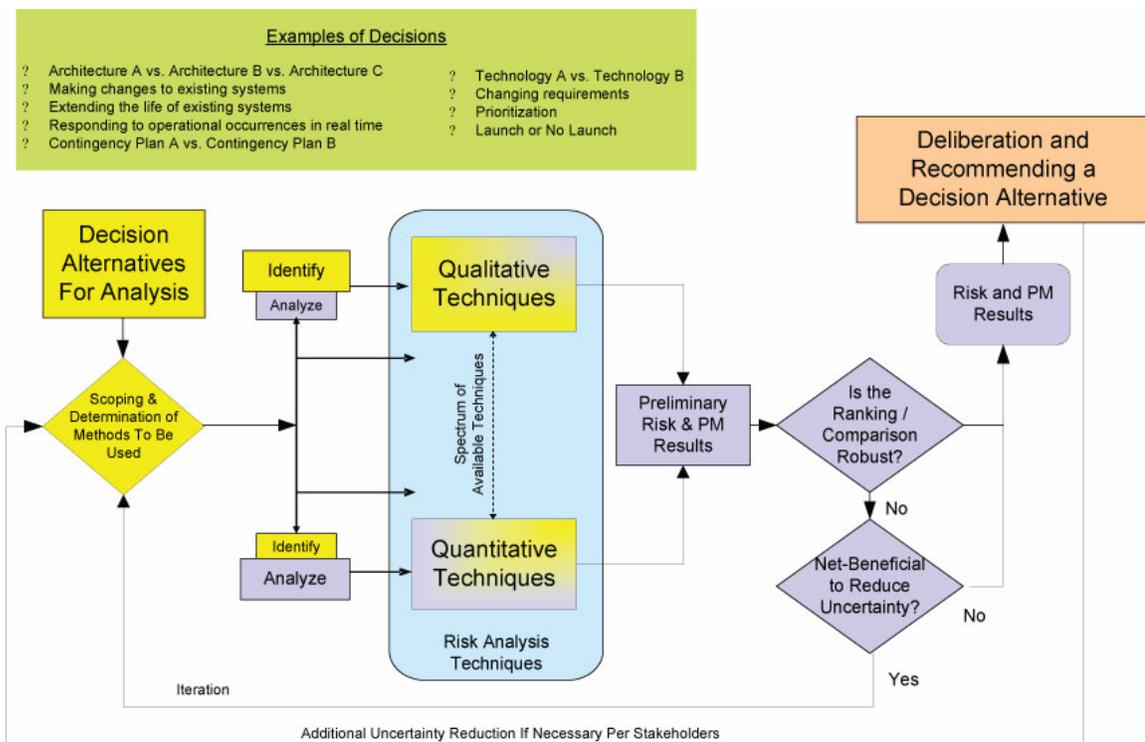


Figure 6-14 Risk Analysis of Decision Alternatives

### 6.4.2.3 *Selection and Application of Appropriate Risk Methods*

The nature and context of the problem, and the specific TPMs, determine the methods to be used. In some projects, qualitative methods are adequate for making decisions; in others, these methods are not precise enough to appropriately characterize the magnitude of the problem, or to allocate scarce risk reduction resources. The technical team needs to decide whether risk identification and judgment-based characterization are adequate, or whether the improved quantification of TPMs through more detailed risk analysis is justified. In making that determination, the technical team must balance the cost of risk analysis against the value of the additional information to be gained. The concept of “value of information” is central to making the determination of what analysis is appropriate and to what extent uncertainty needs to be quantified.

A review of the lessons learned files, data, and reports from previous similar projects can produce insights and information for hazard identification on a new project. This includes studies from similar systems and historical documents, such as mishap files, and near-miss reports. The key to applying this technique is in recognizing what aspects of the old projects and the current project are analogous, and what data from the old projects are relevant to the current project. In some cases the use of quantitative methods can compensate for limited availability of information because, these techniques pull the most value from the information that is available.

#### **Types of Risk**

As part of selecting appropriate risk analysis methods, it is useful to categorize types of risk. Broadly, risk can be related to cost, schedule, and technical performance. Many other categories exist, such as safety, organizational, management, acquisition, supportability, political, and programmatic risk, but these can be thought of as supersets of the broad categories. For example, programmatic risk refers to risk that affects cost and/or schedule, but not technical.

In the early stages of a risk analysis, it is typically necessary to screen contributors to risk in order to determine the drivers, which warrant more careful analysis. For this purpose, conservative bounding approaches may be appropriate. Overestimates of risk significance will be corrected when more detailed analysis is performed. However, it can be misleading to allow bounding estimates to drive risk ranking. For this reason, analysis will typically iterate on a problem, beginning with screening estimates, using these to prioritize subsequent analysis, and moving on to a more defensible risk profile based on careful analysis of significant contributors. This is part of the iteration loop shown in Figure 6-12.

### Example Sources of Risk

In the “Identify” activity, checklists such as this can serve as a reminder to analysts regarding areas in which risks have been identified previously.

Unrealistic schedule estimates or allocation  
Unrealistic cost estimates or budget allocation  
Inadequate staffing or skills  
Uncertain or inadequate contractor capability  
Uncertain or inadequate vendor capability  
Insufficient production capacity  
Operational hazards  
Issues, hazards, and vulnerabilities that could adversely affect the program’s technical effort  
Unprecedented efforts without estimates  
Poorly defined requirements  
No bi-directional traceability of requirements  
Infeasible design  
Inadequate configuration management  
Unavailable technology  
Inadequate test planning  
Inadequate quality assurance  
Requirements prescribing non-developmental products too low in the product tree  
Lack of concurrent development of enabling products for deployment, training, production, operations, support and/or disposal

### Qualitative Methods

Commonly used qualitative methods accomplish the following:

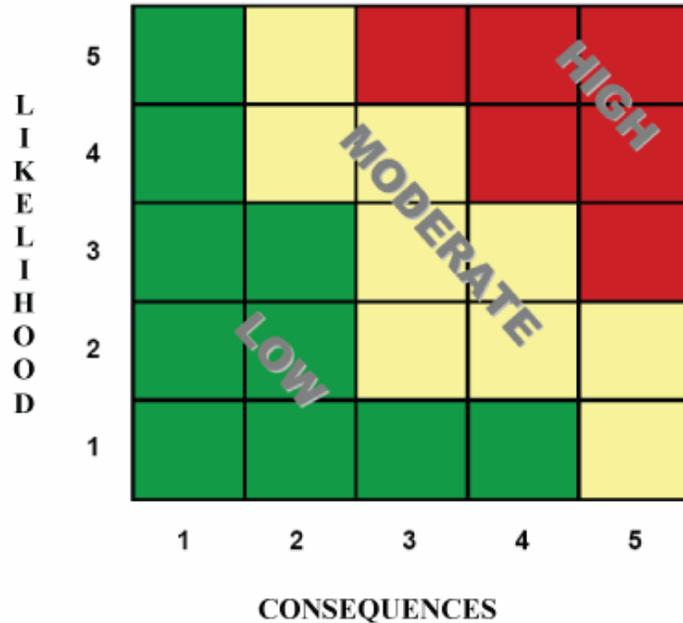
- Help to identify scenarios that are potential risk contributors;
- Provide some input to more quantitative methods;
- Support judgment-based quantification of TPMs.

These qualitative methods are discussed briefly below.

#### *Risk Matrices*

“NxM” (most commonly 5x5), Figure 6-15, risk matrices provide assistance in managing and communicating risk and combines qualitative and semi-quantitative measures of likelihood with similar measures of consequences. The risk matrix is not an assessment tool, but can facilitate risk discussions. Specifically, risk matrices help to:

- Prioritize risk—distinguish between the different levels of risk (high, moderate, and low),
- Track the status and effects of risk-handling efforts, and
- Communicate risk status information.



**Figure 6-15 Risk Matrix**

When ranking risk (high (red), medium (yellow), low (green)) it is important to use a common methodology. Different organizations and sometimes projects establish their own format. This can cause confusion and miscommunication so before using a ranking system the definitions should be clearly established and communicated via a legend or some other method. For the purposes of this handbook, a widely used (by NASA, Industry and other government organizations) definition is provided.

**Low Risk:** Has little or no potential for increase in cost, disruption of schedule, or degradation of performance. Actions within the scope of the planned program and normal management attention should result in controlling acceptable risk.

**Moderate Risk:** May cause some increase in cost, disruption of schedule, or degradation of performance. Special action and management attention may be required to handle risk.

**High Risk:** Likely to cause significant increase in cost, disruption of schedule, or degradation of performance. Significant additional action and high priority management attention will be required to handle risk.

*FMECAs, FMEAs, and Fault Trees*

Failure Modes, Effects, and Criticality Analysis (FMECA), Failure Modes and Effects Analysis (FMEA), and fault trees are specialized techniques for risk identification and characterization. These techniques focus on the hardware components as well as processes that make up the system. According to MIL-STD-1629, FMECA is an ongoing procedure by which each potential

failure in a system is analyzed to determine the results or effects thereof on the system, and to classify each potential failure mode according to its consequence severity. (See Figure 6-16.)

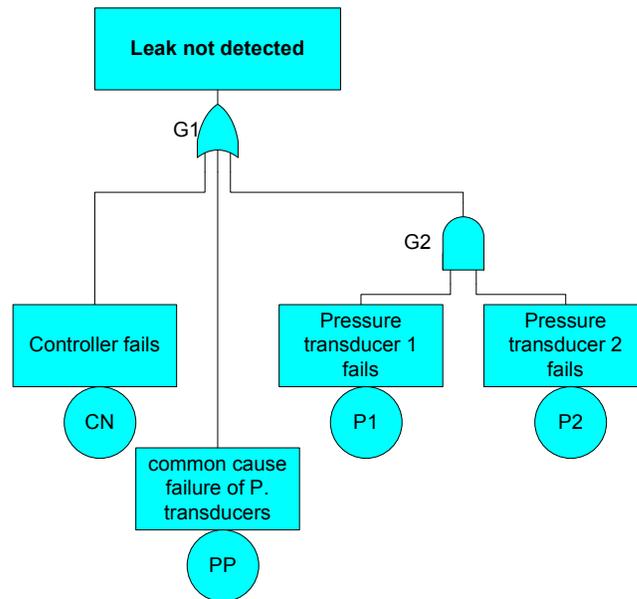


Figure 6-16 An Example of a Fault Tree

#### Limitations of Risk Matrices

- Interaction between risks is not considered. Each risk is mapped onto the matrix individually. (These risks can be related to each item using FMECA or a fault tree).
- Inability to deal with aggregate risks (i.e., total risk).
- Inability to represent uncertainties. A risk is assumed to exist within one likelihood range and consequence range, both of which are assumed to be known.
- Fixed tradeoff between likelihood and consequence. Using the standardized 5x5 matrix, the significance of different levels of likelihood and consequence are fixed and unresponsive to the context of the program.

#### Quantitative Methods

If the stakes are high, or if a risk-informed decision requires careful consideration of uncertainty and/or careful quantification of risk metrics, then more detailed and correspondingly expensive methods are implied. For example, NPR 8715.3 requires that a PRA be performed to quantify the probability of Loss of Crew. PRA adopts a systematic approach to scenario development, in whatever detail is necessary for quantification. Understanding of the structure of the scenarios supports identification of specific risk issues and formulation of corresponding risk mitigation strategies. PRA applies state-of-knowledge methods for quantifying probabilities and consequences (including uncertainty). Quantification of the probabilities may call for significant use of operating experience data from NASA or other program areas, human error modeling, and, depending on the technical scope of the analysis, simulation of phenomenological behavior may also be needed. The resources needed for PRA are justified by the importance of the

consequences modeled. In general, the stopping rule for risk analysis is that the cost in time and resources of further analysis is no longer justified by the expected benefits.

The NASA risk NPR 8715.3, is the policy document to determine the scope and the level of rigor of the risk assessments to be assigned the project and to a Project Priority Ranking based on its Consequence Category and other Criteria/Specifics.

### *Quantification*

TPMs are quantified for each alternative and used to quantify an overall “Performance Index” (PI) or an overall measure of effectiveness for each alternative. These results are then used for ranking alternatives

Bounding approaches are often used for initial screening of possible risk contributors. However, realistic assessments must ultimately be performed on the risk drivers. Bounding approaches are inappropriate for ranking alternatives because they bias each TPM in which they are applied, and are very difficult to do at a quantitatively consistent level from one analysis to the next.

Because different tools employ different simplifications and approximations, it is difficult to compare analysis results in a consistent manner if they are based on different tools or done by different analysts. These sources of inconsistency need to be considered when the work is planned and when the results are applied. Vetting risk and TPM results with these factors in mind is one benefit of deliberation (discussed below).

### *Consideration of Uncertainty Reduction Measures*

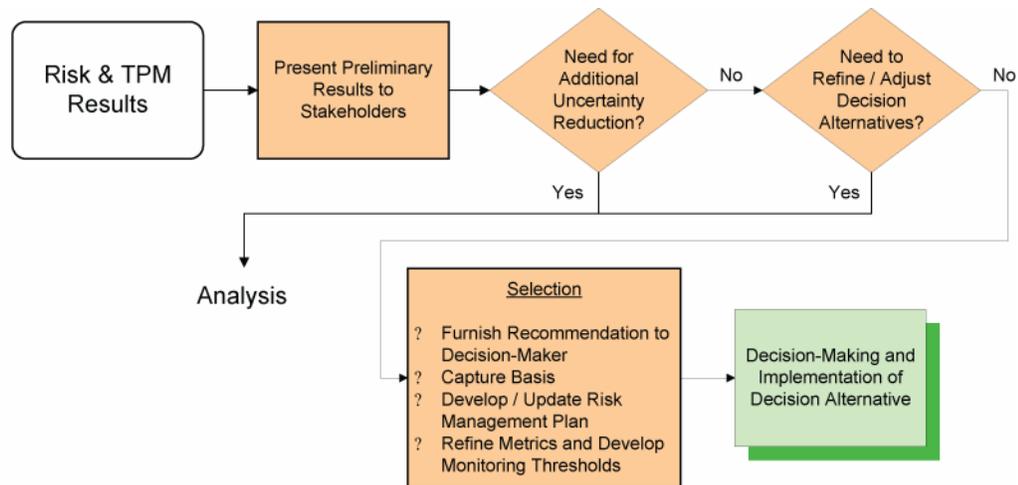
In some cases, the preliminary ranking of alternatives will not be robust. A “robust” ranking is one that is not sensitive to small changes in model parameters or assumptions. As an example, suppose that differences in TPMs of different decision alternatives are sufficiently small that variations of key parameters within the stated uncertainty bounds could change the ranking. This could arise in a range of decision situations, including architecture decisions and risk management decisions for a given architecture. In the latter case, the alternatives result in different risk mitigation approaches.

In such cases, it may be worthwhile to invest in work to reduce uncertainties. Quantification of the “value of information” can help the decision-maker determine whether uncertainty reduction is an efficient use of resources.

## **Deliberation and Recommendation of Decision Alternative (CRM Steps: “Plan,” “Deliberate, Communicate, and Document”)**

### *Deliberation*

Deliberation is recommended in order to make use of collective wisdom to promote selection of an alternative for actual implementation, or perhaps, in the case of complex and high-stakes decisions, to recommend a final round of trade studies or uncertainty reduction efforts, as suggested by the “Analysis” arrow in Figure 6-17.



**Figure 6-17 Deliberation**

*Capturing the Preferred Alternative and the Basis for its Selection*

Depending on the level at which this methodology is being exercised (project level, subtask level, etc.), the technical team chooses an alternative, basing the choice on deliberation to the extent appropriate. The decision itself is made by appropriate authority inside of the systems engineering processes. The purpose of calling out this step is to emphasize that key information about the alternative needs to be captured and that this key information includes the perceived potential program vulnerabilities that are input to the “planning” activity within CRM. By definition, the selection of the alternative is based at least in part on the prospective achievement of certain values of the TPMs. For purposes of monitoring and implementation, these TPM values help to define “success,” and are key inputs to the determination of monitoring thresholds.

*Planning Technical Risk Management of the Selected Alternative*

At this point, a single alternative has been chosen. During analysis, the risk of each alternative will have been evaluated for purposes of TPM quantification, but detailed risk management plans will not have been drawn up. At this stage, detailed planning for technical risk management of the selected alternative takes place, and a formal risk management plan is drafted. In the Plan phase,

- Provisional decisions are made on risk control actions (mitigate, research, watch, or accept)
- Observables are determined for use in measurement of program performance,
- Thresholds are determined for the observables such that nonexceedance of the thresholds indicates satisfactory program performance,
- Protocols are determined guiding how often observables are to be measured, what to do when a threshold is exceeded, how often to update the analyses, decision authority, etc.

- Responsibility for the risk tracking is assigned.

General categories of risk control actions are summarized here [NPR 8000.4]. Each identified and analyzed risk can be managed in one of four ways:

- Mitigate the risk
- Research the risk
- Watch the risk
- Accept the risk.

Steps should be taken to mitigate the risk if it is well understood and the benefits realized by the mitigation are commensurate with the cost. Benefits are determined using the TPMs from the program's objectives hierarchy. The consequences of mitigation alternatives need to be analyzed to ensure that they do not introduce unwarranted new contributions to risk.

If mitigation is not justified, other activities are considered. Suppose that there is substantial uncertainty regarding the risk. For example, there may be uncertainty in the probability of a scenario or in the consequences. This creates uncertainty in the benefits of mitigation, such that a mitigation decision cannot be made with confidence. In this case, research may be warranted to reduce uncertainty and more clearly indicate an appropriate choice for the control method. Research is only an interim measure, eventually leading either to risk mitigation or to acceptance.

If neither risk mitigation nor research is justified and the consequence associated with the risk is small, then it may need to be accepted. The risk acceptance process considers the likelihood and the severity of consequences. This is delineated in NPR 8000.4. NPR 8000.4 delineates the program level with authority to accept risk and requires accepted risk to be reviewed periodically (minimum of every six months) to ensure that conditions and assumptions have not changed requiring the risk acceptance to be reevaluated. These reviews should take the form of quantitative and qualitative analyses, as appropriate.

The remaining cases are those in which neither risk mitigation nor research are justified, and the consequence associated with the risk is large. If there is large uncertainty in the risk, then it may need to be watched. This allows the uncertainty to reduce naturally as the program progresses and knowledge accumulates, without a research program targeting that risk. As with research, watching is an interim measure, eventually leading either to risk mitigation or to acceptance, along guidelines previously cited.

### *Effective Planning*

The balance of this section is aimed primarily at assuring that the implementation plan for risk monitoring is net-beneficial.

A good plan has a high probability of detecting significant deviations from program intent in a timely fashion, without over-burdening the program. In order to accomplish this, a portfolio of observables and thresholds needs to be identified. Selective plan implementation then checks for deviations of actual TPM values from planned TPM values, and does so in a way that adds net value by not over-burdening the project with reporting requirements. Elements of the plan

include financial and progress reporting requirements, which are somewhat pre-determined, and additional program-specific observables, audits, and program reviews.

The selection of observables and thresholds should have the following properties:

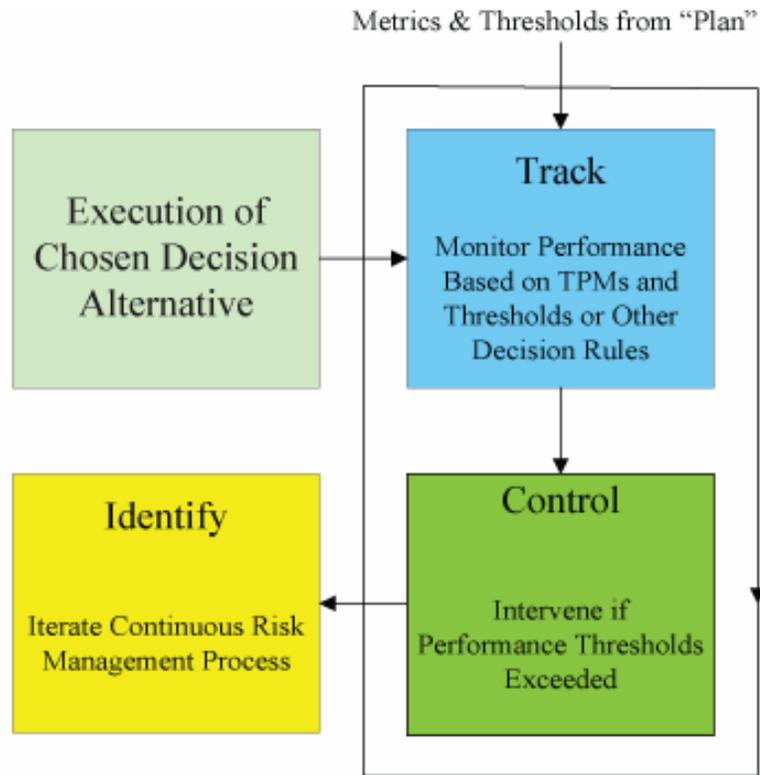
- Measurable parameters (direct measurement of the parameter or of related parameters that can be used to calculate the parameter) exist to monitor system performance against clearly defined, objective thresholds,
- The monitoring program is such that exceedance of a threshold provides timely indication of performance issues,
- The program burden associated with the activity is the minimum needed to satisfy the above.

For example, probability of loss of a specific mission cannot be directly measured, but depends on many quantities that can be measured up to a point, such as lower-level reliability and availability metrics.

Monitoring protocols are established to clarify monitoring requirements, assign responsibility for monitoring, and to establish monitoring intervals. The results of monitoring are collected and analyzed, and responses are triggered if performance thresholds are exceeded. These protocols also determine when the analyses must be updated. For example, technical risk management decisions should be reassessed with analysis if the goals of the program change. Due to the long lead-time required for the high-technology products required by NASA programs, program requirements often change before the program completes its life cycle. These changes may include technical requirements, budget or schedule, risk tolerance, etc.

### **Tracking and Controlling Performance Deviations (CRM Steps: “Track,” “Control”)**

As shown in Figure 6-18, tracking is the process by which parameters are observed, compiled, and reported according to the risk management plan. Risk mitigation/control is triggered when a performance threshold is exceeded, when risk that was assumed to be insignificant is found to be significant, or when risk that was not addressed during the analyses is discovered. Control may also be required if there are significant changes to the program. The need to invoke risk control measures in light of program changes is determined in the risk management plan. Alternatives are proposed and analyzed, and a preferred alternative is chosen based on the performance of the alternatives with respect to the TPMs, sensitivity and uncertainty analyses, and deliberation by the stakeholders. The new preferred alternative is then subjected to planning, tracking, and control.

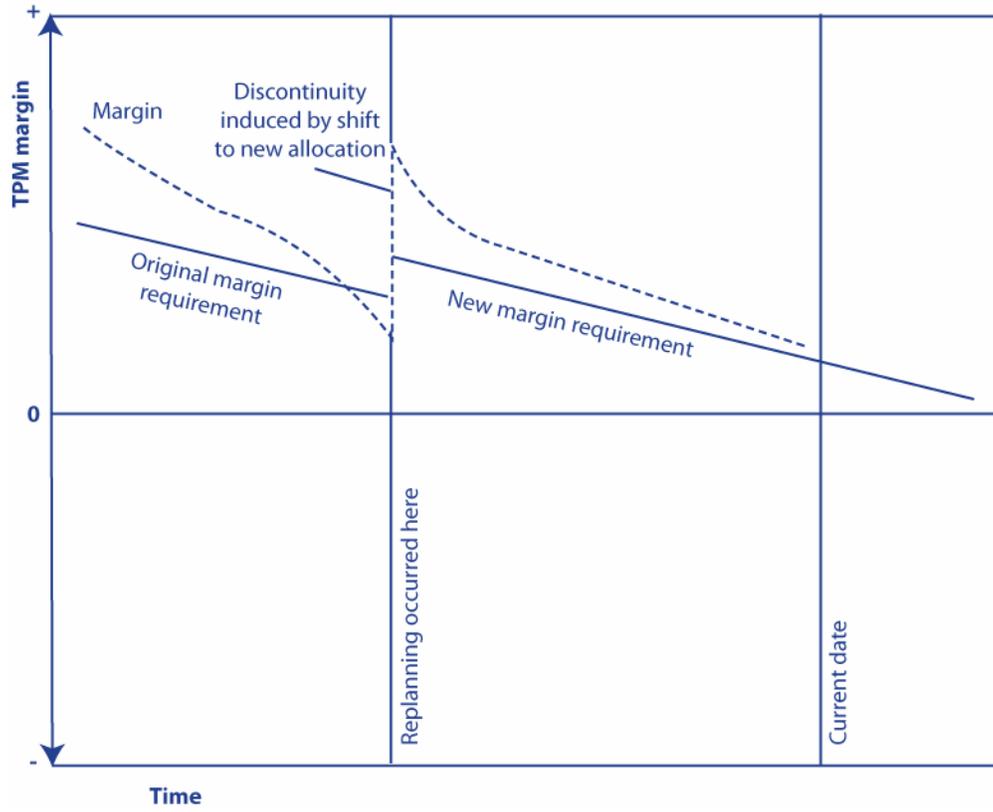


**Figure 6-18 Performance Monitoring and Control of Deviations**

During the planning phase, control alternatives were proactively conceived before there was an indication that they would be required. Once a threshold is triggered, a risk control action (as described in 5.4.3.2.3) is required. At this point, there may be considerably more information available to the decision-maker than existed when the control alternatives were proposed. Therefore, new alternatives or modifications of existing alternatives should be considered in addition to the existing alternatives by iterating this technical risk management process.

Figure 6-19 shows an example of tracking and controlling performance by tracking TPM margins against predetermined thresholds. At a point in time corresponding with the vertical break, the TPM's margin is less than the required margin. At this point, the alternative was changed, such that the margin and margin requirement increased.

Technical risk management is not exited until the program terminates, although the level of activity varies according to the current position of the activity in the life cycle. The main outputs are the technical risk reports, including risk associated with proposed alternatives, risk control alternatives, and decision support data. Risk control alternatives are fed back to technical planning as more information is learned about the alternatives' risk. This continues until the risk management plan is established. This learning process also produces alternatives, issues or problems and supporting data that are fed back into the project. Once a project baseline is chosen, technical risk management focuses on measuring the deviation of project risk from this baseline, and driving decision support requests based on these measurements.



**Figure 6-19 Margin Management Method**

## 6.5 Configuration Management

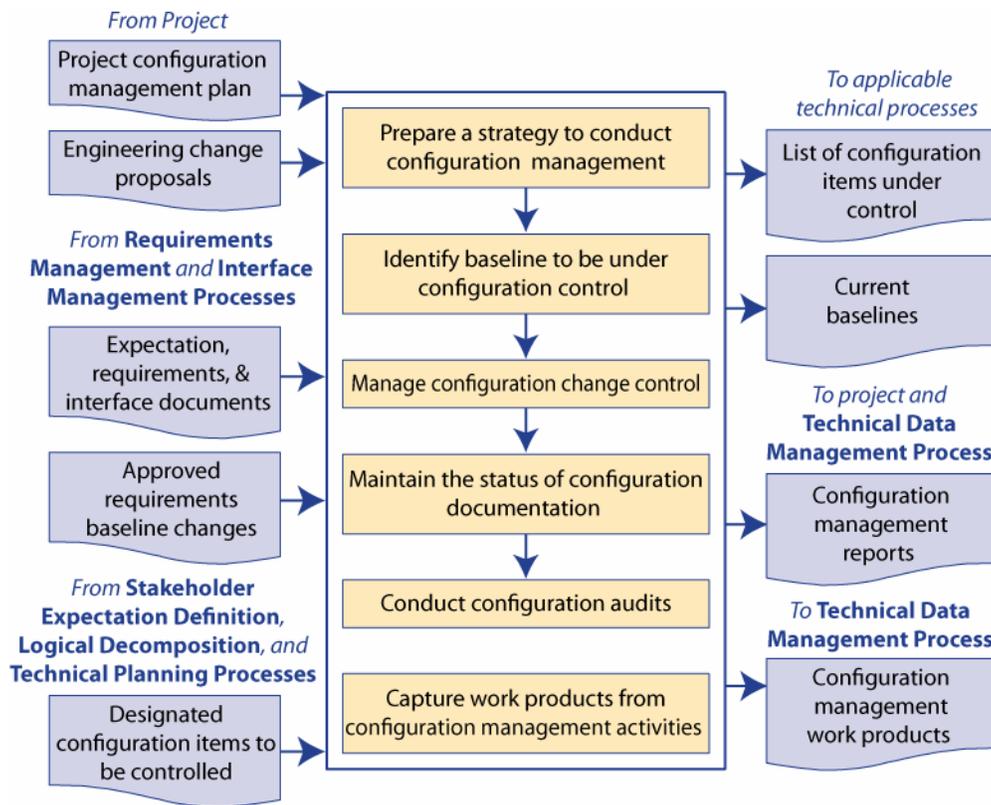
Configuration Management (CM) is a management discipline applied over the product's life cycle to provide visibility and to control performance and functional and physical characteristics. CM ensures that the configuration of a product is known and reflected in product information, that any product change is beneficial and is effected without adverse consequences, and that changes are managed.

Configuration management reduces technical risks by ensuring correct product configurations, distinguishes between product versions, ensures consistency between the product and information about the product, and avoids embarrassment of stakeholder dissatisfaction and complaint. NASA adopts the CM principles as defined by ANSI/EIA 649, NASA methods of implementation as defined by NASA CM professionals, and as approved by NASA management.

When applied to the design, fabrication/assembly, system/sub-system testing, integration, operational and sustaining activities of complex technology items, configuration management represents the "backbone" of the enterprise structure. It instills discipline and keeps the product attributes and documentation consistent. CM enables all stakeholders in the technical effort, at any given time in the life of a product, to use identical data for development activities and decision making. CM principles are applied to keep the documentation consistent with the approved engineering, and to ensure that the product conforms to the functional and physical requirements of the approved design.

### 6.5.1 Process Description

Figure 6-20 provides a typical flow diagram for the Configuration Management Process and identifies typical inputs, outputs, and activities to consider in addressing configuration management.



**Figure 6-20 CM Process**

### 6.5.1.1 Inputs

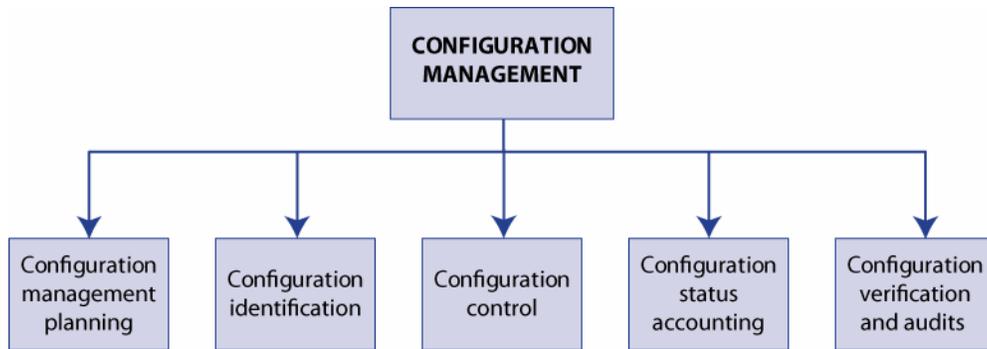
The required input for this process are

- Configuration Management Plan
- Work products to be controlled
- Proposed baseline changes.

### 6.5.1.2 Process Activities

There are five elements of Configuration Management (see Figure 6-21 **Error! Reference source not found.**):

1. CM Planning
2. Configuration Identification
3. Configuration Control
4. Configuration Status Accounting (CSA)
5. Configuration Verification and Audits



**Figure 6-21 Five Elements of Configuration Management**

### **CM Planning**

CM planning starts at a program or project’s inception. The *CM Office* must carefully weigh the value of prioritizing resources into CM tools or into CM surveillance of the contractors. Reviews by the NASA Configuration Management Organization (CMO) are warranted and will cost resources and time, but the correction of systemic CM problems before they erupt into “losing configuration control” are always preferable to explaining why incorrect or misidentified parts are causing major problems in the program/project.

One of the key inputs to *Preparing for Configuration Management Implementation* is a strategic plan for the project’s complete configuration management process. This is typically contained in a *Configuration Management Plan*. See appendix 6-5 for an outline of a typical CM Plan.

This plan has both internal and external uses:

- **Internal.** It is used within the project office to guide, monitor and measure the overall CM process. It describes both the CM activities planned for future acquisition phases and the schedule for implementing those activities.
- **External.** The CM Plan is used to communicate the CM process to the contractors involved in the program. It establishes consistent CM processes and working relationships.

The CM Plan may be a stand-alone document, or it may be combined with other program planning documents. It should describe the criteria for each technical baseline creation, technical authority approvals, and audits.

### **Configuration Identification**

Configuration identification is the systematic process of selecting, organizing and stating the product attributes. Identification requires unique identifiers for a product and its configuration documentation. The configuration management activity associated with identification includes selecting configuration documents, assigning and applying unique identifiers to a product, its components, and associated documents, and maintaining document revisions to product configurations or baselines.

NASA has four baselines, each of which defines a distinct phase in the evolution of a product design. The baseline identifies an agreed-to description of attributes of a Configuration Item (CI) at a point in time and provides a known configuration to which changes are addressed. Baselines are established by agreeing to (and documenting) the stated definition of a CI's attributes. The approved "current" baseline defines the basis of the subsequent change. The System Specification is typically finalized following the Systems Requirements Review (SRR). The functional baseline is established at the Systems Definition Review (SDR) and will usually transfer to NASA's control at that time.

The four baselines (see Figure 6-22) normally controlled by the Program/Project/Center are the following:

- **Functional Baseline.** The Functional Baseline is the approved configuration documentation that describes a system's or top-level CIs performance requirements (functional, interoperability, and interface characteristics) and the verification required to demonstrate the achievement of those specified characteristics. The Functional Baseline is controlled by the NASA.
- **Allocated Baseline.** The Allocated Baseline is the approved performance-oriented configuration documentation for a CI to be developed that describes the functional and interface characteristics that are allocated from a higher level requirements document or a CI and the verification required to demonstrate achievement of those specified characteristics. The allocated baseline extends the top-level performance requirements of the functional baseline to sufficient detail for initiating manufacturing or coding of a CI. The Allocated Baseline is controlled by the NASA. The allocated baseline(s) is typically established at the Preliminary Design Review (PDR). Control of the allocated baseline which would normally occur following the Functional Configuration Audit (FCA).
- **Product Baseline.** The Product Baseline is the approved technical documentation that describes the configuration of a CI during the production, fielding/deployment and operational support phases of its life cycle. It is not established until successful certification following a Functional Configuration Audit and a Physical Configuration Audit or alternate configuration management review process. Once the Product Baseline is established, the NASA determines how it will be controlled. The product baseline(s) is typically established at the Critical Design Review (CDR). The product baseline describes:
  - Detailed physical or form, fit, and function characteristics of a CI,
  - The selected functional characteristics designated for production acceptance testing,
  - The production acceptance test requirements.
- **As-Deployed Baseline.** The As-Deployed Baseline occurs at the Operational Readiness Review (ORR). At this point, the design is considered to be functional and ready for flight. All changes will have been incorporated into the documentation.

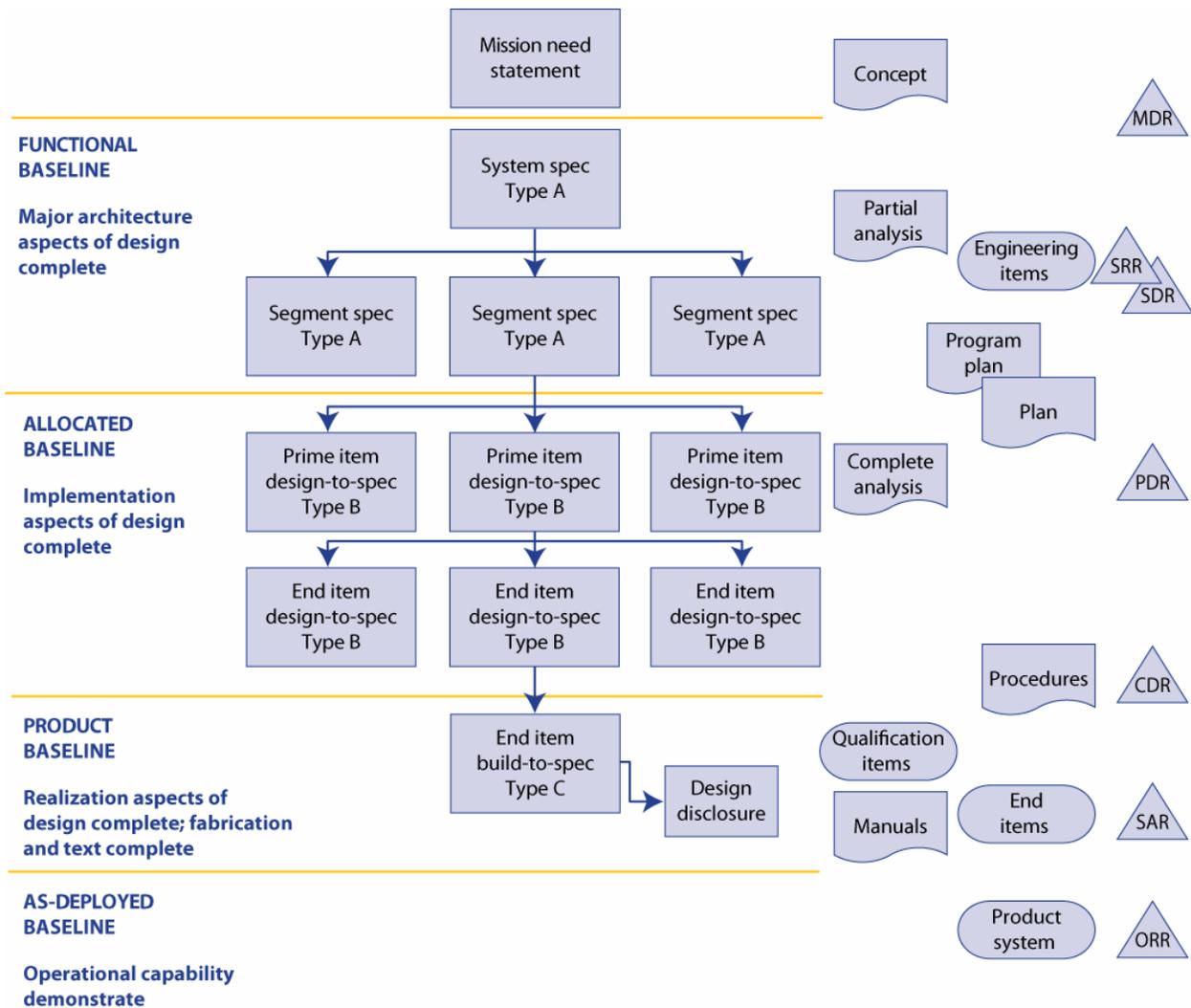


Figure 6-22 Evolution of Technical Baseline

## Configuration Control

Configuration control is change management or controlling changes to a product using a systemic change process. Configuration control is the proposing, justification, evaluation, coordination, disposition, and implementation of proposed changes. It encompasses the implementation of all approved changes in the configuration of a CI after establishment of the configuration baseline(s) for the CI.

Implementing CM in a given program requires unique knowledge of the program objectives and requirements.. The first step establishes a robust and well disciplined internal NASA Configuration Control Board (CCB) system, which chaired by someone with program change authority.. CCB members represent the stakeholders with authority to commit the team they represent. The second step creates CM surveillance of the contractor's activity. The *CM Office* advises the NASA Program/Project Manager to achieve a balanced CM implementation that suits

the unique program/project situation. See Figure 6-23 for an example of a typical change control process.

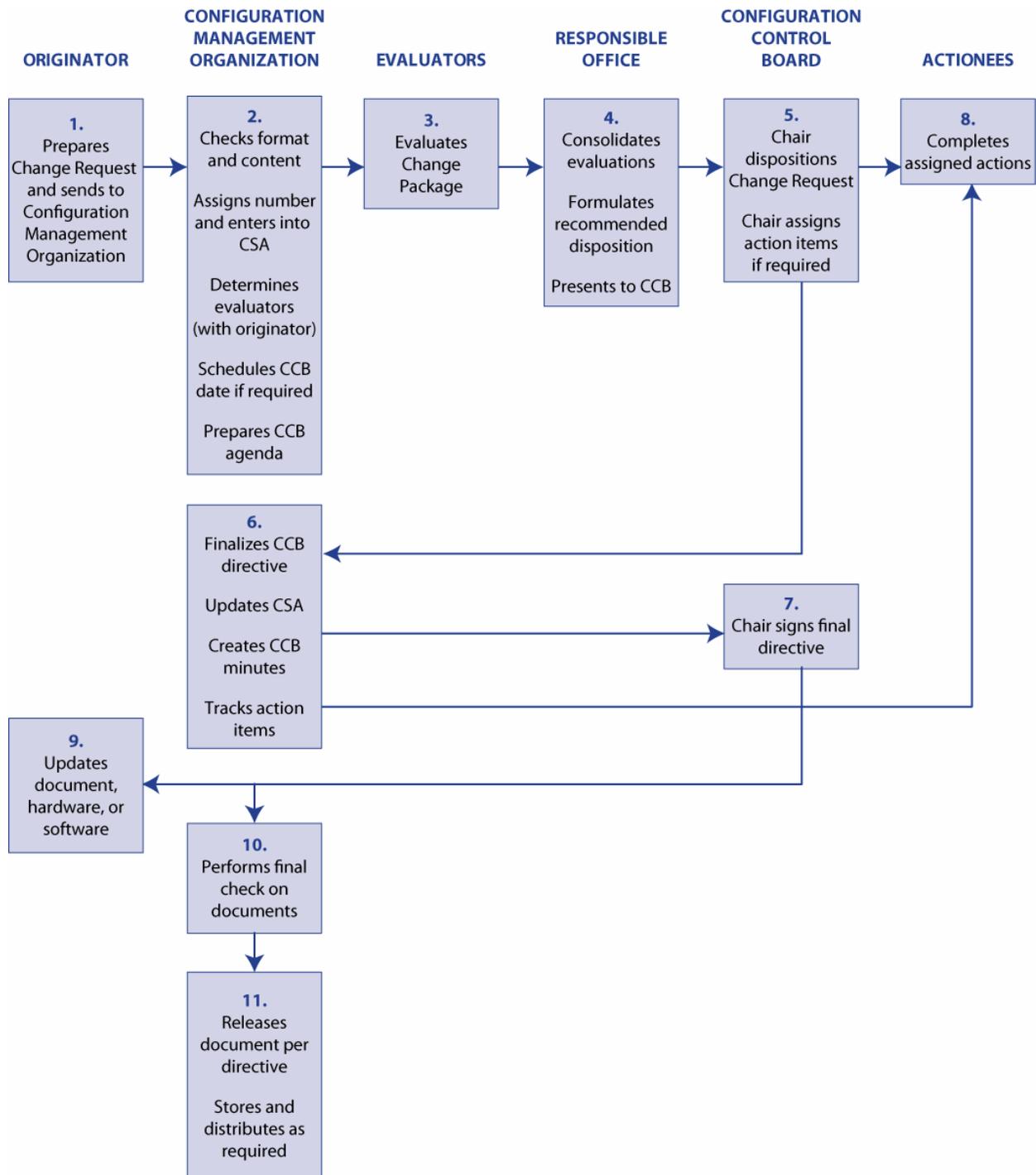


Figure 6-23 Typical Change Control Process

## Types of Configuration Control Changes

### Engineering Change

Alteration in baseline (draft or established) can be major or minor changes. May or may not include a specification change. Changes affecting an external interface must be coordinated and approved by all customers affected.

—“Major change”—a change to baselined configuration documentation which has significant impact (i.e., affects baselined specification; affects cost; requires retrofit of delivered products; affects safety, compatibility with interfacing products, operator or maintenance training)

—“Minor change”—correct or modify configuration documentation or processes without impacting interchangeability of products or system elements in the system structure.

### Variance

A variance is a written permit to depart from a particular requirement or set of requirements of a product's current approved baseline for a specific number of units or a specific period of time. Authorized variances do not constitute change to a baseline.

## Configuration Status Accounting

Priority within the design/build enterprise should be placed on providing access to all who need accurate and verified CM data. The CSA process accounts for the configuration of a product throughout the life cycle. In providing a historical record, design documentation for manufacturing and other stakeholders is maintained. CSA ensures that information is systematically recorded, safeguarded, validated, and disseminated.

The following are critical functions or attributes to consider if designing or purchasing software to assist with the task of managing configuration.

1. Ability to share data real time with internal and external customers securely
2. Version Control and comparison, track history of an object or product
3. Secure User checkout and check in
4. Tracking capabilities for gathering metrics, i.e. time, date, who, time in phases, etc.
5. Web based
6. Notification capability via email
7. Integration with other databases or legacy systems
8. Compatible with required support contractors and/or suppliers, i.e. can accept data from a 3<sup>rd</sup> party as required
9. Integration with drafting and modeling programs as required
10. Provide neutral format viewer for users

11. License agreement allows for multi users within an agreed to number
12. Workflow and Life-Cycle Management
13. Limited customization
14. Migration support for software upgrades
15. User friendly
16. Consideration for users with limited access
17. Ability to attach standard format files from desktop
18. Work flow capability, i.e. route a CI as required based on a specific set of criteria
19. The tool must be capable acting as the one and only source for released information

### **Configuration Verification and Audits**

Configuration verification is accomplished by inspecting documents, products and records; reviewing procedures, processes, and systems of operations to verify that the product has achieved its required performance requirements and functional attributes; and that the product's design is documented. This is sometimes divided into functional and physical configuration audits (FCA/PCA). See technical reviews (section 6.7) for more information.

#### *6.5.1.3 Outputs*

NPR 7120.5 defines a project's life cycle in progressive phases. Beginning with Pre-phase A, these steps in turn are grouped under the headings of Formulation and Implementation. Approval is required to transition between these phases. Key Decision Points (KDPs) define transitions between the phases. CM plays an important role in determining whether a KDP has been met. Major outputs of Configuration Management are procedures, approved baseline changes, configuration status and audit reports.

### **6.5.2 Configuration Management Guidance**

#### *6.5.2.1 What Is the Impact of Not Doing CM?*

The impact of CM is often described after the result of investigations into serious disasters. During the Columbia Accident Investigation Board (CAIB), the Board found inconsistencies related to the hardware and the documentation with "unincorporated documentation changes" that led to failure. No CM issues were cited as a cause of the accident. The usual impact of not implementing CM can be described as "losing configuration control." Within NASA, this has occurred (X-37 Program), resulting in program delays and engineering issues, especially in fast prototyping developments where schedule has priority over recording what is being done to the hardware. If CM procedures are followed, any errors would be found during the inspection of a

Physical Configuration Audit (PCA,) comparing the hardware and software to the documentation. The following impacts are possible and have occurred in the past:

- Mission failure and loss of property and life due to improperly configured or installed hardware or software
- Mission failure to gather mission data due to improperly configured or installed hardware or software
- Significant Mission delay incurring additional cost due to improperly configured or installed hardware or software.
- Significant mission costs or delay due to improperly certified parts/subsystems due to fraudulent verification data

If CM is not implemented properly, problems may occur in the any area, including the following: manufacturing, quality, receiving, and procurement. The user will also experience problems if ILS data is not maintained. Using a shared software system that can route and track tasks provides the team with the resources necessary for a successful project.

### **Warning Signs/Red Flags (How Do You Know When You're in Trouble?)**

General warning signs of an improper implementation of CM include the following:

1. Failure of program to define the "Top Level" Technical requirement ("We don't need a spec").
2. Failure of program to recognize the baseline activities that precede and follow design reviews
3. Program Office reduces the time to evaluate changes to one that is impossible for Engineering, S&MA or other CCB members to meet
4. Program Office declares "there will be no dissent in the record" for CCB documentation.
5. Contract is awarded without CM requirements concurred in by CMO supporting the program office
6. "Redlines" used inappropriately on production floor to keep track of changes to design
7. Material Review Board (MRB) does not know the difference between critical, major and minor non-conformances and the appropriate classification of waivers
8. Drawings are not of high quality and do not contain appropriate notes to identify Critical Engineering Items for configuration control or appropriate tolerancing
9. Vendors do not understand the implication of submitting waivers to safety requirements as defined in engineering
10. Subcontractors/vendors change engineering design without approval of integrating contractor, do not know how to coordinate and write Engineering Change Request, etc.
11. Manufacturing tooling engineering does not keep up with engineering changes that affect tooling concepts, manufacturing tools "loose configuration control" and certification for production
12. Verification data can not be traced to released part number and specification that applies to verification task
13. Operational manuals and repair instructions can not be traced to latest released part number and repair drawings that applies to repair/modification task
14. Maintenance and ground support tools and equipment can not be traced to latest released part number and specification that applies to equipment.
15. Parts and items can not be identified due to improper identification markings
16. Digital Closeout Photography (DCOP) can not be correlated to the latest release engineering.
17. NASA is unable to verify the latest released engineering through access to the contractor's CM website
18. Tools required per installation procedures do not match the fasteners and nuts and bolts used in the design of CIs
19. CIs do not fit into their packaging crates and containers due to "loosing configuration control" in the design of the shipping and packing containers.
20. Supporting procurement/fabrication change procedures do not adequately involve approval by originating engineering organization.

#### *6.5.2.2 When Is It Acceptable to Use Redline Drawings?*

Redline refers to the control process of marking-up drawings and documents during design, fabrication, production, and testing that are found to contain errors or inaccuracies. Work stoppages could occur if the documents were corrected through the formal change process.

All redlines require the approval of the responsible hardware manager and Quality Assurance at a minimum. These managers will determine whether redlines are to be incorporated into the plan or procedure.

The important point is that each project must have a controlled procedure for redlines that specifies redline procedures and approvals.

**Redlines Were identified as One of the Major Causes of the NOAA-N Prime Mishap September 6, 2003**

Excerpts from the NOAA-N Prime Mishap Investigation Final Report, September 13, 2004

“Several elements contributed to the NOAA N-PRIME incident, the most significant of which were the lack of proper TOC [*Turn Over Card*] verification, including the lack of proper PA [*Product Assurance*] witness, the change in schedule and its effect on the crew makeup, the failure of the crew to recognize missing bolts while performing the interface surface wipe down, the failure to notify in a timely fashion or at all the Safety, PA, and Government representatives, and the improper use of procedure redlines leading to a difficult-to-follow sequence of events. The interplay of the several elements allowed a situation to exist where the extensively experienced crew was not focusing on the activity at hand. There were missed opportunities that could have averted this mishap.”

“In addition, the operations team was utilizing a heavily redlined procedure that required considerable ‘jumping’ from step to step, and had not been previously practiced. The poorly written procedure and novel redlines were preconditions to the decision errors made by the RTE [*Responsible Test Engineer*].”

“The I&T [*Integration and Test*] supervisors allowed routine poor test documentation and routine misuse of procedure redlines.”

“Key processes that were found to be inadequate include those that regulate operational tempo, operations planning, procedure development, use of redlines, and GSE [*Ground Support Equipment*] configurations. For instance, the operation during which the mishap occurred was conducted using extensively redlined procedures. The procedures were essentially new at the time of the operation—that is, they had never been used in that particular instantiation in any prior operation. The re-written procedure had been approved through the appropriate channels even though such an extensive use of redlines was unprecedented. Such approval had been given without hazard or safety analyses having been performed.”

## 6.6 Technical Data Management

Data Management is used to plan for, acquire, access, manage, protect, and use data of a technical nature to support the total life cycle of a system. This includes its development, deployment, operations and support, eventual retirement, and retention of appropriate technical data beyond system retirement as required by current NASA policies.

Data Management is illustrated in Figure 6-24. Key aspects of DM for Systems Engineering include:

- Application of policies and procedures for data identification and control
- Timely and economical acquisition of technical data
- Assurance of the adequacy of data and its protection
- Facilitating access and distribution the data to the point of use
- Analysis of data use
- Evaluation of data for future value to other programs/projects
- Process access to information written in legacy software

### 6.6.1 Process Description

Figure 6-24 provides a typical flow diagram for the Technical Data Management Process and identifies typical inputs, outputs, and activities to consider in addressing technical data management.

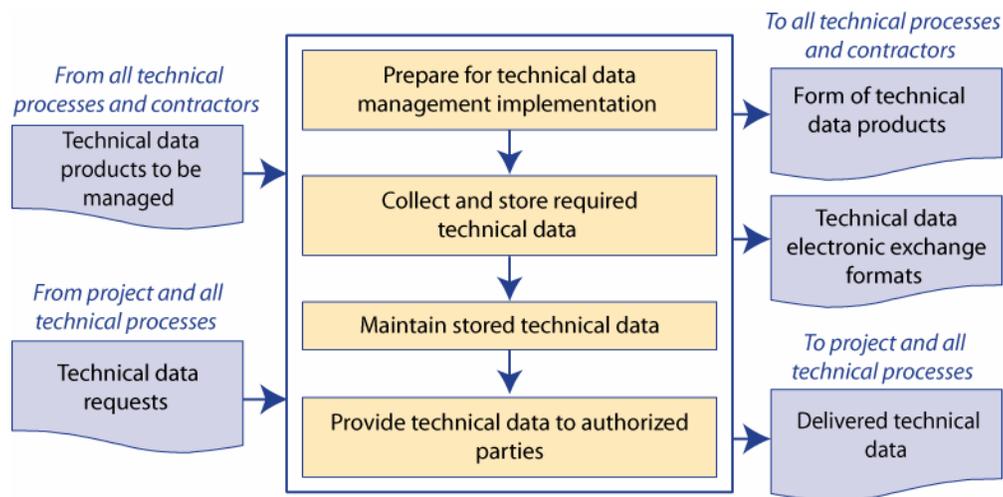


Figure 6-24 Technical Data Management

#### 6.6.1.1 Inputs

Inputs include technical data, regardless of the form or method of recording and whether the data is generated by the contractor or government during the life cycle of the system being developed. Major inputs to the SE Data Management Process include:

- Program DM Plan
- Data Products to be Managed
- Data Requests

#### 6.6.1.2 *Process Activities*

Each Center is responsible for policies and procedures for technical DM. NPR 7120.5 and NPR 7123.1 define the need to manage data, but leave specifics to the individual Centers. However, NPR 7120.5 does require that *DM planning* be provided as either a section in the Program/Project Plan or as a separate document. The Program or Project Manager is responsible for ensuring that the data required is captured, stored, data integrity maintained, and disseminated as required.

Other NASA policies address the acquisition and storage of data and not just the technical data used in the life cycle of a system.

#### **Role of Data Management Plan**

The recommended procedure is that the DM Plan be a separate plan apart from the Program/Project Plan. DM issues are usually of sufficient magnitude to justify a separate plan. The lack of specificity in Agency policy and procedures provides further justification for a more detailed DM planning. The Plan should cover the following major DM topics:

- Identification/Definition of Data Requirements for all aspects of the product life cycle.
- Control Procedures—receipt, modification, review and approval
- Data exchange formats that promote data reuse and help to ensure that data can be used consistently throughout the system, family of systems, or system of systems.
- Data rights and distribution limitations like export-control Sensitive But Unclassified (SBU)
- Storage and maintenance of data, including Master lists where documents and records are maintained and managed

#### **Technical Data Management Key Considerations**

Subsequent activities collect, store and maintain technical data and provide it to authorized parties as required. Some considerations which impact these activities for implementing Technical Data Management include:

- Requirements relating to the flow/delivery of data to or from a contractor should be specified in the Technical Data Management Plan and included in the Request for Proposal and contractor agreement.

- The government should not impose changes on existing contractor data management systems unless the program technical data management requirements, including data exchange requirements, cannot otherwise be met.
- Responsibility for data inputs into the technical data management system lies solely with the author, originator or generator of the data.
- The availability/access of technical data will lie with the author, originator or generator of the data in conjunction with the manager of the technical data management system.
- The established availability/access description and list should be baselined and placed under configuration control.
- For new programs, a digital generation and delivery medium is desired. Existing programs must weigh the cost/benefit trades of digitizing hard copy data.

### **General Data Management Roles**

The Data Management Process provides the basis for applying the policies and procedures to identify and control data requirements; to responsively and economically acquire, access, and distribute data; and to analyze data use.

Adherence to data management principles/rules enables the sharing, integration, and management of data for performing technical efforts by government and industry, and ensures that information generated from managed technical data satisfies requests or meets expectations.

Technical Data Management has a leading role in capturing and organizing technical data and providing information for the following uses:

- Identifying, gathering, storing and maintaining the work products generated by other Systems Engineering Technical and Technical Management Processes as well as the assumptions made in arriving at those work products
- Enabling collaboration and life cycle use of system product data.
- Capturing and organizing technical effort inputs, as well as current, intermediate, and final outputs.
- Data correlation and traceability among requirements, designs, solutions, decision, and rationale.
- Documenting engineering decisions, including procedures, methods, results, and analyses.
- Facilitating technology insertion for affordability improvements during re-procurement and post-production support; and
- Supporting other Technical Management Processes and Technical Processes, as needed.

## Data Identification/Definition

Each Program/Project determines data needs during the life cycle. Data types may be defined in standard documents. Center and Agency directives sometimes specify content of documents and are appropriately used for in-house data preparation. The standard description is modified to suit program/project specific needs, and appropriate language to implement actions described in data is included in statements of work. Data Suppliers may be a contractor, academia or the government. Procurement of data from an outside Supplier is a formal procurement action that requires a procurement document; “in-house” requirements may be handled in a less formal method. Below are the different types of data that might be utilized within a program/project:

- **Data**
  - Data is defined in general as ‘recorded information regardless of the form or method of recording’. However, the terms ‘data’ and ‘information’ are frequently used interchangeably. To be more precise, data generally must be processed in some manner to generate useful, actionable information.
  - The term ‘data’ as used in SE Data Management includes technical data, computer software documentation, representation of facts, numbers, or datum of any nature that can be communicated, stored, and processed to form information required by a contract or agreement to be delivered, or accessed by, the government.
  - ‘Data’ includes that associated with system development; modeling and simulation used in development or test; test and evaluation; installation; parts; spares; repairs; usage data required for product sustainability; and source and/or supplier data.
  - Data specifically not included in Technical Data Management would be data relating to general NASA workforce operations information; communications information (except where related to specific requirement); financial transactions; personnel data; transactional data; and other data of a purely business nature.
- **Data Call.** Solicitation from government stakeholders (specifically IPT leads and functional managers) identifies and justifies their data requirements from a proposed contracted procurement. Since data provided by contractors has a cost to the government, a Data Call (or an equivalent activity) is a common control mechanism used to ensure that the requested data is truly needed. If approved by the Data Call, a description of each data item needed is then developed and placed on contract.
- **Information.** Information is generally considered as processed data. The form of the processed data is dependent on the documentation, report, review formats or templates that are applicable.
- **Technical Data Package (TDP).** A TDP is a technical description of an item adequate for supporting an acquisition strategy, production, engineering, and logistics support. The TDP defines the required design configuration and procedures to ensure adequacy of item performance. It consists of all applicable items such as drawings, associated lists, specifications, standards, performance requirements, Quality Assurance (QA) provisions, and packaging details.

- **Technical Data Management System.** The strategies, plans, procedures, tools, people, data formats, data exchange rules, databases, and other entities and descriptions are required to manage the technical data of a program.

#### **Inappropriate Uses of Technical Data**

Examples of inappropriate uses of technical data include:

- Unauthorized disclosure of classified data or data otherwise provided in confidence.
- Faulty interpretation based on incomplete, out-of-context, or otherwise misleading data.
- Use of data for parts or maintenance procurement for which at least Government Purpose Rights (GPR) have not been obtained.

Ways to help prevent inappropriate use of technical data include:

- Educate Stakeholders (IPTs) on appropriate data use
- Control access to data

### **Initial Data Management System Structure**

When setting up a data management system, it is not necessary to acquire (that is, to purchase and take delivery) all technical data that is generated on a project. Some data may be stored in other locations with accessibility provided to those on a need to know basis.. Data should be purchased only when such access is not sufficient, timely or secure enough to provide for responsive life-cycle planning and system maintenance. Data Calls are a common control mechanism to help address this need.

### **Data Management Planning**

- Prepare a technical data management strategy. This strategy can document how the program Data Management Plan will be implemented by the technical effort or, in the absence of such a program-level plan, be used as the basis for preparing a detailed technical Data Management Plan, including:
  - Items of data that will be managed according to program or organizational policy, agreements or legislation.
  - The data content and format.
  - A framework for data flow within the program and to/from contractors including the language(s) to be employed in technical effort information exchanges.
  - Technical data management responsibilities and authorities regarding the origin, generation, capture, archiving, security, privacy and disposal of data products.
  - Establish the rights, obligations and commitments regarding the retention of, transmission of and access to data items.
  - Relevant data storage, transformation, transmission and presentation standards and conventions to be used according to program or organizational policy, agreements or legislative constraints.

- Obtain strategy/plan commitment from relevant stakeholders.
- Prepare procedures for implementing the technical data management strategy for the technical effort and/or for implementing the activities of the technical data management plan.
- Establish a technical database(s) to use for technical data maintenance and storage or work with the program staff to arrange use of the program database(s) for managing technical data.
- Establish data collection tools, as appropriate to the technical data management scope and available resources. See section on Tools.
- Establish electronic data exchange interfaces in accordance with international standards/agreements and applicable NASA standards.
- Train appropriate Stakeholders (IPTs) and other technical personnel in the established technical data management strategy/plan, procedures and data collection tools, as applicable.
- Expected Outcomes:
  - A strategy and/or plan for implementing technical data management.
  - Established procedures for performing planned Technical Data Management activities.
  - Master list of managed data and its classification by category and use is available.
  - Data collection tools established and available.
  - Qualified technical personnel capable of conducting established technical data management procedures and using available data collection tools.

### **Key Considerations: for Planning of Data Management and for a Tool Selection**

- All data entered into the technical data management system or delivered to a requester from the databases of the system should have traceability to the author, originator or generator of the data.
- All technical data entered into the technical data management system should carry objective evidence of its current status (for approval, for agreement, for information, etc.), version/control number, and date.
- The technical data management approach should be covered as part of the program's Systems engineering Management Plan (SEMP)
- Technical data expected to be used for re-procurement of parts, maintenance services, etc. might need to be reviewed by the Center's legal counsel

Careful consideration should be taken when planning the data access and storage of data that will be generated from a project or program. If a system or tool is needed, many times the Configuration Management tool can be used with less formality. If a separate tool is required to manage the data refer to the section below for some best practices when evaluating a data management tool. Priority must be placed on being able to access the data and ease of inputting the data. Second priority should be the consideration of the value of the specific data to current project/program, future programs/projects, NASA's overall efficiency, and uniqueness to NASA's engineering knowledge.

The following are critical functions or attributes to consider if designing or purchasing software to assist with the task of managing data:

- Ability to share data with internal and external customers securely
- Version Control and comparison, track history of an object or product
- Secure User updating
- Access Control down to the file level
- Web based
- Ability to link data to Configuration Management system or elements
- Compatible with required support contractors and/or suppliers, i.e. can accept data from a 3<sup>rd</sup> party as required
- Integrate with drafting and modeling programs as required
- Provide neutral format viewer for users
- License agreement allows for multi user seats
- Workflow and Life-Cycle Management is a suggested option
- Limited customization
- Migration support between software version upgrades
- User friendly
- Straight forward search capabilities
- Ability to attach standard format files from desktop

## Value of Data

Storage of engineering data needs to be planned at the beginning of a program or project. Some of the data types will fall under the control of Records Retention, NPR 1440.1; those that do not will have to be addressed. It is best to evaluate all data that will be produced and decide how long it is of value to the program or project or to NASA Engineering as a whole. There are four basic questions to ask when evaluating data's value:

- Does the data describe the product/system that is being developed or built?
- Is the data required to accurately produce the product/system being developed or built?
- Does the data offer insight for similar future programs or projects?
- Does the data hold key information that needs to be maintained in NASA's knowledge base for future engineers to use or kept as a learning example?

## Technical Data Capture Tasks

The following chart defines the tasks required to capture technical data:

**Table 6-1 Technical Data Tasks**

Description	Tasks	Expected Outcomes
<b>Technical Data Capture</b>	<ul style="list-style-type: none"> <li>• Collect and store inputs and technical effort outcomes from the technical and technical management processes, including:               <ul style="list-style-type: none"> <li>○ Results from technical assessments.</li> <li>○ Descriptions of methods, tools and metrics used.</li> <li>○ Recommendations, decisions, assumptions and impacts of technical efforts and decisions.</li> <li>○ Lessons learned.</li> <li>○ Deviations from plan.</li> <li>○ Anomalies and out-of-tolerances relative to requirements.</li> <li>○ Other data for tracking requirements</li> </ul> </li> <li>• Perform data integrity checks on collected data to ensure compliance with content and format as well as for technical data check to ensure there are no errors in specifying or recording the data.</li> <li>• Report integrity check anomalies or variances to the authors or generators of the data for correction.</li> <li>• Prioritize, review, and update data collection and storage procedures as part of regularly scheduled maintenance.</li> </ul>	<ul style="list-style-type: none"> <li>• Sharable data needed to perform and control the technical and technical management processes is collected and stored.</li> <li>• Stored data inventory.</li> </ul>
<b>Technical Data Maintenance</b>	<ul style="list-style-type: none"> <li>• Implement: technical management roles and responsibilities with technical data products received.</li> <li>• Manage database(s) to ensure that collected data has proper quality and integrity, is properly retained,</li> </ul>	<ul style="list-style-type: none"> <li>• Records of technical data maintenance.</li> <li>• Technical effort</li> </ul>

Description	Tasks	Expected Outcomes
	<p>secure, and available to those with access authority.</p> <ul style="list-style-type: none"> <li>• Periodically review technical data management activities to ensure consistency and identify anomalies and variances.</li> <li>• Review stored data to ensure completeness, integrity, validity, availability, accuracy, currency and traceability.</li> <li>• Perform technical data maintenance, as required.</li> <li>• Identify and document significant issues, their impacts and changes made to technical data to correct issues and mitigate impacts.</li> <li>• Maintain control and prevent the stored data from being used inappropriately.</li> <li>• Store data in a manner that enables easy and speedy retrieval.</li> <li>• Maintain stored data in a manner that protects the technical data against foreseeable hazards, e.g., fire, flood, earthquake, etc.</li> </ul>	<p>data, including captured work products, contractor delivered documents and acquirer-provided documents, is controlled and maintained.</p> <ul style="list-style-type: none"> <li>• Status of data stored is maintained, to include: version description, timeline and security classification.</li> </ul>
<b>Technical Data/Information Distribution</b>	<ul style="list-style-type: none"> <li>• Maintain an information library or reference index to provide technical data availability and access instructions.</li> <li>• Receive and evaluate requests to determine data requirements and delivery instructions.</li> <li>• Process special requests for technical effort data or information according to established procedures for handling such requests.</li> <li>• Ensure that required and requested data is appropriately distributed to satisfy the needs of the acquirer and requesters in accordance with the agreement, program directives and technical data management plans and procedures.</li> <li>• Ensure that electronic access rules are followed before database access is allowed or any requested data is electronically released/transferred to the requester</li> <li>• Provide proof of correctness, reliability and security of technical data provided to internal and external recipients.</li> </ul>	<ul style="list-style-type: none"> <li>• Access information (e.g., available data, access means, security procedures, time period for availability and personnel cleared for access) is readily available.</li> <li>• Technical data is provided to authorized requesters in the appropriate format, with the appropriate content and by a secure mode of delivery, as applicable.</li> </ul>
<b>Data Management System Maintenance</b>	<ul style="list-style-type: none"> <li>• Implement safeguards to ensure protection of the technical database and of enroute technical data from unauthorized access or intrusion.</li> <li>• Establish proof of coherence of the overall technical data set to facilitate effective and efficient use.</li> <li>• Maintain, as applicable, back-ups of each technical database.</li> <li>• Evaluate the technical data management system to identify collection and storage performance issues and problems; satisfaction of data users; risks associated with delayed or corrupted data, unauthorized access, or survivability of information from hazards such as fire, flood, earthquake, etc.</li> </ul>	<ul style="list-style-type: none"> <li>• Current technical data management system</li> <li>• Technical data is appropriately and regularly backed-up to prevent data loss.</li> </ul>

Description	Tasks	Expected Outcomes
	<ul style="list-style-type: none"> <li>• Review systematically the technical data management system, including the database capacity, to determine its appropriateness for successive phases of the Defense Acquisition Framework.</li> <li>• Recommend improvements for discovered risks and problems: <ul style="list-style-type: none"> <li>○ Handle risks identified as part of technical risk management.</li> <li>○ Control recommended changes through established program change management activities.</li> </ul> </li> </ul>	

<b>Data Collection Checklist</b>
<ol style="list-style-type: none"> <li>1) Have the frequency of collection and the points in the technical and technical management processes when data inputs will be available been determined?</li> <li>2) Has the timeline that is required to move data from the point of origin to storage repositories or stakeholders been established?</li> <li>3) Who is responsible for the input of the data?</li> <li>4) Who is responsible for data storage, retrieval and security?</li> <li>5) Have necessary supporting tools been developed or acquired?</li> </ol>

### **Protection for Data Deliverables**

All data deliverables should include distribution statements and procedures to protect all data that contains critical technology information, as well as to ensure that limited distribution data, intellectual property data, or proprietary data are properly handled during systems engineering activities. This injunction applies whether the data are hard copy or digital.

As part of overall Program Protection Planning, NASA has established special procedures for the protection of so-called ‘Critical Program Information (CPI)’. CPI may include components; engineering, design, or manufacturing processes; technologies; system capabilities and vulnerabilities; and any other information that gives a system its distinctive operational capability.

CPI protection should be a key consideration for the Technical Data Management effort.

#### *6.6.1.3 Outputs*

Outputs include timely, secure availability of needed data in various representations to those authorized to receive it. Major outputs from the technical Data Management Process include (refer to Figure 6-24):

- Technical Data Management Procedures
- Data Representation Forms
- Data Exchange Formats

- Requested Data/Information Delivered

## 6.6.2 Technical Data Management Guidance

### Data Security and ITAR

NASA generates an enormous amount of information, much of which is unclassified/nonsensitive in nature with few restrictions on its use and dissemination. NASA also generates and maintains classified national security information (CNSI) under a variety of Agency programs, projects, and through partnerships and collaboration with other federal agencies, academia, and private enterprises. Sensitive But Unclassified (SBU) markings requires the author, distributor, and receiver to keep control of the sensitive document and data or pass the control to an established control process. Public release is prohibited and a document/data marked as such must be transmitted by secure means. Secure means are encrypted e-mail, secure fax, or person-to-person tracking. WebEx is a non-secure environment. Standard e-mail is not permitted to transmit SBU documents and data. A secure way to send SBU information via e-mail is using the Public Key Infrastructure (PKI) to transmit the file(s). PKI is a system that manages keys to lock and unlock computer data. The basic purpose of PKI is to enable you to share your data keys with other people in a secure manner. PKI provides desktop security, as well as security for desktop and network applications, including electronic and Internet commerce.

Data items such as detailed design data (models, drawings, presentations, etc), limited rights data, source selection data, bid and proposal information, financial data, emergency contingency plans, and restricted computer software are all examples of SBU data. Items that are deemed SBU must be conspicuously and clearly marked on each and every page that has SBU data. Pages that are not marked are not viewed as SBU data. Data or items that cannot be directly marked, like computer models and analyses must have an attached copy of NASA form 1686 that indicates the entire package is SBU data. Documents are required to have a NASA form 1686 as a cover sheet. SBU documents and data should be safeguarded. Some examples of ways to safeguard SBU data are: access is limited on a need-to-know basis, items are copy controlled, items are attended while being used, items are properly marked (document header, footer, and NASA form 1686), items are stored in locked containers or offices and secure servers, transmitted by secure means, and destroyed by approved methods (shredding, etc.). For more information on SBU data, see *NPR 1600.1, NASA Security Program Procedural Requirements*.

ITAR regulations implement the Arms Export Control Act, and contain the United States Munitions List (USML). The USML lists articles, services, and related technical data that are designated as "Defense Articles" and "Defense Services," pursuant to sections 38 and 47(7) of the Arms Export Control Act. The ITAR is administered by the U.S. Department of State. "Technical Data" as defined in the ITAR does not include information concerning general scientific, mathematical, or engineering principles commonly taught in schools, colleges, and universities or information in the public domain (as that term is defined in 22 CFR 120.11). It also does not include basic marketing information on function and purpose or general system descriptions. For purposes of the ITAR, the following definitions apply.

- **"Defense Article" (22 CFR 120.6).** A "Defense Article" is any item or "Technical Data" on the United States Munitions List (22 CFR 121.1). The term includes "Technical Data" recorded or stored in any physical form, models, mockups, or other items that reveal "Technical Data" directly relating to items designated in the USML. Examples of "Defense Articles" included on the USML are (1) launch vehicles, including their specifically designed or modified components, parts, accessories, attachments, and associated equipment, (2) remote sensing satellite systems, including ground control stations for telemetry, tracking, and control of such satellites, as well as passive ground stations if such stations employ any cryptographic items controlled on the USML, or they employ any uplink command capability, all components, parts, accessories, attachments, and associated equipment (including ground support equipment) that is specifically designed, modified, or configured for such systems. (See 22 CFR 121.1 for the complete listing.)

- **"Technical Data" (22 CFR 120.10).** Information which is required for the design, development, production, manufacture, assembly, operation, repair, testing, maintenance, or modification of "Defense Articles." This includes information in the form of blueprints, drawings, photographs, plans, instructions, and documentation.
- **Classified information relating to "Defense Articles" and "Defense Services."** Information covered by an invention secrecy order (35 U.S.C. 181 et seq.; 35 CFR Part 5).
- **Software directly related to "Defense Articles,"** including, but not limited to, system functional design, logic flow, algorithms, application programs, operating systems, and support software for design, implementation, test, operations, diagnosis, and repair.

"Technical Data" as defined in the ITAR does not include information concerning general, scientific, mathematical, or engineering principles commonly taught in schools, colleges, and universities or information in the public domain (as that term is defined in 22 CFR 120.11). It also does not include basic marketing information on function and purpose or general system descriptions.

## 6.7 Technical Assessment

Technical assessment is the crosscutting process used to help monitor technical progress of a program/project through periodic technical reviews. It also provides status information in support of assessing system design, product realization, and to support technical management decisions.

### 6.7.1 Process Description

Figure 6-25 provides a typical flow diagram for the Technical Assessment Process and identifies typical inputs, outputs, and activities to consider in addressing technical assessment.

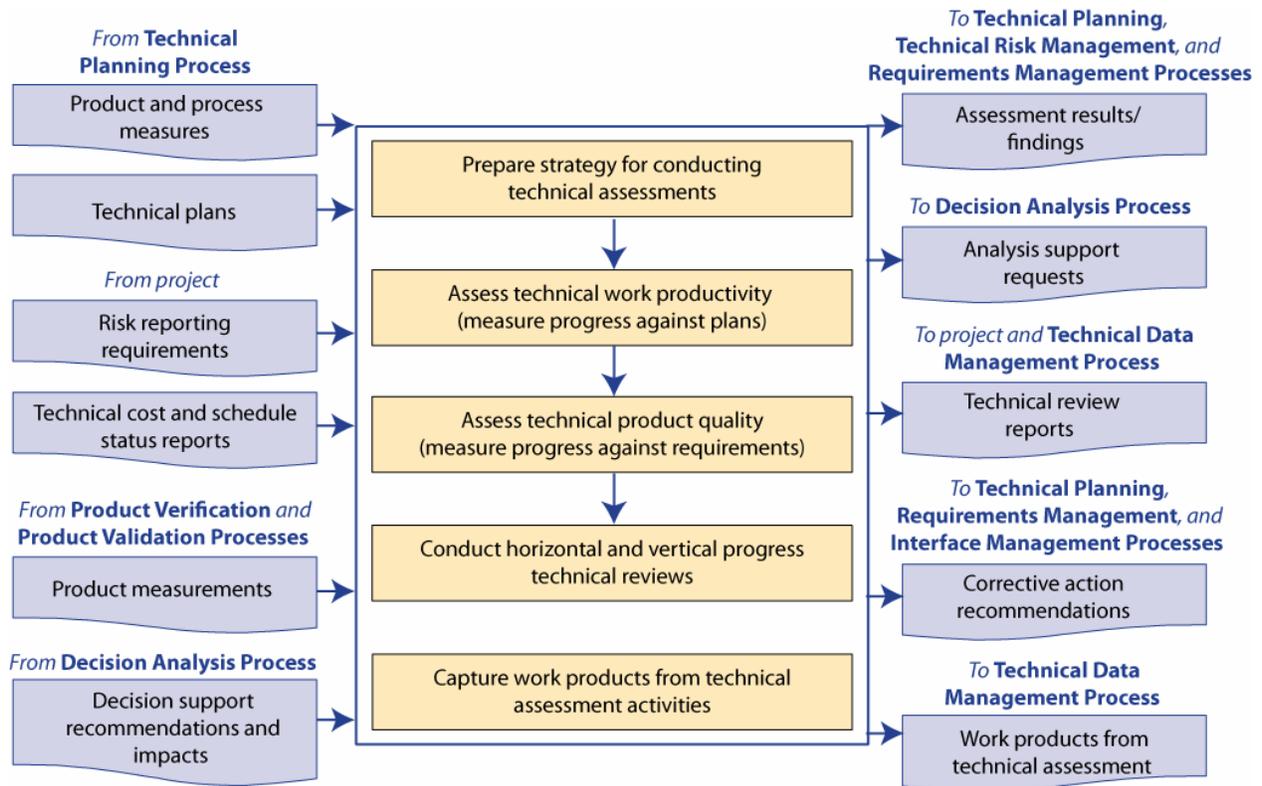


Figure 6-25 Technical Assessment Process

#### 6.7.1.1 Inputs

Typical inputs needed for the technical assessment process would include the following:

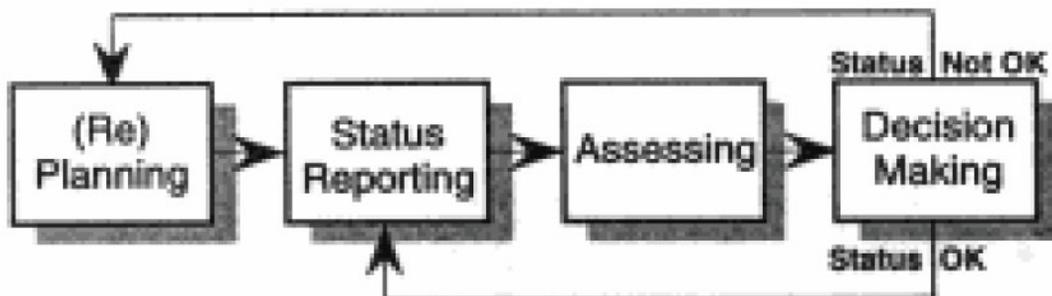
- Technical Plans:** These are the planning documents that will outline the technical reviews/assessment process as well as identify the technical product/process measures that will be tracked and assessed to determine technical progress. Examples of these plans will be the Systems Engineering Management Plan (SEMP), Review Plans, Earned Value Management (EVM) Plan.

- **Technical Measures:** These are the identified technical measures that will be tracked to determine technical progress. These measures are also referred to Measures of Effectiveness (MOEs), Measures of Performance (MOPs), and Technical Performance Measures (TPMs).
- **Reporting Requirements:** These are the requirements on the methodology in which the technical measures status will be reported in regards to risk, cost, schedule, etc. The methodology and tools used for reporting the status will be established on a project-by-project basis.

### 6.7.1.2 Process Activities

As outlined in Figure 6-25, the technical plans (e.g., SEMP, Review Plans) provide the initial inputs into the technical assessment process. These documents will outline the technical reviews/assessment approach as well as identify the technical measures that will be tracked and assessed to determine technical progress. An important part of the technical planning is determining what is needed in time, resources, and performance to complete a system that meets desired goals and objectives. Project managers need visibility into the progress of those plans in order to exercise proper management control. Typical activities in determining progress against the identified technical measures will include status reporting and assessing the data. *Status reporting will identify* where the project stands in regards to a particular technical measure. *Assessing will analytically convert* the output of the status reporting into a more useful form from which trends can be determined and variances from expected results can be understood. Results of the assessment activity will then feed into the Decision Analysis process (see section 6.8) where potential corrective action is necessary.

These activities together form the feedback loop depicted in Figure 6-26.



**Figure 6-26 Planning and Status Reporting Feedback Loop**

This loop takes place on a continual basis throughout the project life cycle. This loop is applicable at each level of the project hierarchy. Planning data, status reporting data, and assessments flow up the hierarchy with appropriate aggregation at each level; decisions cause actions to be taken down the hierarchy. Managers at each level determine (consistent with policies established at the next higher level of the project hierarchy) how often, and in what form, reporting data and assessments should be made. In establishing these status reporting and assessment requirements, some principles of good practice are:

- Use an agreed-upon set of well-defined technical measures (See Section 6.7.2.2, Technical Measures)
- Report these technical measures in a consistent format at all project levels
- Maintain historical data for both trend identification and cross-project analyses
- Encourage a logical process of rolling up technical measures, (e.g., use the WBS for status reporting and PBS for mass status reporting)
- Support assessments with quantitative risk measures
- Summarize the condition of the project by using color-coded (red, yellow, and green) alert zones for all technical measures.

Regular, periodic (e.g., monthly) tracking of the technical measures is recommended, though some measures should be tracked more often when there is rapid change or cause for concern. Key reviews, such as PDRs and CDRs, are points at which technical measures and their trends should be carefully scrutinized for early warning signs of potential problems. Should there be indications that existing trends, if allowed to continue, will yield an unfavorable outcome, corrective action should begin as soon as practical. Section 6.7.2.2 provides additional information on status reporting and assessment techniques for costs and schedules (including Earned Value Management), technical performance, and systems engineering process metrics.

The measures are predominantly assessed during the program's/project's technical reviews. Typical activities performed for technical reviews include 1) identifying, planning, and conducting phase-to-phase technical reviews, 2) establishing each review's purpose, objective, and entry and success criteria, 3) establishing the makeup of the review team, and 4) identifying and resolving action items resulting from the review. Section 6.7.2.1 summarizes the types of technical reviews typically conducted on a program/project and the role of these reviews in supporting management decision processes. It also identifies some general principles for holding reviews, but leaves explicit direction for executing a review to the program/project team to define.

The process of executing technical assessment has close relationships to other areas, such as risk management, decision analysis and technical planning. These areas may provide input into the technical assessment process or be the benefactor of outputs from the process.

### 6.7.1.3 *Outputs*

Typical outputs of the technical assessment process would include the following:

- **Assessment Results, Findings, and Recommendations:** *This is the collective data on the established measures* from which trends can be determined and variances from expected results can be understood. Results will then feed into the Decision Analysis process where potential corrective action is necessary.
- **Technical Review Reports/Minutes:** This is the collective information coming out of each review that captures the results, recommendations, and actions in regards to meeting the review's success criteria.

## 6.7.2 *Technical Assessment Guidance*

### 6.7.2.1 *Reviews, Audits, and Key Decision Points*

In order to gain a general understanding of the various technical reviews called out in Agency policy (e.g., NPR7120.5 and NPR7123.1) we need to examine the intent of the policy within each of the above mentioned documents. The NPR7120.5's primary focus is to inform the Decision Authority as to the readiness of a Program/Project to proceed into the next phase of the life cycle. This is done for each "Milestone Review" and is tied to a Key Decision Point (KDP) throughout the life cycle. These KDP/Milestone Reviews involve External Independent reviewers known as Standing Review Board (SRB) members evaluating the Program/Project and in the end, reporting their findings to the Decision Authority. In order for a Program or Project to prepare for the SRB they must conduct their own internal/peer review process. This process typically includes both informal and formal peer reviews at the subsystem and systems level. This handbook attempts to provide sufficient insight and guidance into both policy documents so that practitioners can understand how they are to be successfully integrated, however, the main focus in this handbook will be toward the "Internal" review process.

The intent and policy for reviews, audits, and key decision points should be developed during Phase A and defined in the Program/Project Plan. The specific implementation of these activities should be consistent with the types of reviews and audits described in this section, and with the NASA Program Life Cycle chart (see Figure 3-1) and the NASA Project Life Cycle chart (see Figure 3-2). However, the timing of reviews, audits, and key decision points should accommodate the need of each specific project.

### **Purpose and Definition**

The purpose of a review is to furnish the forum and process to provide NASA management and their contractors assurance that the most satisfactory approach, plan or design has been selected, that a configuration item has been produced to meet the specified requirements, or that a configuration item is ready. Reviews help to develop a better understanding among task or project participants, open communication channels, alert participants and management to problems, and open avenues for solutions. Reviews are intended to "add-value" to the project and enhance project quality and the likelihood of success. This is aided by inviting outside experts to confirm the viability of the presented approach, concept or baseline, or recommend

alternatives. Reviews may be program life-cycle reviews, project life-cycle reviews, or internal reviews.

The purpose of an audit is to provide NASA management and its contractors a thorough examination of adherence to program/project policies, plans, requirements, and specifications. Audits are the systematic examination of tangible evidence to determine adequacy, validity, and effectiveness of the activity or documentation under review. An audit may examine documentation of policies and procedures, as well as verify adherence to them.

The purpose of a Key Decision Point (KDP) is to provide a scheduled event at which the decision authority determines the readiness of a program/project to progress to the next phase of the life cycle (e.g., B to C, C to D, etc.) or to the next KDP. KDPs are part of NASA's Oversight and Approval process for Programs/Projects. For a detailed description of the process and management oversight teams please see NPR 7120.5. Essentially, KDPs serve as gates through which programs and projects must pass. Within each phase, a KDP is preceded by one or more reviews, including the governing Program Management Council (PMC) review. Allowances are made within a phase for the differences between human and robotic space flight programs and projects, but phases always end with the KDP. The potential outcomes at a KDP include:

1. Approval for continuation to the next KDP;
2. Approval for continuation to the next KDP, pending resolution of actions;
3. Disapproval for continuation to the next KDP. In such cases, follow-up actions may include a request for more information and/or a delta independent review; a request for a Termination Review (described below) for the program or the project (Phases B, C, D, E only); direction to continue in the current phase; or re-direction of the program/project.

The Decision Authority reviews materials submitted by the governing PMC, SRB, Program Manager, Project Manager and Center Management Council (CMC) in addition to agreements and program/project documentation to support the decision process. The Decision Authority makes their decision by considering a number of factors, including continued relevance to Agency strategic needs, goals, and objectives; continued cost affordability with respect to the Agency's resources; the viability and the readiness to proceed to the next phase; and remaining program or project risk (cost, schedule, technical, safety). Appeals against the final decision of the Decision Authority go to the next higher Decision Authority.

### **Project Termination**

It should be noted that project termination, while usually disappointing to project personnel, may be a proper reaction to changes in external conditions or to an improved understanding of the system's projected cost-effectiveness.

### Termination Review

A review initiated by the Decision Authority for the purpose of securing a recommendation as to whether to continue or terminate a program or project. Failing to stay within the parameters or levels specified in controlling documents will result in consideration of a termination review.

At the Termination Review, the program and the project teams present status, including any material requested by the Decision Authority. Appropriate support organizations are represented (e.g., procurement, external affairs, legislative affairs, public affairs) as needed. The decision and basis of decision are fully documented and reviewed with the NASA Associate Administrator prior to final implementation.

### General Principles for Reviews

Several factors can affect the implementation plan for any given review, such as design complexity, schedule, cost, visibility, NASA Center practices, the review itself, etc. As such, there is no set standard for conducting a review across the agency, however there are key elements, or principles, that should be included in a Review Plan. These include; definition of review scope, objectives, success criteria (consistent with NPR 7123.1), and process. Definition of the review process should include identification of schedule, including duration of the face-to-face meeting (and draft agenda), definition of roles and responsibilities of participants, identification of presentation material and data package contents, and a copy of the form to be used for Review Item Disposition/Request For Action/Comment. The review process for screening and processing discrepancies/requests/comments should also be included in the plan. The Review Plan must be agreed to by the technical team lead, project manager and for SRB-type reviews the SRB chair prior to the review.

It is recommended that all reviews consist of oral presentations of the applicable project requirements and the approaches, plans, or designs that satisfy those requirements. These presentations are normally provided by the cognizant design engineer or their immediate supervisor. It is also recommended that in addition to the SRB, the review audience should include key stakeholders, such as the science community, program executive, etc. This ensures that the project obtains buy-in from the personnel that have control over the project as well as those that benefit from a successful mission. It is also very beneficial to have project personnel in attendance that are not directly associated with the design being reviewed (e.g., EPS attending a thermal discussion). This gives the project an additional opportunity to utilize cross-discipline expertise to identify design shortfalls or recommend improvements. Of course, the audience should also include non-project specialists from safety, quality and mission assurance, reliability, verification, and testing.

#### *Program Technical Life-Cycle Reviews*

Within NASA there are various types of programs:

- Single-project programs (e.g., James Webb Space Telescope Program) tend to have long development and/or operational lifetimes, represent a large investment of Agency resources in one program/project and have contributions to that program/project from multiple organizations/agencies.

- Uncoupled programs (e.g., Discovery Program, Explorer) are implemented under a broad scientific theme and/or a common program implementation concept, such as providing frequent flight opportunities for cost-capped projects selected through Announcements of Opportunity or NASA Research Announcements. Each such project is independent of the other projects within the program.
- Loosely coupled programs (e.g., Mars Exploration Program or Lunar Precursor and Robotic Program) address specific scientific or exploration objectives through multiple space flight projects of varied scope. While each individual project has an assigned set of mission objectives, architectural and technological synergies and strategies that benefit the program as a whole are explored during the formulation process. For instance, all orbiters designed for more than one year in Mars orbit are required to carry a communication system to support present and future landers.
- Tightly coupled programs (e.g., Constellation Program) have multiple projects that execute portions of a mission or missions. No single project is capable of implementing a complete mission. Typically, multiple NASA Centers contribute to the program. Individual projects may be managed at different Centers. The program may also include other agency or international partner contributions.

Regardless of the type, all Programs are required to undergo the two technical reviews listed in Table 6-2. The main difference lies between Uncoupled/Loosely coupled Programs that tend to conduct “Status” type reviews on their projects after KDP-I and Single-project/Tightly coupled Programs that tend to follow the Project Technical Life-Cycle Review process post KDP-I.

After KDP-I Single Project/Tightly coupled Programs are responsible for conducting the System-level reviews. These reviews bring the Projects together and help to ensure the flow-down of requirements and that the overall System/Subsystems design solution satisfy the Program requirements. The Program/Program Reviews also help to resolve interface/integration issues between projects. For the sake of this handbook, Single Project Programs and Tightly Coupled Programs will follow the project life-cycle review process define after this table. Best practices and lessons learned drive Programs to conduct their “Concept and Requirement-type” reviews prior to the Projects “Concept and Requirements” reviews and the Program “Design and Acceptance-type” reviews lag the Projects.

**Table 6-2 Program Technical Reviews**

Program/System Requirements Review (P/SRR)	The P/SRR examines the functional and performance requirements defined for the program (and its constituent projects) and ensures that the requirements and the selected concept will satisfy the program and higher level requirements. It is an internal review. ROM budgets and schedules are presented.
Program/System Definition Review (P/SDR)	The P/SDR examines the proposed program architecture and the flow down to the functional elements of the system.

### *Project Technical Life-Cycle Reviews*

The phrase “project life cycle/project milestone reviews” has, over the years, come to mean different things to various Centers. Some equate it to mean the project’s controlled formal review using RIDS and pre-boards/boards. While others use it to mean the activity tied to RFAs and SRB/KDP process. This document will use the latter process to define the term. Project life-cycle reviews are mandatory reviews convened by the DA, that summarize the results of internal technical processes (peer reviews) throughout the project life cycle to NASA management and/or an independent review team, such as a Standing Review Board (see NPR 7120.5). These reviews are used to assess the progress and health of a project by providing NASA management assurance that the most satisfactory approach, plan or design has been selected, that a configuration item has been produced to meet the specified requirements, or that a configuration item is ready for launch/operation. Some examples of life-cycle reviews include System Requirements Review, Preliminary Design Review, Critical Design Review and Acceptance Review.

Typically, each life-cycle review is followed by a KDP in which the Decision Authority for the Project determines, based on results and recommendations from the life-cycle review teams, whether or not the Project can proceed to the next life-cycle phase.

### *Standing Review Boards*

The SRB’s role<sup>3</sup> is advisory to the program/project and the convening authorities, and does not have authority over any program/project content. Its review provides expert assessment of the technical and programmatic approach, risk posture, and progress against the program/project baseline. When appropriate, it may offer recommendations to improve performance and/or reduce risk.

### *Internal Reviews*

During the course of a project or task, it is necessary to conduct internal reviews that present technical approaches, trade studies, analyses, and problem areas to a peer group for evaluation and comment. The timing, participants, and content of these reviews is normally defined by the project manager or the manager of the performing organization with support from the technical team. In preparation for the life-cycle reviews a project will initiate an internal review process as defined in the Project Plan. These reviews are not just “meetings” to share ideas and resolve issues, but are internal reviews that allow the project to establish baseline requirements, plans or design through the review of technical approaches, trade studies and analyses.

Internal peer reviews provide an excellent means for controlling the technical progress of the project. They should also be used to ensure that all interested parties are involved in the development early on and throughout the process. Thus, representatives from areas such as manufacturing and quality assurance should attend the internal reviews as active participants. It is also a good practice to include representatives from other Centers/Outside providing support or developing systems/subsystems that may interface to your system/subsystem. They can then, for example, ensure that the design is producible, integratable, and that quality is managed through the project life cycle.

---

<sup>3</sup> A review board handbook will be issued by PA&E.

Since internal peer reviews will be at a much greater level of detail than the life-cycle reviews the team may utilize internal and external experts to help develop and assess approaches and concepts at the internal reviews. Some organizations form a “Red Team” to provide an internal, independent, peer-review to identify deficiencies and offer recommendations. Projects often refer to their internal reviews as “tabletop” reviews or “interim” design reviews. Whatever the name, the purpose is the same...to ensure the readiness of the baseline for successful Project life-cycle review.

It should be noted that due to the importance of these reviews each review should have well-defined entrance and success criteria established prior to the review.

### **Internal Review Boards**

As part of the Program/Project internal review process it may be necessary to establish a panel of experts from the required discipline areas (technical and management) to assist in adjudication of issues. The Program/Project Manager and Program/Project Chief Engineer work together to establish this/these group(s). It is common for the Program/Project Manager and Program/Project Chief Engineer to chair these Boards (i.e., Board and Pre-board respectively), however, the remaining membership is largely independent of the Program/Project. This approach helps to ensure a more thorough review by leveraging off both inside and outside experts. At the conclusion of this activity the project should have a clear understanding of project risk (Technical, cost and Schedule), continue to refine risk mitigation strategies and make the final determination of their readiness to proceed with the SRB/KDP process.

### **Required Technical Reviews**

This section describes the purpose, timing, objectives, success criteria, and results of the NPR 7123.1 required technical reviews in the NASA program and project life cycle. This information is intended to provide guidance to program/project managers and systems engineers, and to illustrate the progressive maturation of review activities and systems engineering products. For FS&GS projects, the NASA life-cycle phases of Formulation and Implementation divide into seven project phases. The checklists provided below aid in the preparation of specific review entry and exit criteria, but do not take their place. To minimize extra work, review material should be keyed to program/project documentation.

#### *Program/System Requirements Review*

The P/SRR is used to ensure that the program requirements are properly formulated and correlated with the Agency and mission directorate strategic objectives.

**Table 6-3 P/SRR Entrance and Success Criteria**

<b>Program/System Requirements Review</b>	
<b>Entrance Criteria</b>	<b>Success Criteria</b>
<ol style="list-style-type: none"> <li>1. A Formulation Authorization Document (FAD) has been approved.</li> <li>2. Program requirements have been defined that support mission directorate requirements on the program.</li> <li>3. Major program risks and corresponding mitigations strategies have been identified.</li> <li>4. The high-level program requirements have been documented to include:</li> <li>5. performance,</li> <li>6. safety, and</li> <li>7. programmatic requirements.</li> <li>8. An approach for verifying compliance with program requirements has been defined.</li> <li>9. Procedures for controlling changes to program requirements have been defined and approved.</li> <li>10. Traceability of program requirements to individual projects is documented in accordance with Agency needs, goals, and objectives, as described in the NASA Strategic Plan.</li> <li>11. Top program/project risks with significant technical, safety, cost, and schedule impacts are identified.</li> </ol>	<ol style="list-style-type: none"> <li>1. With respect to mission and science requirements, defined high-level program requirements are determined to be complete and are approved.</li> <li>2. Defined interfaces with other programs are approved.</li> <li>3. The program requirements are determined to provide a cost-effective program.</li> <li>4. The program requirements are adequately levied on either the single-program project or the multiple projects of the program.</li> <li>5. The plans for controlling program requirement changes have been approved.</li> <li>6. The approach for verifying compliance with program requirements has been approved.</li> <li>7. The mitigation strategies for handling identified major risks have been approved.</li> </ol>

*Program/System Definition Review*

The P/SDR applies to all NASA space flight programs to ensure the readiness of these programs to enter an approved Program Commitment Agreement (PCA). The approved PCA permits programs to transition from the program formulation phase to the program implementation phase. A Program Approval Review (PAR) is conducted as part of the P/SDR to provide Agency management with an independent assessment of the readiness of the program to proceed into implementation.

The P/SDR examines the proposed program architecture and the flow down to the functional elements of the system. The proposed program’s objectives and the concept for meeting those objectives are evaluated. Key technologies and other risks are identified and assessed. The baseline Program Plan, budgets, and schedules are presented.

The technical team provides the technical content to support the P/SDR.

**Table 6-4 P/SDR Entrance and Success Criteria**

Program/System Definition Review	
Entrance Criteria	Success Criteria
<ol style="list-style-type: none"> <li>1. A Program/System Requirements Review has been satisfactorily completed. A program plan has been prepared that includes the following:               <ol style="list-style-type: none"> <li>a. how the program will be managed;</li> <li>b. a list of specific projects;</li> <li>c. the high-level program requirements (including risk criteria); performance, safety, and programmatic requirements correlated to Agency <b>Error! Bookmark not defined.</b> and directorate strategic objectives;</li> <li>d. description of the systems to be developed (hardware and software), legacy systems, system interfaces, and facilities; and</li> <li>e. identification of major constraints affecting system development (e.g., cost, launch window, required launch vehicle, mission planetary environment, engine design, international partners, and technology drivers).</li> </ol> </li> <li>2. Program level SEMP which includes project technical approaches and management plans to implement the allocated program requirements including constituent launch, flight, and ground systems, and operations and logistics concepts.</li> <li>3. Independent Cost Analyses (ICAs) and Independent Cost Estimates (ICEs)</li> <li>4. Management plan for resources other than budget.</li> <li>5. Documentation for obtaining the program commitment agreement that includes the following:               <ol style="list-style-type: none"> <li>a. the feasibility of the program mission solution with a cost estimate within acceptable cost range,</li> <li>b. project plans adequate for project formulation initiation,</li> <li>c. identified and prioritized program concept evaluation criteria to be used in project evaluations,</li> <li>d. estimates of required annual funding levels,</li> <li>e. credible program cost and schedule allocation estimates to projects,</li> <li>f. acceptable risk and mitigation strategies (supported by a technical risk assessment),</li> <li>g. organizational structures and defined work assignments,</li> <li>h. defined program acquisition strategies,</li> <li>i. interfaces to other programs and partners,</li> <li>j. a draft plan for program implementation, and</li> <li>k. a defined program management system.</li> </ol> </li> <li>6. A draft program control plan that includes:               <ol style="list-style-type: none"> <li>a. how the program plans to control program requirements, technical design, schedule, and cost to achieve its high-level requirements;</li> <li>b. how the requirements, technical design, schedule, and cost of the program will be controlled;</li> <li>c. how the program will utilize its technical, schedule, and cost reserves to control the baseline;</li> <li>d. how the program plans to report technical, schedule, and cost status to the MDAA, including frequency and the level of detail; and</li> <li>e. how the program will address technical waivers and how dissenting opinions will be handled.</li> </ol> </li> <li>7. For each project, a top-level description has been documented.</li> </ol>	<ol style="list-style-type: none"> <li>1. An approved program plan and management approach.</li> <li>2. Approved SEMP and technical approach.</li> <li>3. Estimated costs are adequate.</li> <li>4. Documentation for obtaining the Program Commitment Agreement is approved.</li> <li>5. An approved draft program control plan.</li> <li>6. Agreement that the program is aligned with the Agency needs, goals and objectives.</li> <li>7. The technical approach is adequate.</li> <li>8. The schedule is adequate and consistent with cost, risk and mission goals.</li> <li>9. Resources other than budget are adequate and available.</li> </ol>

The P/SDR examines the proposed program architecture and the flow down to the functional elements of the system.

*Mission Concept Review*

The MCR will affirm the mission need and examine the proposed mission’s objectives and the concept for meeting those objectives. It is an internal review that usually occurs at the cognizant organization for system development.

The MCR should be completed prior to entering the concept development phase (Phase A).

**Table 6-5 MCR Entrance and Success Criteria**

Mission Concept Review	
Entrance Criteria	Success Criteria
<ol style="list-style-type: none"> <li>1. Mission goals and objectives.</li> <li>2. Analysis of Alternative Concepts to show at least one is feasible.</li> <li>3. Concept of Operations.</li> <li>4. Preliminary mission descope options</li> <li>5. Preliminary risk assessment including technologies and associated risk management/mitigation strategies and options.</li> <li>6. Conceptual test and evaluation strategy.</li> <li>7. Preliminary technical plans to achieve next phase (preliminary SEMP).</li> <li>8. Defined MOEs and MOPs.</li> <li>9. Conceptual life-cycle support strategies (logistics, manufacturing, operation, etc.).</li> </ol>	<ol style="list-style-type: none"> <li>1. Mission objectives are clearly defined and stated and are unambiguous and internally consistent.</li> <li>2. The preliminary set of requirements satisfactorily provides a system which will meet the mission objectives.</li> <li>3. The mission is feasible. A solution has been identified which is technically feasible. A rough cost estimate is within an acceptable cost range.</li> <li>4. The concept evaluation criteria to be used in candidate systems evaluation have been identified and prioritized.</li> <li>5. The need for the mission has been clearly identified.</li> <li>6. The cost and schedule estimates are credible.</li> <li>7. A technical search was done to identify existing assets or products that could satisfy the mission or parts of the mission.</li> <li>8. Technical planning is sufficient to proceed to the next phase.</li> <li>9. Risk and mitigation strategies have been identified and are acceptable.</li> </ol>

**Objectives.** The objectives of the review are to:

- a. Ensure a through review of the products supporting the review.
- b. Ensure the products meet the entrance criteria and success criteria
- c. Ensure issues raised during the review are appropriately documented and a plan for resolution is prepared

**Results of Review.** A successful MCR supports the determination that the proposed mission meets the customer need, and has sufficient quality and merit to support a field center management decision to propose further study to the cognizant NASA Program Associate Administrator (PAA) as a candidate Phase A effort.

*System Requirements Review*

- The SRR examines the functional and performance requirements defined for the system and the preliminary program or project plan and ensures that the requirements and the selected concept will satisfy the mission.
- The SRR is conducted during the concept development phase (Phase A) and before conducting the SDR or MDR.

**Table 6-6 SRR Entrance and Success Criteria**

<b>System Requirements Review</b>	
<b>Entrance Criteria</b>	<b>Success Criteria</b>
<ol style="list-style-type: none"> <li>1. Successful completion of the MCR and responses made to all MCR Requests for Actions (RFAs) and Review Item Discrepancies (RIDs).</li> <li>2. A preliminary SRR agenda, success criteria, and charge to the review board have been agreed to by the technical team, project manager, and review chair prior to the SRR.</li> <li>3. The following technical products for hardware and software system elements are available to the cognizant participants prior to the review:               <ol style="list-style-type: none"> <li>a. system requirements document;</li> <li>b. system software functionality description;</li> <li>c. updated concept of operations;</li> <li>d. updated mission requirements, if applicable;</li> <li>e. baselined SEMP;</li> <li>f. risk management plan;</li> <li>g. preliminary system requirements allocation to the next lower level system;</li> <li>h. updated cost estimate;</li> <li>i. Technology Development Maturity Assessment Plan;</li> <li>j. updated risk assessment and mitigations (including PRA, as applicable);</li> <li>k. logistics documentation (e.g., preliminary maintenance plan);</li> <li>l. preliminary human rating plan, if applicable;</li> <li>m. software Development Plan (SDP);</li> <li>n. system safety and mission assurance plan;</li> <li>o. configuration management plan;</li> <li>p. initial document tree;</li> <li>q. verification and validation approach;</li> <li>r. preliminary system safety analysis; and</li> <li>s. other specialty disciplines, as required.</li> </ol> </li> </ol>	<ol style="list-style-type: none"> <li>1. The project utilizes a sound process for the allocation and control of requirements throughout all levels, and a plan has been defined to complete the definition activity within schedule constraints.</li> <li>2. Requirements definition is complete with respect to top-level mission and science requirements, and interfaces with external entities and between major internal elements have been defined.</li> <li>3. Requirements allocation and flow down of key driving requirements have been defined down to subsystems.</li> <li>4. Preliminary approaches have been determined for how requirements will be verified and validated down to the subsystem level.</li> <li>5. Major risks have been identified and technically assessed, and viable mitigation strategies have been defined.</li> </ol>

**Objectives.** The objectives of the review are to:

- a. Ensure a thorough review of the products supporting the review.
- b. Ensure the products meet the entrance criteria and success criteria
- c. Ensure issues raised during the review are appropriately documented and a plan for resolution is prepared

**Results of Review.** Successful completion of the SRR freezes program/project requirements and leads to a formal decision by the cognizant Program Associate Administrator (PAA) to proceed with proposal request preparations for project implementation.

*Mission Definition Review (Robotic Missions Only)*

- The MDR examine the functional and performance requirements defined for the system and the preliminary program or project plan and ensures that the requirements and the selected concept will satisfy the mission.
- MDR is conducted during the concept development phase (Phase A) following completion of the concept studies phase (Pre-Phase A) and before the preliminary design phase (Phase B).

**Table 6-7 MDR Entrance and Success Criteria**

<b>Mission Definition Review</b>	
<b>Entrance Criteria</b>	<b>Success Criteria</b>
<ol style="list-style-type: none"> <li>1. Successful completion of the MCR and responses made to all MCR Requests for Actions (RFAs) and Review Item Discrepancies (RIDs).</li> <li>2. A preliminary MDR agenda, success criteria, and charge to the board have been agreed to by the technical team, project manager, and review chair prior to the MDR.</li> <li>3. The following technical products for hardware and software system elements are available to the cognizant participants prior to the review:               <ol style="list-style-type: none"> <li>a. system architecture;</li> <li>b. updated system requirements document, if applicable;</li> <li>c. system software functionality description;</li> <li>d. updated concept of operations, if applicable;</li> <li>e. updated mission requirements, if applicable;</li> <li>f. updated SEMP, if applicable;</li> <li>g. updated risk management plan, if applicable;</li> <li>h. Technology Development Maturity Assessment Plan;</li> <li>i. preferred system solution definition including major trades and options;</li> <li>j. updated risk assessment and mitigations (including PRA, as applicable);</li> <li>k. updated cost and schedule data;</li> <li>l. logistics documentation (e.g., preliminary maintenance plan);</li> <li>m. Software Development Plan (SDP);</li> <li>n. system safety and mission assurance plan;</li> <li>o. configuration management plan;</li> <li>p. updated initial document tree, if applicable;</li> <li>q. preliminary system safety analysis; and</li> <li>r. other specialty disciplines as required.</li> </ol> </li> </ol>	<ol style="list-style-type: none"> <li>1. The resulting overall concept is reasonable, feasible, complete, responsive to the mission requirements, and is consistent with system requirements and available resources (cost, schedule, and mass power).</li> <li>2. System and subsystem design approaches and operational concepts exist and are consistent with the requirements set.</li> <li>3. The requirements, design approaches, and conceptual design will fulfill the mission needs within the estimated costs.</li> <li>4. Major risks have been identified and technically assessed, and viable mitigation strategies have been defined.</li> </ol>

**Objectives.** The objectives of the review are to:

- a. Ensure a thorough review of the products supporting the review.
- b. Ensure the products meet the entrance criteria and success criteria
- c. Ensure issues raised during the review are appropriately documented and a plan for resolution is prepared

**Results of Review.** A successful MDR supports the decision to further develop the system architecture/design and any technology needed to accomplish the mission. The results reinforce the mission’s merit and provide a basis for the system acquisition strategy.

*System Definition Review (Human Space Flight Missions Only)*

- The SDR examines the proposed system architecture/design and the flow down to all functional elements of the system.
- SDR is conducted at the end of the concept development phase (Phase A) and before the preliminary design phase (Phase B) begins.

**Table 6-8 SDR Entrance and Success Criteria**

<b>System Definition Review</b>	
<b>Entrance Criteria</b>	<b>Success Criteria</b>
<ol style="list-style-type: none"> <li>1. Successful completion of the SRR/MDR and responses has been made to all SRR/MDR RFAs and RIDs.</li> <li>2. A preliminary SDR agenda, success criteria, and charge to the board have been agreed to by the technical team, project manager, and review chair prior to the SDR.</li> <li>3. SDR technical products listed below for both hardware and software system elements have been made available to the cognizant participants prior to the review:               <ol style="list-style-type: none"> <li>a. system architecture;</li> <li>b. preferred system solution definition including major trades and options;</li> <li>c. updated baselined documentation, as required;</li> <li>d. preliminary functional baseline (with supporting trade-off analyses and data);</li> <li>e. preliminary system software functional requirements;</li> <li>f. SEMP changes, if any;</li> <li>g. updated risk management plan;</li> <li>h. updated risk assessment and mitigations (including PRA, as applicable);</li> <li>i. updated technology development, maturity, and assessment plan;</li> <li>j. updated cost and schedule data;</li> <li>k. updated logistics documentation;</li> <li>l. based on system complexity, updated human rating plan;</li> <li>m. software test plan;</li> <li>n. software requirements document(s);</li> <li>o. interface requirements documents (including software);</li> <li>p. technical resource utilization estimates and margins;</li> <li>q. updated safety and mission assurance (S&amp;MA) plan; and</li> <li>r. updated PHA.</li> </ol> </li> </ol>	<ol style="list-style-type: none"> <li>1. Systems requirements, including mission success criteria and any sponsor-imposed constraints, are defined and form the basis for the proposed conceptual design.</li> <li>2. All technical requirements are allocated, and the flow down to subsystems is adequate. The requirements, design approaches, and conceptual design will fulfill the mission needs consistent with the available resources (cost, schedule, mass, and power).</li> <li>3. The requirements process is sound and can reasonably be expected to continue to identify and flow detailed requirements in a manner timely for development.</li> <li>4. The technical approach is credible and responsive to the identified requirements.</li> <li>5. Technical plans have been updated, as necessary.</li> <li>6. The trade-offs are completed, and those planned for Phase B adequately address the option space.</li> <li>7. Significant development, mission, and safety risks are identified and technically assessed, and a risk process and resources exist to manage the risks.</li> <li>8. Adequate planning exists for the development of any enabling new technology.</li> <li>9. The ConOps is consistent with proposed design concept(s) and is in alignment with the mission requirements.</li> </ol>

**Objectives.** The objectives of the review are to:

- a. Ensure a through review of the products supporting the review.

- b. Ensure the products meet the entrance criteria and success criteria
- c. Ensure issues raised during the review are appropriately documented and a plan for resolution is prepared

**Results of Review.** As a result of successful completion of the SDR, the system and its operation are well enough understood to warrant design and acquisition of the end items. Approved specifications for the system, its segments, and preliminary specifications for the design of appropriate functional elements may be released. A configuration management plan is established to control design and requirement changes. Plans to control and integrate the expanded technical process are in place.

*Preliminary Design Review*

- The Preliminary Design Review (PDR) demonstrates that the preliminary design meets all system requirements with acceptable risk and within the cost and schedule constraints and establishes the basis for proceeding with detailed design. It will show that the correct design option has been selected, interfaces have been identified, and verification methods have been described.
- PDR occurs near the completion of the preliminary design phase (Phase B) as the last review in the Formulation Phase.

**Table 6-9 PDR Entrance and Success Criteria**

Preliminary Design Review	
Entrance Criteria	Success Criteria
<ol style="list-style-type: none"> <li>1. Successful completion of the SDR and responses has been made to all SDR RFAs, or a timely closure plan exists for those remaining open.</li> <li>2. A preliminary PDR agenda, success criteria, and charge to the board have been agreed to by the technical team, project manager, and review chair prior to the PDR.</li> <li>3. PDR technical products listed below for both hardware and software system elements have been made available to the cognizant participants prior to the review:               <ol style="list-style-type: none"> <li>a. Updated baselined documentation, as required.</li> <li>b. Preliminary subsystem design specifications for each configuration item (hardware and software), with supporting tradeoff analyses and data, as required. The preliminary software design specification needs to include a completed definition of the software architecture and a preliminary database design description as applicable.</li> <li>c. Updated technology development maturity assessment plan.</li> <li>d. Updated risk assessment and mitigation.</li> <li>e. Updated cost and schedule data.</li> <li>f. Updated logistics documentation, as required.</li> <li>g. Applicable technical plans (e.g., technical performance measurement plan, contamination</li> </ol> </li> </ol>	<ol style="list-style-type: none"> <li>1. Agreement exists for the top-level requirements, including mission success criteria, TPMs, and any sponsor-imposed constraints, and that these are finalized, stated clearly, and are consistent with the preliminary design.</li> <li>2. The flow down of verifiable requirements is complete and proper or, if not, an adequate plan exists for timely resolution of open items. Requirements are traceable to mission goals and objectives.</li> <li>3. The preliminary design is expected to meet the requirements at an acceptable level of risk.</li> <li>4. Definition of the technical interfaces is consistent with the overall technical maturity and proves an acceptable level of risk.</li> <li>5. Adequate technical interfaces are consistent with the overall technical maturity and provide an acceptable level of risk.</li> <li>6. Adequate technical margins exist with respect to technical performances measures (TPMs).</li> <li>7. Any required new technology has been developed to an adequate state of readiness, or back-up options exist and are supported to make them a viable alternative.</li> <li>8. The project risks are understood, and plans and a process and resources exist to effectively</li> </ol>

Preliminary Design Review	
Entrance Criteria	Success Criteria
control plan, parts management plan, environments control plan, EMI/EMC control plan, payload-to-carrier integration plan, producibility/manufacturability program plan, reliability program plan, quality assurance plan, etc.). h. Applicable standards. i. Safety analyses and plans. j. Engineering drawing tree. k. Interface control documents. l. Verification/validation plan. m. Plans to respond to regulatory requirements (e.g., Environmental Impact Statement), as required. n. Disposal plan. o. Technical resource utilization estimates and margins. p. System-level hazard analysis. q. Preliminary limited life items list (LLIL).	manage them. 9. Safety and mission assurance (i.e., safety, reliability, maintainability, quality, and EEE parts) have been adequately addressed in preliminary designs and any applicable S&MA products (i.e., hazard analysis and failure modes and effects analysis) have been approved. 10. The operational concept is technically sound, that it includes (where appropriate) human factors that apply, and that requirements for its execution flow down. 11. The Life-Cycle Cost Estimate (LCCE) is approved. For projects requiring Earned Value Management the Performance Measurement Baseline Management Reserve, and Schedule Reserve are approved.

**Objectives.** The objectives of the review are to:

- a. Ensure a thorough review of the products supporting the review.
- b. Ensure the products meet the entrance criteria and success criteria
- c. Ensure issues raised during the review are appropriately documented and a plan for resolution is prepared

**Results of Review.** As a result of successful completion of the PDR, the “design-to” baseline is approved. It also authorizes the project to proceed into implementation and toward final design.

*Critical Design Review*

- The purpose of the CDR is to demonstrate that the maturity of the design is appropriate to support proceeding with full scale fabrication, assembly, integration, and test, and that the technical effort is on track to complete the flight and ground system development and mission operations in order to meet mission performance requirements within the identified cost and schedule constraints.
- CDR occurs during the final design phase (Phase C).

**Table 6-10 CDR Entrance and Success Criteria**

<b>Critical Design Review</b>	
<b>Entrance Criteria</b>	<b>Success Criteria</b>
<ol style="list-style-type: none"> <li>1. Successful completion of the PDR and responses has been made to all PDR RFAs and RIDs, or a timely closure plan exists for those remaining open.</li> <li>2. A preliminary CDR agenda, success criteria, and charge to the board have been agreed to by the technical team, project manager, and review chair prior to the CDR.</li> <li>3. CDR technical work products listed below for both hardware and software system elements have been made available to the cognizant participants prior to the review:               <ol style="list-style-type: none"> <li>a. updated baselined documents, as required;</li> <li>b. product build-to specifications for each hardware and software configuration item, along with supporting trade-off analyses and data;</li> <li>c. fabrication, assembly, integration, and test plans and procedures;</li> <li>d. technical data package (e.g., integrated schematics, spares provisioning list, interface control documents, engineering analyses, and specifications);</li> <li>e. operational limits and constraints;</li> <li>f. technical resource utilization estimates and margins;</li> <li>g. acceptance criteria;</li> <li>h. command and telemetry list;</li> <li>i. verification plan (including requirements and specification);</li> <li>j. validation plan;</li> <li>k. Launch Site Operations Plan;</li> <li>l. checkout and activation plan;</li> <li>m. disposal plan (including decommissioning or termination);</li> <li>n. updated Technology Development Maturity Assessment Plan;</li> <li>o. updated risk assessment and mitigation;</li> <li>p. update reliability analyses and assessments;</li> <li>q. updated cost and schedule data;</li> <li>r. updated logistics documentation;</li> <li>s. software design document(s) (including interface design documents);</li> <li>t. updated LLIL;</li> <li>u. subsystem-level and preliminary operations hazards analyses;</li> <li>v. systems and subsystem certification plans and requirements (as needed); and</li> <li>w. system safety analysis with associated verifications.</li> </ol> </li> </ol>	<ol style="list-style-type: none"> <li>1. The detailed design is expected to meet the requirements with adequate margins at an acceptable level of risk.</li> <li>2. Interface control documents are appropriately matured to proceed with fabrication, assembly, integration, and test, and plans are in place to manage any open items.</li> <li>3. High confidence exists in the product baseline, and adequate documentation exists or will exist in a timely manner to allow proceeding with fabrication, assembly, integration, and test.</li> <li>4. The product verification and product validation requirements and plans are complete.</li> <li>5. The testing approach is comprehensive, and the planning for system assembly, integration, test, and launch site and mission operations is sufficient to progress into the next phase.</li> <li>6. Adequate technical and programmatic margins and resources exist to complete the development within budget, schedule, and risk constraints.</li> <li>7. Risks to mission success are understood and credibly assessed, and plans and resources exist to effectively manage them.</li> <li>8. Safety and mission assurance (i.e., safety, reliability, maintainability, quality, and EEE parts) have been adequately addressed in system and operational designs, and any applicable S&amp;MA products (i.e., PRA, system safety analysis, and failure modes and effects analysis) have been approved.</li> </ol>

**Objectives.** The objectives of the review are to:

- a. Ensure a thorough review of the products supporting the review.
- b. Ensure the products meet the entrance criteria and success criteria
- c. Ensure issues raised during the review are appropriately documented and a plan for resolution is prepared

**Results of Review.** As a result of successful completion of the CDR, the “build-to” baseline, production, and verification plans are approved. Approved drawings are released and authorized for fabrication. It also authorizes coding of deliverable software (according to the “build-to” baseline and coding standards presented in the review), and system qualification testing and integration. All open issues should be resolved with closure actions and schedules.

*Production Readiness Review*

- A PRR is held for FS&GS projects developing or acquiring multiple or similar systems greater than three or as determined by the project.
- The PRR determines the readiness of the system developers to efficiently produce the required number of systems.
- It ensures that the production plans; fabrication, assembly, and integration enabling products; and personnel are in place and ready to begin production.
- PRR occurs during the final design phase (Phase C).

**Table 6-11 PRR Entrance and Success Criteria**

Production Readiness Review	
Entrance Criteria	Success Criteria
<ol style="list-style-type: none"> <li>1. The significant production engineering problems encountered during development are resolved.</li> <li>2. The design documentation is adequate to support production.</li> <li>3. The production plans and preparation are adequate to begin fabrication.</li> <li>4. The production-enabling products and adequate resources are available, have been allocated, and are ready to support end product production.</li> </ol>	<ol style="list-style-type: none"> <li>1. The design is appropriately certified.</li> <li>2. The system requirements are fully met in the final production configuration.</li> <li>3. Adequate measures are in place to support production.</li> <li>4. Design-for-manufacturing considerations ensure ease and efficiency of production and assembly.</li> <li>5. Risks have been identified, credibly assessed, and characterized, and mitigation efforts have been defined.</li> <li>6. The bill of materials has been reviewed and critical parts identified.</li> <li>7. Delivery schedules have been verified.</li> <li>8. Alternate sources for resources have been identified, as appropriate.</li> <li>9. Adequate spares have been planned and budgeted.</li> <li>10. Required facilities and tools are sufficient for end product production.</li> <li>11. Specified special tools and test equipment are available in proper quantities.</li> <li>12. Production and support staff are qualified.</li> <li>13. Drawings are certified.</li> <li>14. Production engineering and planning are sufficiently mature for cost-effective production.</li> <li>15. Production processes and methods are consistent with quality requirements and compliant with occupational safety, environmental, and energy conservation regulations.</li> <li>16. Qualified suppliers are available for materials that are to be procured.</li> </ol>

**Objectives.** The objectives of the review are to:

- a. Ensure a through review of the products supporting the review.
- b. Ensure the products meet the entrance criteria and success criteria
- c. Ensure issues raised during the review are appropriately documented and a plan for resolution is prepared

**Results of Review.** As a result of successful completion of the PRR, the final production “build-to” baseline, production, and verification plans are approved. Approved drawings are released and authorized for production. It also authorizes coding of deliverable software (according to the “build-to” baseline and coding standards presented in the review), and system qualification testing and integration. All open issues should be resolved with closure actions and schedules.

*System Integration Review*

- An SIR ensures that the system is ready to be integrated.
- Segments, components, and subsystems are available and ready to be integrated into the system.
- Integration facilities, support personnel, and integration plans and procedures are ready for integration.
- SIR is conducted at the end of the final design phase (Phase C) and before the systems assembly, integration and test phase (Phase D) begins.

**Table 6-12 SIR Entrance and Success Criteria**

System Integration Review	
Entrance Criteria	Success Criteria
<ol style="list-style-type: none"> <li>1. Integration plans and procedures have been completed and approved.</li> <li>2. Segments and/or components are available for integration.</li> <li>3. Mechanical and electrical interfaces have been verified against the interface control documentation.</li> <li>4. All applicable functional, unit level, subsystem, and qualification testing has been conducted successfully.</li> <li>5. Integration facilities, including clean rooms, ground support equipment, handling fixtures, overhead cranes and electrical test equipment, are ready and available.</li> <li>6. Support personnel have been adequately trained.</li> <li>7. Handling and safety requirements have been documented.</li> <li>8. All known system discrepancies have been identified</li> </ol>	<ol style="list-style-type: none"> <li>1. Adequate integration plans and procedures are completed and approved for the system to be integrated.</li> <li>2. Previous component, subsystem, and system test results form a satisfactory basis for proceeding to integration.</li> <li>3. Risk level is identified and accepted by program/project leadership, as required.</li> <li>4. The integration procedures and work flow have been clearly defined and documented.</li> <li>5. The review of the integration plans, as well as the procedures, environment, and the configuration of the items to be integrated, provides a reasonable expectation that the integration will</li> </ol>

<p>and dispositioned in accordance with an agreed-upon plan.</p> <p>9. All previous design review success criteria and key issues have been satisfied in accordance with an agreed-upon plan.</p> <p>10. The quality control organization is ready to support the integration effort.</p>	<p>proceed successfully.</p> <p>6. Integration personnel have received appropriate training in the integration and safety procedures.</p>
---	---

**Objectives.** The objectives of the review are to:

- a. Ensure a through review of the products supporting the review.
- b. Ensure the products meet the entrance criteria and success criteria
- c. Ensure issues raised during the review are appropriately documented and a plan for resolution is prepared

**Results of Review.** As a result of successful completion of the SIR, the final as-built baseline, and verification plans are approved. Approved drawings are released and authorized to support integration. All open issues should be resolved with closure actions and schedules. The subsystems/systems integration procedures, GSE, facilities, logistical needs and support personnel are planned for and are ready to support integration.

*Test Readiness Review*

- A TRR ensures that the test article (hardware/software), test facility, support personnel, and test procedures are ready for testing and data acquisition, reduction, and control.
- A TRR is held prior to commencement of verification or validation testing.

**Table 6-13 TRR Entrance and Success Criteria**

Test Readiness Review	
Entrance Criteria	Success Criteria
<ol style="list-style-type: none"> <li>1. The objectives of the testing have been clearly defined and documented and that all of the test plans, procedures, environment, and the configuration of the test item support those objectives.</li> <li>2. Configuration of system under test has been defined and agreed to. All interfaces have been placed under configuration management or have been defined in accordance with an agreed to plan, and a version description document has been made available to TRR participants prior to the review.</li> <li>3. All applicable functional, unit level, subsystem, system, and qualification testing has been conducted successfully.</li> <li>4. All TRR specific materials such as test plans, test</li> </ol>	<ol style="list-style-type: none"> <li>1. Adequate test plans are completed and approved for the system under test.</li> <li>2. Adequate identification and coordination of required test resources is completed</li> <li>3. Previous component, subsystem, system test results form a satisfactory basis for proceeding into planned tests.</li> <li>4. Risk level is identified and accepted by program/competency leadership as required.</li> <li>5. Plan to capture any lessons learned from the test program</li> </ol>

<p>cases, and procedures have been available to all participants prior to conducting the review.</p> <ol style="list-style-type: none"> <li>5. All known system discrepancies have been identified and dispositioned in accordance with an agreed upon plan.</li> <li>6. All previous design review success criteria and key issues have been satisfied in accordance with an agreed upon plan.</li> <li>7. All required test resources (people (including a designated test director) facilities, test articles, test instrumentation) have been identified and are available to support required tests.</li> <li>8. Roles and responsibilities of all test participants are defined and agreed to.</li> <li>9. Test contingency planning has been accomplished, and all personnel have been trained.</li> </ol>	<ol style="list-style-type: none"> <li>6. The objectives of the testing have been clearly defined and documented, and the review of all the test plans, as well as the procedures, environment, and the configuration of the test item, provide a reasonable expectation that the objectives will be met</li> <li>7. The test cases have been reviewed and analyzed for expected results, and the results are consistent with the test plans and objectives</li> <li>8. Test personnel have received appropriate training in test operation and safety procedures.</li> </ol>
---	---

**Objectives.** The objectives of the review are to:

- a. Ensure a through review of the products supporting the review.
- b. Ensure the products meet the entrance criteria and success criteria
- c. Ensure issues raised during the review are appropriately documented and a plan for resolution is prepared

**Results of Review.** A successful TRR signifies that test and safety engineers have certified that preparations are complete, and that the project manager has authorized formal test initiation.

*System Acceptance Review*

- The purpose of the SAR is to verify the completeness of the specific end item with respect to the expected maturity level and to assess compliance to stakeholder expectations. The SAR examines the system, its end items and documentation, and test data and analyses that support verification and validation. It also ensures that the system has sufficient technical maturity to authorize its shipment to the designated operational facility or launch site.

**Table 6-14 SAR Entrance and Success Criteria**

<b>System Acceptance Review</b>	
<b>Entrance Criteria</b>	<b>Success Criteria</b>
<ol style="list-style-type: none"> <li>1. A preliminary agenda has been coordinated (nominally) prior to the SAR.</li> <li>2. The following SAR technical products have been made available to the cognizant participants prior to the review:               <ol style="list-style-type: none"> <li>a. results of the SARs conducted at the major suppliers;</li> <li>b. transition to production and/or manufacturing plan;</li> <li>c. product verification results;</li> <li>d. product validation results;</li> <li>e. documentation that the delivered system complies with the established acceptance criteria;</li> <li>f. documentation that the system will perform properly in the expected operational environment;</li> <li>g. technical data package updated to include all test results;</li> <li>h. certification package;</li> <li>i. updated risk assessment and mitigation;</li> <li>j. successfully completed previous milestone reviews; and</li> <li>k. remaining liens or unclosed actions and plans for closure.</li> </ol> </li> </ol>	<ol style="list-style-type: none"> <li>1. Required tests and analyses are complete and indicate that the system will perform properly in the expected operational environment.</li> <li>2. Risks are known and manageable.</li> <li>3. System meets the established acceptance criteria.</li> <li>4. Required safe shipping, handling, checkout, and operational plans and procedures are complete and ready for use.</li> <li>5. Technical data package is complete and reflects the delivered system.</li> <li>6. All applicable lessons learned for organizational improvement and system operations are captured.</li> </ol>

**Objectives.** The objectives of the review are to:

- a. Ensure a thorough review of the products supporting the review.
- b. Ensure the products meet the entrance criteria and success criteria
- c. Ensure issues raised during the review are appropriately documented and a plan for resolution is prepared

**Results of Review.** As a result of successful completion of the SAR, the system is accepted by the buyer, and authorization is given to ship the hardware to the launch site or operational facility, and to install software and hardware for operational use.

*Operational Readiness Review*

- The ORR examines the actual system characteristics and the procedures used in the system or product’s operation and ensures that all system and support (flight and ground) hardware, software, personnel, procedures, and user documentation accurately reflects the deployed state of the system.

**Table 6-15 ORR Entrance and Success Criteria**

<b>Operational Readiness Review</b>	
<b>Entrance Criteria</b>	<b>Success Criteria</b>
<ol style="list-style-type: none"> <li>1. All validation testing has been completed.</li> <li>2. Test failures and anomalies from validation testing have been resolved and the results incorporated into all supporting and enabling operational products.</li> <li>3. All operational supporting and enabling products (e.g., facilities, equipment, documents, updated databases) that are necessary for the nominal and contingency operations have been tested and delivered/installed at the site(s) necessary to support operations.</li> <li>4. Approved operations handbook.</li> <li>5. Training has been provided to the users and operators on the correct operational procedures for the system.</li> <li>6. Operational contingency planning has been accomplished, and all personnel have been trained.</li> </ol>	<ol style="list-style-type: none"> <li>1. The system, including any enabling products, is determined to be ready to be placed in an operational status.</li> <li>2. All applicable lessons learned for organizational improvement and systems operations have been captured.</li> <li>3. All waivers and anomalies have been closed.</li> <li>4. Systems hardware, software, personnel, and procedures are in place to support operations.</li> </ol>

**Objectives.** The objectives of the review are to:

- a. Ensure a thorough review of the products supporting the review.
- b. Ensure the products meet the entrance criteria and success criteria
- c. Ensure issues raised during the review are appropriately documented and a plan for resolution is prepared

**Results of Review.** As a result of successful ORR completion, the system is ready to assume normal operations.

*Flight Readiness Review*

- The FRR examines tests, demonstrations, analyses, and audits that determine the system’s readiness for a safe and successful flight/launch and for subsequent flight operations. It also ensures that all flight and ground hardware, software, personnel, and procedures are operationally ready.

**Table 6-16 FRR Entrance and Success Criteria**

<b>Flight Readiness Review</b>	
<b>Entrance Criteria</b>	<b>Success Criteria</b>
<ol style="list-style-type: none"> <li>1. Receive certification that flight operations can safely proceed with acceptable risk.</li> <li>2. Confirm that the system and support elements are properly configured and ready for flight.</li> <li>3. Establish that all interfaces are compatible and function as expected.</li> <li>4. Establish that the system state supports a launch “go” decision based on go/no-go criteria.</li> <li>5. Flight failures and anomalies from previously completed flights and reviews have been resolved and the results incorporated into all supporting and enabling operational products.</li> <li>6. The system has been configured for flight.</li> </ol>	<ol style="list-style-type: none"> <li>1. The flight vehicle is ready for flight.</li> <li>2. The hardware is deemed acceptably safe for flight (i.e., meeting the established acceptable risk criteria or documented as being accepted by the PM).</li> <li>3. Flight and ground software elements are ready to support flight and flight operations.</li> <li>4. Interfaces are checked out and found to be functional.</li> <li>5. Open items and waivers have been examined and found to be acceptable.</li> <li>6. The flight and recovery environmental factors are within constraints.</li> <li>7. All open safety and mission risk items have been addressed.</li> </ol>

**Objectives.** The objectives of the review are to:

- a. Ensure a thorough review of the products supporting the review.
- b. Ensure the products meet the entrance criteria and success criteria
- c. Ensure issues raised during the review are appropriately documented and a plan for resolution is prepared

**Results of Review.** As a result of successful FRR completion, technical and procedural maturity exists for system launch and flight authorization. and in some cases initiation of system operations.

*Post Launch Assessment Review*

- A PLAR is a post-deployment evaluation of the readiness of the spacecraft systems to proceed with full-up, routine operations.
- The review evaluates the status, performance, and capabilities of the project evident from the flight operations experience since launch. This can also mean assessing readiness to transfer responsibility from the development organization to the operations organization.
- The review also evaluates the status of the project plans and the capability to conduct the mission with emphasis on near-term operations and mission-critical events. This review is typically held after the early flight operations and initial checkout.

**Table 6-17 PLAR Entrance and Success Criteria**

<b>Post-Launch Assessment Review</b>	
<b>Entrance Criteria</b>	<b>Success Criteria</b>
<ol style="list-style-type: none"> <li>1. The launch and early operations performance, including (when appropriate) the early propulsive maneuver results, are available.</li> <li>2. The observed spacecraft and science instrument performance, including instrument calibration plans and status, are available.</li> <li>3. The launch vehicle performance assessment and mission implications, including launch sequence assessment and launch operations experience with lessons learned, are completed.</li> <li>4. The mission operations and ground data system experience, including tracking and data acquisition support and spacecraft telemetry data analysis, are available.</li> <li>5. The mission operations organization, including status of staffing, facilities, tools, and mission software (e.g., spacecraft analysis, and sequencing), are available.</li> <li>6. In-flight anomalies and the response taken, including any autonomous fault protection actions taken by the spacecraft, or any unexplained spacecraft telemetry, including alarms, are documented.</li> <li>7. The need for significant changes to procedures, interface agreements, software, and staffing has been documented.</li> <li>8. Documentation is updated, including any updates originating from the early operations experience.</li> <li>9. Future development/test plans are developed.</li> </ol>	<ol style="list-style-type: none"> <li>1. The observed spacecraft and science payload performance agrees with prediction, or if not, it is adequately understood so that future behavior can be predicted with confidence.</li> <li>2. All anomalies have been adequately documented, and their impact on operations assessed. Further, anomalies impacting spacecraft health and safety or critical flight operations have been properly dispositioned.</li> <li>3. The mission operations capabilities, including staffing and plans, are adequate to accommodate the actual flight performance.</li> <li>4. Liens, if any, on operations, identified as part of the ORR, have been satisfactorily dispositioned.</li> </ol>

**Objectives.** The objectives of the review are to:

- a. Ensure a thorough review of the products supporting the review.
- b. Ensure the products meet the entrance criteria and success criteria
- c. Ensure issues raised during the review are appropriately documented and a plan for resolution is prepared

*Critical Event Readiness Review*

- A CERR confirms the project’s readiness to execute the mission’s critical activities during flight operation.

**Table 6-18 CERR Entrance and Success Criteria**

<b>Critical Event Readiness Review</b>	
<b>Entrance Criteria</b>	<b>Success Criteria</b>
<ol style="list-style-type: none"> <li>1. Mission overview and context for the critical event(s).</li> <li>2. Activity requirements and constraints.</li> <li>3. Critical activity sequence design description including key trade-offs and rationale for selected approach.</li> <li>4. Fault protection strategy.</li> <li>5. Critical activity operations plan including planned uplinks and criticality.</li> <li>6. Sequence verification (testing, walk-throughs, peer review) and critical activity validation.</li> <li>7. Operations team training plan and readiness report.</li> <li>8. Risk areas and mitigations.</li> <li>9. Spacecraft readiness report.</li> <li>10. Open items and plans.</li> </ol>	<ol style="list-style-type: none"> <li>1. The critical activity design is compliant with the requirements on it.</li> <li>2. The preparation for the critical activity, including the verification and validation, is thorough.</li> <li>3. The project (including all the systems, supporting services, and documentation) is ready to support the activity.</li> <li>4. The requirements for the successful execution of the critical event(s) are complete and understandable and have been flowed down to the appropriate levels for implementation.</li> </ol>

**Objectives.** The objectives of the review are to:

- a. Ensure a through review of the products supporting the review.
- b. Ensure the products meet the entrance criteria and success criteria
- c. Ensure issues raised during the review are appropriately documented and a plan for resolution is prepared

*Post-Flight Assessment Review*

- The PFAR evaluates the activities from the flight after recovery.
- The review identifies all anomalies that occurred during the flight and mission and determines the actions necessary to mitigate or resolve the anomalies for future flights.

**Table 6-19 PFAR Entrance and Success Criteria**

<b>Post-Flight Assessment Review</b>	
<b>Entrance Criteria</b>	<b>Success Criteria</b>
<ol style="list-style-type: none"> <li>1. All anomalies that occurred during the mission, as well as during preflight testing, countdown, and ascent, identified.</li> <li>2. Report on overall post-recovery condition.</li> <li>3. Report any evidence of ascent debris.</li> <li>4. All photo and video documentation available.</li> <li>5. Retention plans for scrapped hardware completed.</li> <li>6. Post-Flight Assessment Team Operating Plan completed.</li> <li>7. Disassembly activities planned and scheduled.</li> <li>8. Processes and controls to coordinate in-flight anomaly</li> </ol>	<ol style="list-style-type: none"> <li>1. Formal final report documenting flight performance and recommendations for future missions.</li> <li>2. All anomalies have been adequately documented and dispositioned.</li> <li>3. The impact of anomalies on future flight operations has been assessed.</li> <li>4. Plans for retaining assessment documentation and imaging have</li> </ol>

<p>trouble shooting and post-flight data preservation developed.</p> <p>9. Problem reports, corrective action requests, Post Flight Anomaly Records (PFARs), and final post-flight documentation completed.</p> <p>10. All post-flight hardware and flight data evaluation reports completed.</p>	<p>been made.</p> <p>5. Reports and other documentation have been added to a database for performance comparison and trending.</p>
---	--

**Objectives.** The objectives of the review are to:

- a. Ensure a through review of the products supporting the review.
- b. Ensure the products meet the entrance criteria and success criteria
- c. Ensure issues raised during the review are appropriately documented and a plan for resolution is prepared

*Decommissioning Review*

- The purpose of the DR is to confirm the decision to terminate or decommission the system and assess the readiness for the safe decommissioning and disposal of system assets.
- The DR is normally held near the end of routine mission operations upon accomplishment of planned mission objectives. It may be advanced if some unplanned event gives rise to a need to prematurely terminate the mission, or delayed if operational life is extended to permit additional investigations.

**Table 6-20 DR Entrance and Success Criteria**

<b>Decommissioning Review</b>	
<b>Entrance Criteria</b>	<b>Success Criteria</b>
<ol style="list-style-type: none"> <li>1. Requirements associated with decommissioning and disposal.</li> <li>2. Plans for decommissioning, disposal, and any other removal from service activities.</li> <li>3. Resources in place to support decommissioning and disposal activities, plans for disposition of project assets, and archival of essential mission and project data.</li> <li>4. Description of safety, environmental and any other constraints.</li> <li>5. Description of the current system capabilities.</li> <li>6. For off-nominal operations, description of all contributing events, conditions, and changes to the originally expected baseline.</li> </ol>	<ol style="list-style-type: none"> <li>1. The reasons for decommissioning disposal are documented.</li> <li>2. The decommissioning and disposal plan is complete, approved by appropriate management, and compliant with applicable Agency safety, environmental, and health regulations. Operations plans for all potential scenarios, including contingencies, are complete and approved. All required support systems are available.</li> <li>3. All personnel have been properly trained for the nominal and contingency procedures.</li> <li>4. Safety, health, and environmental hazards have been identified. Controls have been verified.</li> <li>5. Risks associated with the disposal have been identified and adequately mitigated. Residual risks have been accepted by the required management.</li> <li>6. If hardware is to be recovered from orbit:               <ol style="list-style-type: none"> <li>a. Return site activity plans have been defined and approved.</li> <li>b. Required facilities are available and meet requirements, including those for contamination control, if needed.</li> <li>c. Transportation plans are defined and approved. Shipping containers and handling equipment, as well as contamination and environmental control and monitoring devices, are available.</li> </ol> </li> <li>7. Plans for disposition of mission-owned assets (hardware, software, facilities, etc.) have been defined and approved.</li> <li>8. Plans for archival and subsequent analysis of mission data have been defined and approved. Arrangements have been finalized for the execution of such plans. Plans for the capture and dissemination of appropriate lessons learned during the project life cycle have been defined and approved. Adequate resources (schedule, budget, and staffing) have been identified and are available to successfully complete all decommissioning, disposal, and disposition activities.</li> </ol>

**Objectives.** The objectives of the review are to:

- a. Ensure a thorough review of the products supporting the review.
- b. Ensure the products meet the entrance criteria and success criteria
- c. Ensure issues raised during the review are appropriately documented and a plan for resolution is prepared

**Results of Review.** A successful DR completion ensures that the decommissioning and disposal of system items and processes are appropriate and effective.

**Other Technical Reviews**

These typical technical reviews are some that have been conducted on previous programs and projects but are not required as part of NPR 7123.1 systems engineering process.

*Design Certification Review*

**Purpose.** The Design Certification Review (DCR) ensures that the qualification verifications demonstrated design compliance with functional and performance requirements.

**Timing.** Follows the system CDR, and after qualification tests and all modifications needed to implement qualification-caused corrective actions have been completed.

**Objectives.** The objectives of the review are to:

- Confirm that the verification results met functional and performance requirements, and that test plans and procedures were executed correctly in the specified environments
- Certify that traceability between test article and production article is correct, including name, identification number, and current listing of all waivers
- Identify any incremental tests required or conducted due to design or requirements changes made since test initiation, and resolve issues regarding their results.

**Criteria for Successful Completion.** The following items comprise a checklist to aid in determining the readiness of DCR product preparation:

- Are the pedigrees of the test articles directly traceable to the production units?
- Is the verification plan used for this article current and approved?
- Do the test procedures and environments used comply with those specified in the plan?
- Are there any changes in the test article configuration or design resulting from the as-run tests? Do they require design or specification changes, and/or retests?
- Have design and specification documents been audited?
- Do the verification results satisfy functional and performance requirements?
- Do the verification, design, and specification documentation correlate?

**Results of Review.** As a result of a successful DCR, the end item design is approved for production. All open issues should be resolved with closure actions and schedules.

#### *Functional and Physical Configuration Audits*

Configuration audits confirm that the configured product is accurate and complete. The two types of configuration audits are the Functional Configuration Audit (FCA) and the Physical Configuration audit (PCA). The FCA examines the functional characteristics of the configured product and verifies that the product has met, via test results, the requirements specified in its functional baseline documentation approved at the PDR and CDR. FCAs will be conducted on both hardware or software configured products and will precede the PCA of the configured product. The PCA (also known as a configuration inspection) examines the physical configuration of the configured product and verifies that the product corresponds to the “build-to” (or “code-to”) product baseline documentation previously approved at the CDR. PCAs will be conducted on both hardware and software configured products.

**Table 6-21 Functional and Physical Configuration Audits**

Representative Audit Data List	
FCA	PCA
<ul style="list-style-type: none"> <li>• Design specifications</li> <li>• Design drawings and parts list</li> <li>• Engineering Change Proposals/Engineering Change Requests (ECPs/ECRs)</li> <li>• Deviation/Waiver Approval Requests (DARs) incorporated and pending</li> <li>• Specification and drawing tree</li> <li>• Fracture Control Plan</li> <li>• Structural dynamics, analyses, loads, and models documentation</li> <li>• Materials Usage Agreements/Materials Identification Usage List (MUAs/MIULs)</li> <li>• Verification and Validation requirements, plans, procedures, and reports</li> <li>• Software requirements and development documents</li> <li>• Listing of accomplished tests and test results</li> <li>• Critical Design Review (CDR) completion documentation including Review Item Discrepancies/Requests For Actions (RIDs/RFAs) and disposition reports</li> <li>• Analysis Reports</li> <li>• ALERTS Tracking Log</li> <li>• Hazard analysis/risk assessment</li> </ul>	<ul style="list-style-type: none"> <li>• Final version of all specifications</li> <li>• Product drawings and parts list</li> <li>• Configuration accounting and status reports</li> <li>• Final version of all software documents</li> <li>• Copy of all Functional Configuration Audit (FCA) findings for each product</li> <li>• List of approved and outstanding ECPs, ECRs, and DARs</li> <li>• Indentured Parts List</li> <li>• As run test procedures</li> <li>• Drawing and specification tree</li> <li>• Manufacturing and inspection “build” records</li> <li>• Inspection records</li> <li>• As-build discrepancy reports</li> <li>• Product log books</li> </ul>

*Technical Peer Reviews*

Peer reviews provide the technical insight essential to ensure product and process quality. Peer reviews are focused, in-depth technical reviews that support the evolving design and development of a product, including critical documentation or data packages. They are often, but not always, held as supporting reviews for technical reviews such as PDR and CDR. A purpose of the peer review is to add value and reduce risk through expert knowledge, infusion, confirmation of approach, identification of defects, and specific suggestions for product improvements.

The results of the engineering peer reviews (EPRs) comprise a key element of the review process. The results and issues that surface during these reviews are documented and reported out at the appropriate next higher element level.

The peer reviewers should be selected from outside the project, but they should have a similar technical background, and they should be selected for their skill and experience. Peer reviewers should be selected to have as their only concern the technical integrity and quality of the product. Peer reviews should be kept simple and informal. They should concentrate on a review of the documentation and minimize the viewgraph presentations. A “round-table” format rather than a stand-up presentation is preferred. The peer reviews should give the full technical picture of items being reviewed.

Technical depth should be to a level that allows the review team to gain insight into the technical risks. Rules need to be established to ensure consistency in the peer review. At the conclusion of the review, a report on the findings and actions must be distributed.

Peer reviews must be part of the contract for those projects where systems engineering is done out-of-house.

Additional guidance on establishing and conducting peer reviews can be found in appendix 6-7.

#### 6.7.2.2 *Status Reporting and Assessment*

This section provides additional information on status reporting and assessment techniques for costs and schedules (including Earned Value Management), technical performance, and systems engineering process metrics.

### **Cost and Schedule Control Measures**

Status reporting and assessment on costs and schedules provides the project manager and systems engineer visibility into how well the project is tracking against its planned cost and schedule targets. From a management point of view, achieving these targets is on a par with meeting the technical performance requirements of the system. It is useful to think of cost and schedule status reporting and assessment as measuring the performance of the “system that produces the system.”

NPR 7120.5, *NASA Program and Project Management Processes and Requirements*, provides specific requirements for the application of Earned Value Management (EVM) to support cost and schedule management. EVM is applicable to both in-house and contracted efforts. The level of EVM System implementation will depend on the dollar value and risk of a project or contract. The standard for EVM Systems is ANSI-EIA-748. The project manager/systems engineer will use the guidelines to establish the program and project earned value management implementation plan.

#### *Assessment Methods*

Performance measurement data is used to assess project cost, schedule, and technical performance and their impacts on completion cost and schedule of the project. In program control terminology, a difference between actual performance and planned costs or schedule status is called a *variance*. Variances must be controlled at the control account level, which is typically at the subsystem WBS level. The person responsible for this activity is frequently called the “Control Account Manager” (CAM). The CAM develops work and product plans, schedules, and time phased resource plans. The technical subsystem manager/lead often takes on this role as part of their subsystem management responsibilities.

Figure 6-27 illustrates two types of variances, cost and schedule, and some related concepts. A product oriented Work Breakdown Structure (WBS) divides the project work into discrete tasks and products. Associated with each task and product (at any level in the WBS) is a schedule and a budgeted (i.e., planned) cost. The *Budgeted Cost for Work Scheduled* ( $BCWS_t$ ) for any set of WBS elements is the sum of the budgeted cost of all work on tasks and products in those

elements scheduled to be completed by time  $t$ . The *Budgeted Cost for Work Performed* ( $BCWP_t$ ), also called *Earned Value* ( $EV_t$ ), is the sum of the budgeted cost for tasks and products that have actually been produced (completed or in progress) at time  $t$  in the schedule for those WBS elements. The difference,  $BCWP_t - BCWS_t$ , is called the schedule variance at time  $t$ . A negative value indicates that the work is behind schedule.

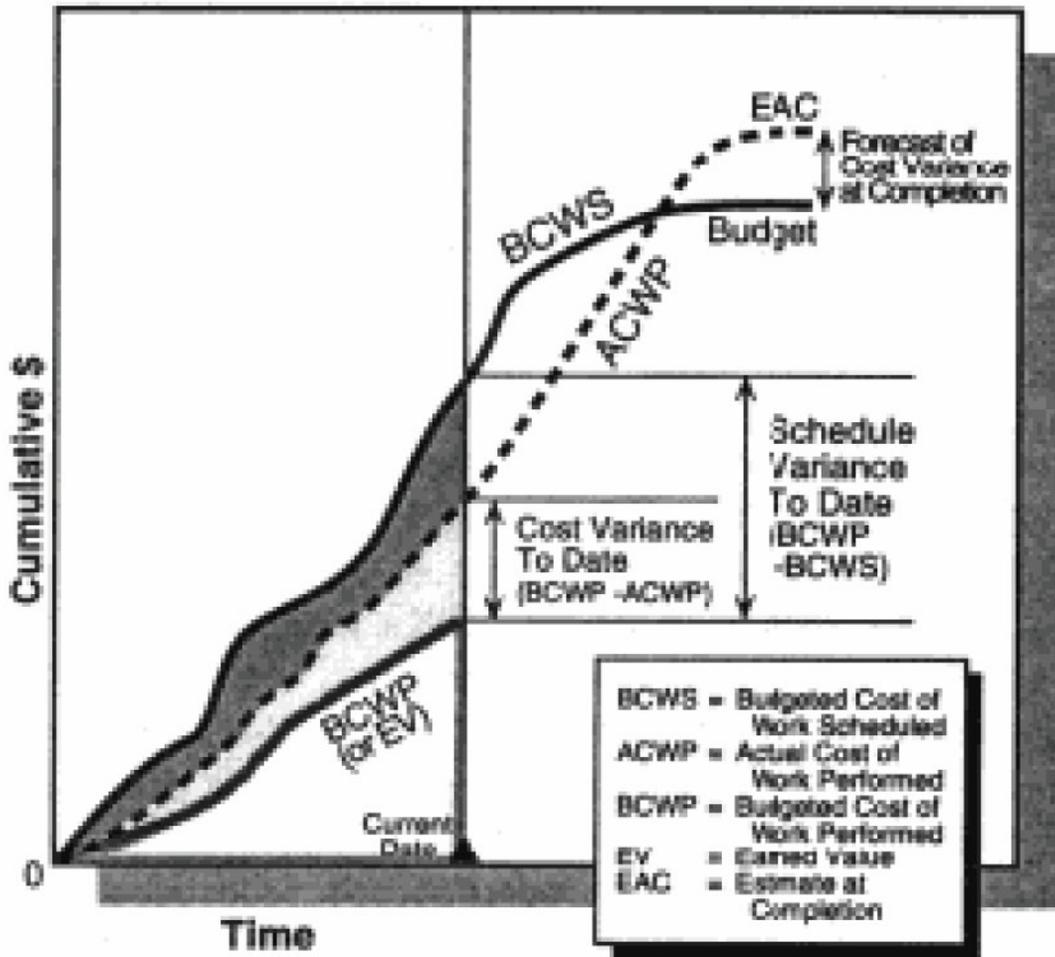


Figure 6-27 Cost and Schedule Variances

The *Actual Cost of Work Performed* ( $ACWP_t$ ) represents the funds that have been expended up to time  $t$  on those WBS elements. The difference between the budgeted and actual costs,  $BCWP_t - ACWP_t$ , is called the cost variance at time  $t$ . A negative value here indicates a cost overrun.

When either schedule variance or cost variance exceeds pre-established control-account-level thresholds that represent significant departures from the baseline plan, the conditions must be analyzed to identify why the variance exists. Once the cause is understood, the Control Account Manager (CAM) can make an informed forecast of the time and resources needed to complete the control account. When corrective actions are feasible (can stay within BCWS), the plan for implementing them must be included in the analysis. Sometimes no corrective action is feasible; overruns or schedule slips may be unavoidable. One must keep in mind that the earlier a technical problem is identified as a result of schedule or cost variances, the more likely the project team can minimize the impact on completion.

Variances may indicate that the cost *Estimate at Completion* ( $EAC_t$ ) of the project is likely to be different from the Budget at Completion (BAC). The difference between the BAC and the EAC is the Variance at Completion (VAC). A negative VAC is generally unfavorable, while a positive is usually favorable. These variances may also point toward a change in the scheduled completion date of the project. These types of variances enable a program analyst to estimate the EAC at any point in the project life cycle. (See sidebar on analyzing EAC.) These analytically-derived estimates should be used only as a “sanity check” against the estimates prepared in the variance analysis process.

If the cost and schedule baselines and the technical scope of the work are not adequately defined and fully integrated, then it is very difficult (or impossible) to estimate the current cost EAC of the project.

Other efficiency factors can be calculated using the performance measurement data. The Schedule Performance Index (SPI) is a measure of work accomplishment in dollars. The SPI is calculated by dividing work accomplished in dollars or BCWP by the dollar value of the work scheduled or BCWS. Just like any other ratio, a value less than one is a sign of a behind schedule condition; equal to one indicates an on schedule status; and greater than one denotes that work is ahead of schedule. The Cost Performance Index (CPI) is a measure of cost efficiency and is calculated as the ratio of the earned value or BCWP for a segment of work compared to the cost to complete that same segment of work or ACWP. A CPI will show how much work is being accomplished for every dollar spent on the project. A CPI of less than one reveals negative cost efficiency; equal to one is right on cost; and greater than one is positive. Note that traditional measures compare planned cost to actual cost, however this comparison is never made using EV data. Comparing planned to actual costs is an indicator only of spending and not of overall project performance.

### Analyzing the Estimate at Completion

An EAC can be estimated at any point in the project and should be reviewed at least on a monthly basis. The EAC requires a detailed review by the Control Account Manager (CAM). A statistical estimate can be used as a cross-check of the CAM's estimate and to develop a range to bound the estimate. The appropriate formula used to calculate the statistical EAC depends upon the reasons associated for any variances that may exist. If a variance exists due to a one-time event, such as an accident, then  $EAC = ACWP + (BAC - BCWP)$ . The CPI and SPI should also be considered in developing the EAC.

If there is a growing number of liens, action items, or significant problems that will increase the difficulty of future work, the EAC might grow at a greater rate than estimated by the above equations. Such factors could be addressed using risk management methods described in the Risk Management section.

### Technical Measures—MOEs, MOPs, and TPMs

#### *Measure of Effectiveness*

Measures of Effectiveness are the “operational” measures of success that are closely related to the achievement of mission or operational objectives in the intended operational environment. MOEs are intended to focus on how well mission or operational objectives are achieved, not on how they are achieved, i.e., MOEs should be independent of any particular solution. As such, MOEs are the standards against which the “goodness” of each proposed solution may be assessed in trade studies and decision analyses. Measuring or calculating MOEs not only makes it possible to compare alternative solutions quantitatively, but sensitivities to key assumptions regarding operational environments and to any underlying Measures of Performance (MOPs) can also be investigated. (See MOP discussion below.)

In the systems engineering process, Measures of Effectiveness are used to:

- Define high-level operational requirements from the customer/stakeholder viewpoint.
- Compare and rank alternative solutions in trade studies.
- Investigate the relative sensitivity of the projected mission or operational success to key operational assumptions and performance parameters.
- Determine that the mission or operational success quantitative objectives remain achievable as system development proceeds. (See TPM discussion below.)

#### *Measure of Performance*

Measures of Performance are the measures that characterize physical or functional attributes relating to the system, e.g., engine  $I_{sp}$ , max thrust, mass, and payload-to-orbit. These attributes are generally measured under specified test conditions or operational environments. MOPs are attributes deemed important in achieving mission or operational success, but do not measure it directly. Usually multiple MOPs contribute to an MOE. MOPs often become system performance requirements that, when met by a design solution, result in achieving a critical threshold for the system MOEs.

The distinction between MOEs and MOPs is that they are formulated from different viewpoints. A MOE refers to the effectiveness of a solution from the mission or operational success criteria

expressed by the user/customer/stakeholder. An MOE represents a stakeholder expectation that is critical to the success of the system and failure to attain a critical value for it will cause the stakeholder to judge the system a failure. An MOP is a measure of actual performance of a (supplier's) particular design solution, which taken alone may only be indirectly related to the customer/stakeholder's concerns.

### *Technical Performance Measures*

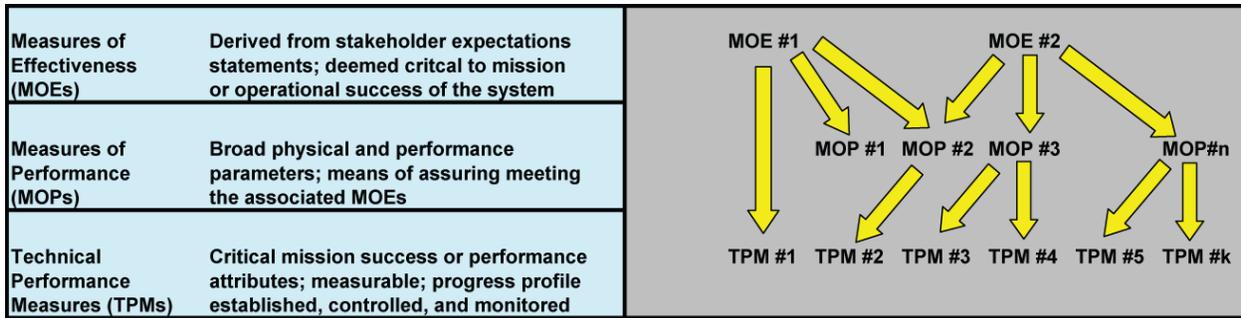
Technical Performance Measures are critical or key mission success or performance parameters that are monitored during implementation by comparing the current actual achievement of the parameters with the values that were anticipated for the current time and projected for future dates. They are used to confirm progress and identify deficiencies that might jeopardize meeting a system requirement or put the project at cost or schedule risk. When a TPM value falls outside the expected range around the anticipated value, it signals a need for evaluation and corrective action.

In the systems engineering process, TPMs are used to:

- Forecast values to be achieved by critical parameters at major milestones or key events during implementation.
- Identify differences between the actual and planned values for those parameters.
- Provide projected values for those parameters in order to assess the implications for system effectiveness.
- Provide early warning for emerging risks requiring management attention (when negative margins exist).
- Provide early identification of potential opportunities to make design trades that reduce risk or cost, or increase system effectiveness (when positive margins exist).
- Support assessment of proposed design changes.

### *Selecting TPMs*

Technical Performance Measures are typically selected from the defined set of MOEs and MOPs. Care should be exercised in selecting a small set of succinct TPMs that accurately reflect key parameters or risk factors, are readily measurable, and that can be affected by altering design decisions. In general, TPMs can be generic (attributes that are meaningful to each Product Breakdown Structure (PBS) element, like mass or reliability) or unique (attributes that are meaningful only to specific PBS elements). The relationship of MOEs, MOPs, and TPMs can be found in Figure 6-28. The systems engineer needs to decide which generic and unique TPMs are worth tracking at each level of the PBS. (See sidebar on examples of TPMs.) At lower levels of the PBS, TPMs worth tracking can be identified through the functional and performance requirements levied on each individual system, subsystem, etc.



**Figure 6-28 Relationships of MOEs, MOPs, and TPMs**

As TPMs are intended to provide an early warning of the adequacy of a design in satisfying selected critical technical parameter requirements, the systems engineer should select TPMs that fall within well-defined (quantitative) limits for reasons of system effectiveness or mission feasibility. Usually these limits represent either a firm upper or lower bound constraint. A typical example of such a TPM for a spacecraft is its injected mass, which must not exceed the capability of the selected launch vehicle. Tracking injected mass as a high-level TPM is meant to ensure that this does not happen. A high-level TPM like injected mass must often be “budgeted” and allocated to multiple system elements. Tracking and reporting should be required at these lower levels to gain visibility into the sources of any variances.

## Examples of Technical Performance Measures

### TPMs from Measures of Effectiveness

Mission performance (e.g., total science data volume returned)

Safety (e.g., probability of loss of crew,  $\Pr\{\text{LOC}\}$ , and loss of mission,  $\Pr\{\text{LOM}\}$ )

Achieved availability (e.g.,  $(\text{system uptime})/(\text{system uptime} + \text{system downtime})$ )

### TPMs from Measures of Performance

Thrust versus predicted/specified

$I_{sp}$  versus predicted/specified

End-of-Mission (EOM) dry mass

Injected mass (includes EOM dry mass, baseline mission plus reserve propellant, other consumables and upper stage adaptor mass)

Propellant margins at EOM

Other consumables margins at EOM

Electrical power margins over mission life

Control system stability margins

EMI/EMC susceptibility margins

Onboard data processing memory demand

Onboard data processing throughput time

Onboard data bus capacity

Total pointing error

Total vehicle mass at launch

Payload mass (at nominal altitude or orbit)

Reliability

Mean time before refurbishment required

Total crew maintenance time required

System turn-around time

Fault detection capability

Percentage of system designed for on-orbit crew access

In summary, for a TPM to be a valuable status and assessment tool, certain criteria must be met:

- Be a significant descriptor of the system (e.g., weight, range, capacity, response time, safety parameter) that will be monitored at key events (e.g., reviews, audits, planned tests).
- Can be measured (either by test, inspection, or analysis).
- Is such that reasonable projected progress profiles can be established (e.g., from historical data or based on test planning).

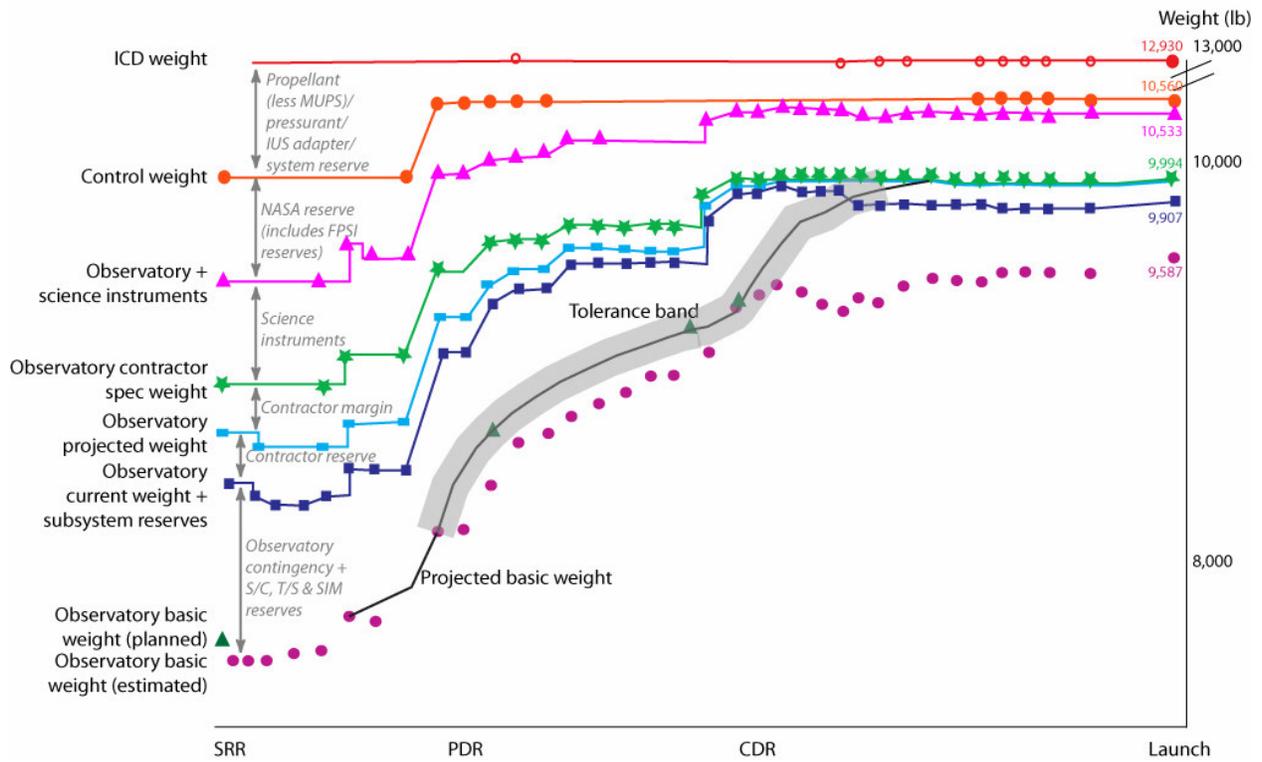
### *TPM Assessment and Reporting Methods*

Status reporting and assessment of the system's TPMs complements cost and schedule control.

There are a number of assessment and reporting methods that have been used on NASA projects, including the planned profile method and the margin management method.

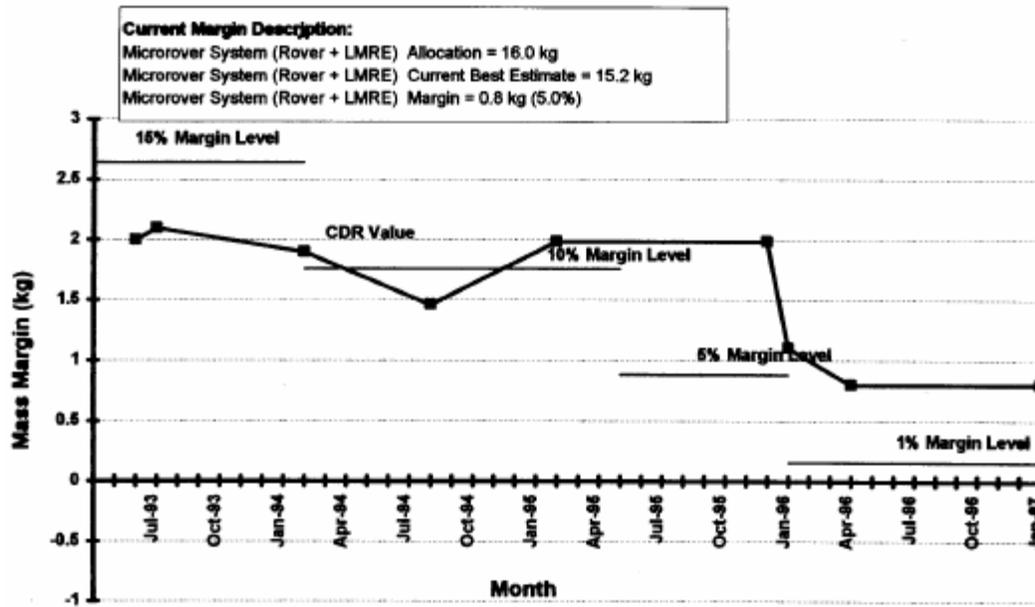
A detailed example of the planned profile method for the Chandra Project weight TPM is illustrated in Figure 6-29. This figure depicts the subsystem contributions, various constraints, project limits and management reserves from project SRR to launch.

### Chandra Project: Weight Changes



**Figure 6-29 Use of the Planned Profile Method for the Weight TPM with Rebaseline in Chandra Project**

A detailed example of the margin management method for the *Sojourner* mass TPM is illustrated in Figure 6-30. This figure depicts the margin requirements (horizontal straight lines) and actual mass margins from project SRR to launch.



**Figure 6-30 Use of the Margin Management Method for the Mass TPM in the Microrover Flight Experiment (*Sojourner*)**

*Relationship of TPM Assessment Program to the SEMP*

The SEMP is the usual document for describing the project’s TPM assessment program. This description should include a master list of those TPMs to be tracked, and the measurement and assessment methods to be employed. If analytical methods and models are used to measure certain high-level TPMs, then these need to be identified. The reporting frequency and timing of assessments should be specified as well. In determining these, the systems engineer must balance the project’s needs for accurate, timely, and effective TPM tracking against the cost of the TPM tracking program.

The TPM assessment program plan, which may be a part of the SEMP or a standalone document for large programs/projects, should specify each TPM’s allocation, time-phased planned profile or margin requirement, and alert zones, as appropriate to the selected assessment method..

A formal TPM assessment program should be fully planned and baselined with the SEMP. Tracking TPMs should begin as soon as practical in Phase B. Data to support the full set of selected TPMs may, however, not be available until later in the project life cycle. As the project life cycle proceeds through Phases C and D, the measurement of TPMs should become increasingly more accurate with the availability of more “actual” data about the system.

For the WBS model in the system structure, typically the following activities are performed:

- Analyze stakeholder expectation statements to establish a set of MOEs by which overall system or product effectiveness will be judged and customer satisfaction will be determined.

- Define MOPs for each identified MOE.
- Define appropriate TPMs and document the TPM assessment program in the SEMP.

## **Systems Engineering Process Metrics**

Status reporting and assessment of systems engineering process metrics provides additional visibility into the performance of the “system that produces the system.” As such, these metrics supplement the cost and schedule control measures discussed in this section

Systems engineering process metrics try to quantify the effectiveness and productivity of the systems engineering process and organization. Within a single project, tracking these metrics allows the systems engineer to better understand the health and progress of that project. Across projects (and over time), the tracking of systems engineering process metrics allows for better estimation of the cost and time of performing systems engineering functions. It also allows the systems engineering organization to demonstrate its commitment to continuous improvement.

### *Selecting Systems Engineering Process Metrics*

Generally, systems engineering process metrics fall into three categories — those that measure the progress of the systems engineering effort, those that measure the quality of that process, and those that measure its productivity. Different levels of systems engineering management are generally interested in different metrics. For example, a project manager or lead systems engineer may focus on metrics dealing with systems engineering staffing, project risk management progress, and major trade study progress. A subsystem systems engineer may focus on subsystem requirements and interface definition progress and verification procedures progress. It is useful for each systems engineer to focus on just a few process metrics. Which metrics should be tracked depends on the systems engineer’s role in the total systems engineering effort. The systems engineering process metrics worth tracking also change as the project moves through its life cycle.

Collecting and maintaining data on the systems engineering process is not without cost. Status reporting and assessment of systems engineering process metrics divert time and effort from the activities itself. The systems engineer must balance the value of each systems engineering process metric against its collection cost. The value of these metrics arises from the insights they provide into the activities that cannot be obtained from cost and schedule control measures alone. Over time, these metrics can also be a source of hard productivity data, which are invaluable in demonstrating the potential returns from investment in systems engineering tools and training.

### *Examples and Assessment Methods*

Table 6-22 lists some systems engineering process metrics to be considered. This list is not intended to be exhaustive. Because some of these metrics allow for different interpretations, each NASA field center needs to define them in a common-sense way that fits its own processes. For example, each field center needs to determine what is meant by a *completed* versus an *approved* requirement, or whether these terms are even relevant. As part of this definition, it is important to recognize that not all requirements, for example, need be lumped together. It may be more useful to track the same metric separately for each of several different types of requirements.

**Table 6-22 Systems Engineering Process Metrics**

Function	Systems Engineering Process Metric	Category
Requirements development and management	Requirements identified vs. completed vs. approved	S
	Requirements volatility	Q
	Trade studies planned vs. completed	S
	Requirements approved per systems engineering hour	P
Design and development	Specifications planned vs. completed	S
	Processing of ECRs/ECOs	Q
	Engineering drawings planned vs. released	S
Verification and validation (V&V)	V&V plans identified vs. approved	S
	V&V procedures planned vs. completed	S
	Functional requirements approved vs. verified	S
	V&V plans approved per systems engineering hour	P
	Processing of Problem/Failure Reports	Q
Reviews	Processing of review item discrepancies (RIDs)	Q
	Processing of action items	Q

S = progress or schedule related; Q = quality related; P = productivity related

Quality-related (Q) metrics should serve to indicate when a part of the systems engineering process is overloaded and/or breaking down. These metrics can be defined and tracked in several different ways. For example, requirements volatility can be quantified as the number of newly identified requirements, or as the number of changes to already-approved requirements. As another example, Engineering Change Request (ECR) processing could be tracked by comparing cumulative ECRs opened versus cumulative ECRs closed, or by plotting the age profile of open ECRs, or by examining the number of ECRs opened last month versus the total number open. The systems engineer should apply his/her own judgment in picking the status reporting and assessment method.

Productivity-related (P) metrics provide an indication of systems engineering output per unit of input. Although more sophisticated measures of input exist, the most common is the number of systems engineering hours dedicated to a particular function or activity. Because not all systems engineering hours cost the same, an appropriate weighing scheme should be developed to ensure comparability of hours across systems engineering personnel.

Schedule-related (S) metrics can be depicted in a table or graph of planned quantities vs. actuals. For example, comparing planned number of verification closure notices against actual. This metric should not be confused with EVM described in this section. EVM is focused on integrated cost and schedule at the desired level whereas this metric focuses on an individual process or product within a subsystem, system or project itself.

The combination of quality- and productivity- and schedule-related metrics, can provide trends that are generally more important than isolated snapshots. The most useful kind of assessment method allows comparisons of the trend on a current project with that for a successfully

completed project of the same type. The latter provides a benchmark against which the systems engineer can judge his/her own efforts.

## 6.8 Decision Analysis

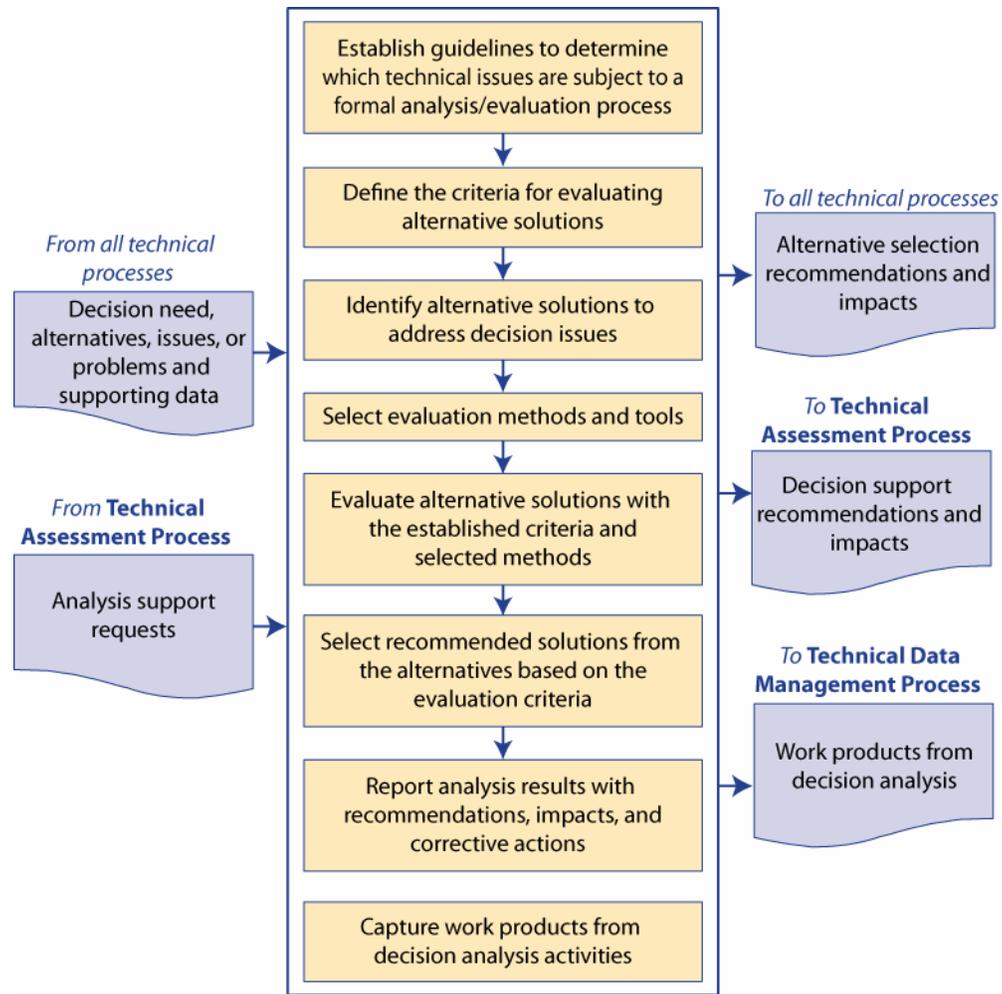
The purpose of this section is to provide a description of the Decision Analysis Process, including alternative tools and methodologies. Decision analysis offers individuals and organizations a methodology for making decisions; it also offers techniques for modeling decision problems mathematically and finding optimal decisions numerically. Decision models have the capacity for accepting and quantifying human subjective inputs: judgments of experts and preferences of decision-makers. Implementation of models can take the form of simple paper-and-pencil procedures or sophisticated computer programs known as decision aids or decision systems. The methodology is broad and must always be adapted to the issue under consideration. The problem is structured by identifying alternatives, one of which must be decided upon; possible events, one of which occurs thereafter; and outcomes, each of which results from a combination of decision and event. Decisions are made throughout a program/project life cycle and often are made through a hierarchy of panels and boards and teams with increasing, complementary authority, wherein each progressively more detailed decision is affected by the assumptions made at the lower level. Not all decisions need a formal process, but it is important to establish a process for those decisions which require a formal process. Important decisions must be completely documented as well as supporting information (including assumptions made), tools, and models so that new information can be incorporated and assessed and past decisions can be researched in context. The Decision Analysis Process accommodates this iterative environment and occurs throughout the project life cycle.

The outputs from this process support the decision makers difficult task of deciding among competing alternatives without complete knowledge, therefore it is critical to understand the assumptions and limitation of any tool or methodology and integrate that with other factors when deciding among viable options.

Early in the project life cycle, high-level decisions are made regarding which technology could be used, such as solid or liquid rockets for propulsion. Without specifying the component-level detail of each design alternative, operational scenarios, probabilities, and consequences are determined, and the design decision is made. Once high-level design decisions are made, nested systems engineering processes occur at progressively more detailed design levels flowed down through the entire system. Each progressively more detailed decision is affected by the assumptions made at the lower level. For example, the solid rocket design is constrained by the operational assumptions made during the decision process that selected that design. This is an iterative process between elements of the system. Also early on in the life cycle, the technical team should determine the types of data and information products required to support the Decision Analysis Process during the later stages of the project and design, develop, or acquire the models, simulations, and other tools that will supply the required information to decision makers. In this section application of different levels and kinds of analysis are discussed at different stages of the project life cycle.

### 6.8.1 Process Description

The Decision Analysis Process is used to help evaluate technical issues, alternatives, and their uncertainties to support decisionmaking. A typical process flow diagram is provided in Figure 6-31, including inputs, activities, and outputs.



**Figure 6-31 Decision Analysis Process**

Typical processes that use decision analysis are:

- Determining how to allocate limited resources (e.g., budget, mass, power, etc.) among competing subsystem interests to favor the overall outcome of the project.
- Configuration management processes for major change requests or problems reports.
- Design processes for making major design decisions and selecting design approaches.
- Key decision point reviews or technical review decisions (e.g., PDR, CDR) as defined in NPR 7120.5 and 7123.1.
- Go/No Go decisions (e.g., FRR)
- Go—authorization to proceed.
- No go—repeat some specific aspects of development or conduct further research.

- Project management of major issues, schedule delays, budget increases.
- Procurement of major items.
- Technology decisions.
- Risk management of major risks (e.g., red or yellow).
- Safety and Mission Assurance decisions.
- Miscellaneous decisions (e.g., whether to intervene in the project to address an emergent performance issue).

Decision analysis can also be used in emergency situations. Under such conditions process steps, procedures, and meetings can be combined, and the decision analysis documentation may be completed at the end of the process (i.e., after the decision is made). However, a decision matrix should be completed and used during the decision. Decision analysis documentation must be archived as soon as possible following the emergency situation.

#### *6.8.1.1 Inputs*

Formal decision analysis has the potential to consume significant resources and time. Typically, its application to a specific decision is warranted only when some of the following conditions are met:

- **High stakes:** High stakes are involved in the decision.
- **Complexity:** The actual ramifications of alternatives are difficult to understand without detailed analysis.
- **Uncertainty:** Uncertainty in key inputs creates substantial uncertainty in the ranking of alternatives and points to risks that may need to be managed.
- **Multiple attributes:** A greater numbers of attributes cause a greater need for formal analysis.
- **Diversity of stakeholders:** Extra attention is warranted to clarify objectives and formulate technical performance measures (TPMs) when the set of stakeholders reflects a diversity of values, preferences, and perspectives.

Satisfaction of all of these conditions is not a requirement for initiating decision analysis. The point is rather that the need for decision analysis increases as a function of the above conditions. When the Decision Analysis Process is triggered, the following are inputs:

- Decision need, identified alternatives, issues, or problems and supporting data (from all technical management processes)
- Analysis support requests (from Technical Assessment Process)

- High-level objectives (from the program/project)

### 6.8.1.2 Process Activities

For the Decision Analysis Process, the following activities are typically performed:

#### **Establish Guidelines to Determine Which Technical Issues Are Subject to a Formal Analysis/Evaluation Process**

This step includes:

1. When to use a formal decision making procedure,
2. What needs to be documented,
3. Who will be the decision makers and their responsibilities and decision authorities, and
4. How decisions will be handled that do not require a formal evaluation procedure.

Decisions are based on facts and open communications to facilitate the flow of information throughout the hierarchy of forums where technical analyses and evaluations are presented and assessed and where decisions are made. The extent of technical analysis and evaluation required should be commensurate with the consequences of the issue requiring a decision. The work required to conduct a formal evaluation is not insignificant and applicability must be based on the nature of the problem to be resolved. Guidelines for use can be determined by the magnitude of the possible consequences of the decision to be made.

For example, the consequence table from a risk scorecard can be used to ascribe numerical values for applicability according to impacts to mission success, flight safety, cost and schedule. Actual numerical thresholds for use would then be set by a decision authority. Sample values could be as shown in Table 6-23:

**Table 6-23 Consequence Table**

<b>Numerical Value</b>	<b>Consequence</b>	<b>Applicability</b>
Consequence = 5, 4	High	Mandatory
Consequence = 3	Moderate	Optional
Consequence = 1, 2	Low	Not Required

#### **Define the Criteria for Evaluating Alternative Solutions**

This step includes:

1. The types of criteria to consider, including technology limitations, environmental impact, safety, risks, total ownership and life-cycle costs, and schedule impact;
2. The acceptable range and scale of the criteria; and
3. The rank of each criterion by its importance.

This step includes identifying criteria, establishing a range and scale for values to be used, and a method for doing ranking. Decision criteria are requirements for individually assessing options and alternatives being considered. Typical decision criteria are cost, schedule, risk, safety, mission success, supportability, etc. However, considerations should include technical criteria specific to the decision being made. Criteria should be objective and measurable. Criteria should also permit distinguishing among options or alternatives. Some criteria may not be meaningful to a decision; however, they should be documented as having been considered. Identify criteria that are mandatory (i.e., “must have”) versus the other criteria (i.e., “nice to have”). If mandatory criteria are not met, that option should be disregarded. For complex decisions, criteria can be grouped into categories or objectives (e.g., see the analytical hierarchy process).

Ranking or prioritizing the criteria is probably the hardest part of completing a decision matrix. Ranking is typically accomplished by assigning weights to each. The idea is that not all criteria have the same importance. To avoid gaming the decision matrix (i.e., changing decision outcomes by playing with criteria weights), it is best to agree upon weights before the decision matrix is completed. Weights should only be changed with consensus from all decision stakeholders.

*For example*, ranking can be done using a simple approach like using percentage. Have all the weights for each criterion add up to 100. Assign the percents based on how important the criteria is. (The higher the number of percentages, the more important, such as a single criterion worth 50 percent). The weights need to be divided by 100 to calculate percents. Using this approach, the option with the highest percentage is typically the recommended option. Ranking can also be done using sophisticated decision tools. For example, pair-wise comparison is a decision technique that calculates the weights using paired comparisons among criteria and options. Other methods include:

- Formulation of objectives hierarchy and technical performance measures (TPMs)
- Analytical hierarchy process, which addresses criteria and paired comparisons
- Risk-based decision analysis process with weighting of TPMs

### **Identify Alternative Solutions to Address Decision Issues**

This step includes alternatives for consideration in addition to those that may be provided with the issue.

Almost every decision will have options to choose from. Brainstorm decision options, and document option summary names for the available options. For complex decisions, it is also a best practice to perform a literature search to identify options. Reduce the decision options down to a reasonable set (e.g., seven plus or minus two). Some options will obviously be bad options. Document the fact that these options were considered. The use of mandatory criteria can also help reduce the number of options. A few decisions may only have one option. It is a best practice to document a decision matrix even for one option if it is a major decision. (Sometimes doing nothing or not making a decision is an option.)

## Select Evaluation Methods and Tools

Select evaluation methods (see note below) and tools/techniques based on the purpose for analyzing a decision and on the availability of the information used to support the method and/or tool.

Typical evaluation methods include: simulations; weighted trade-off matrices; engineering, manufacturing, cost, and technical opportunity of trade studies; surveys; extrapolations based on field experience and prototypes; user review and comment; testing, etc.

Tools and techniques to be used should be selected based on the purpose for analyzing a decision and on the availability of the information used to support the method and/or tool. Regardless of method or tool used, results must include an evaluation of assumptions related to the evaluation criteria used and of the evidence that supports the assumptions and an evaluation of whether uncertainty in the values for alternative solutions affects the evaluation.

Additional evaluation methods include:

- Decision matrix (Figure 6-32 in this section.)
- Decision analysis process support, evaluation methods, and tools

## Evaluate Alternative Solutions with the Established Criteria and Selected Methods

This step includes:

1. Evaluation of assumptions related to evaluation criteria and of the evidence that supports the assumptions, and
2. Evaluation of whether uncertainty in the values for alternative solutions affects the evaluation.

Alternatives can be compared to evaluation criteria via the use of a decision matrix shown below as Figure 6-32. Evaluation criteria are typically in the rows on the left-hand side of the matrix. Alternatives are typically the row headings on the top of the matrix (and to the right top). Criteria “weights” are typically assigned to each criterion. In the example below, there are also “mandatory criteria.” If mandatory criteria are not met, the option is scored at 0 percent.

When decision criteria have different measurement bases (e.g., numbers, money, weight, dates, etc.), normalization can be used to establish a common base for mathematical operations. The process of “normalization” is making a scale so that all different kinds of criteria can be compared or added together. This can be done informally (e.g., low, medium, high), on a scale (e.g., 1-3-9), or more formally with a tool. No matter how normalization is done, the most important thing to remember is to have operational definitions of the scale. An operational definition is a repeatable, measurable number. For example, “high” could mean “a probability of 67 percent and above.” “Low” could mean “a probability of 33 percent and below.” For complex decisions, decision tools usually provide an automated way for normalization. Be sure to question and understand the operational definitions for the weights and scales of the tool.

Note: Completing the decision matrix can be thought of as a default evaluation method. Completing the decision matrix is iterative in nature. Each cell for each criteria and each option needs to be completed by the team. Use evaluation methods as needed to complete the entire decision matrix.

Decision Matrix Example for Battery EOL			ENTER SCORES	Extend Old Battery Life	Buy New Batteries	Collect Experient Data with Alternative Experiment	Cancelled Experiment
CRITERIA	Mandatory (Y=1/N=0)?	Weight	SCALE				
Mission Success (Get Experiment Data)	1	30	3 = Most Supportive 1 = Least Supportive	2	3	3	0
Cost per Option	0	10	3 = Least Expensive 1 = Most Expensive	1	2	3	1
Risk (Overall Option Risk)	0	15	3 = Least Risk 1 = Most Risk	2	1	2	3
Schedule	0	10	3 = Shortest Schedule 1 = Longest Schedule	3	2	1	3
Safety	1	15	3 = Most Safe 1 = Least Safe	2	1	2	3
Uninterrupted Data Collection	0	20	3 = Most Supportive 1 = Least Supportive	3	1	2	1
WEIGHTED TOTALS in %		100.00%	3	73.33%	60.00%	76.67%	0.00%
			SCALE 1-3				

Figure 6-32 Example Decision Matrix

Other methods include:

- Decision analysis process support, evaluation methods, and tools
- Risk-based decision analysis process
- Trade-off analysis and decision alternatives

### Select Recommended Solutions from the Alternatives Based on the Evaluation Criteria

This step includes documenting the information that justifies the recommendations and gives the impacts of taking the recommended course of action.

The highest score (e.g., percentage, total score, etc.) is typically the option that is recommended to management. If a different option is recommended, an explanation must be provided as to why the lower score is preferred. Usually, if a lower score is recommended, the “risks” or “disadvantages” were too great for the highest score. Sometimes the benefits and advantages of a lower or close score outweigh the highest score. Ideally, all risks/benefits and advantages/disadvantages would show up in the decision matrix as criteria. But this is not always possible. Sometimes if there is a lower score being recommended, the weighting or scores given may not be accurate.

This step also includes documenting the information, including assumptions and limitations of the evaluation methods used, that justifies the recommendation made and gives the impacts of taking the recommended course of action.

### **Report the Analysis and Evaluation Results and Findings with Recommendations, Impacts, and Corrective Actions**

Typically a technical team of subject matter experts makes a recommendation to a NASA decision maker (e.g., a NASA board, forum, panel, etc.). It is highly recommended that the team produce a white paper to document all major recommendations to serve as a backup to any presentation materials used. A presentation can also be used, but a paper in conjunction with a decision matrix is preferred (especially for complex decisions). Decisions are typically captured in meeting minutes, but can be captured in the white paper.

### **Capture Work Products from Decision Analysis Activities**

This step includes:

1. Decision analysis guidelines generated and strategy and procedures used;
2. Analysis/evaluation approach, criteria, and methods and tools used;
3. Analysis/evaluation results, assumptions made in arriving at recommendations, uncertainties, and sensitivities of the recommended actions or corrective actions; and
4. Lessons learned and recommendations for improving future decision analyses.

Typical information captured in a decision report is shown in Table 6-24:

**Table 6-24 Typical Information to Capture in a Decision Report**

#	Section	Section Description
1	Executive Summary	Describe a short half-page executive summary of the report: Recommendation (short summary—1 sentence) Problem/issue requiring a decision (short summary—1 sentence)
2	Problem/Issue Description	Describe the problem/issue that requires a decision. Provide background, history, the decisionmaker(s) (e.g., board, panel, forum, council), and decision recommendation team, etc.
3	Decision Matrix Set-Up Rationale	Provide the rationale for setting up the decision matrix: Criteria selected Options selected Weights selected Evaluations methods selected Provide a copy of the set-up decision matrix
4	Decision Matrix Scoring Rationale	Provide the rationale for the scoring of the decision matrix. Provide the results of populating the scores of the matrix using the evaluation methods selected.

#	Section	Section Description
5	Final Decision Matrix	Cut and paste the final MS Excel spreadsheet into the MS Word document. Also include any important snapshots of the decision matrix.
6	Risk/Benefits	For the final options being considered, document the risks and benefits of each option.
7	Recommendation and/or Final Decision	Describe the recommendation that is being made to the decisionmaker(s) and the rationale why the option was selected. Can also document the final decision in this section.
8	Dissent	If applicable, document any dissent with the recommendation (e.g., a team member disagrees with the recommendation). Document how dissent was addressed (e.g., decision matrix, risk, etc.).
9	References	Provide any references.
A	Appendices	Provide the results of the literature search, including lessons learned, previous related decisions, and previous related dissent. Also document any detailed data analysis and risk analysis used for the decision. Can also document any decision metrics.

(Reference: NASA JSC Decision Process)

### 6.8.1.3 *Outputs*

Decision analysis continues throughout the life cycle. The resulting products from decision analysis include:

- Alternative selection recommendations and impacts (to all Technical Management Processes)
- Decision support recommendations and impacts (to Technical Assessment Process)
- Work products of decision analysis activities (to Technical Data Management Process)
- Technical risk status measurements (to Technical Risk Management Process)
- TPMs, PIs for alternatives, the program- or project-specific objectives hierarchy, and the decisionmakers' preferences (to all Technical Management Processes)

## 6.8.2 *Decision Analysis Guidance*

The purpose of this subsection is to provide guidance, methods, and tools to support the Decision Analysis Process at NASA.

### 6.8.2.1 *Systems Analysis, Simulation, and Performance*

Systems analysis can be better understood in the context of the system's overall life cycle [Blanchard and Fabrycky]. Systems analysis within the context of the life cycle is responsive to

the needs of the customer at every phase of the life cycle, from pre-Phase A through Phase B to realizing the final product and beyond (refer to Figure 6-33, Figure 6-34, and **Error! Reference source not found.**).

Systems analysis of a product must support the transformation from a need into a realized, definitive product, be able to support compatibility with all physical and functional requirements, and support the operational scenarios in terms of reliability, maintainability, supportability, serviceability, and disposability, while maintaining performance and affordability.

Systems analysis support is provided from cradle to grave of the system. This covers the product design, verification, manufacturing, operations and support, and disposal. Viewed in this manner, life-cycle engineering is the basis for concurrent engineering.

Systems analysis should support concurrent engineering. Appropriate systems analysis can be conducted early in the life cycle to support planning and development. The intent here is to support seamless systems analysis optimally planned across the entire life cycle. For example, systems engineering early in the life cycle can support optimal performance of the deployment, operations, and disposal facets of the system.

Historically, this has not been the case. Systems analysis would only focus on the life cycle that the project occupied at that time. The systems analyses for the later phases were treated serially, in chronological order. This resulted in major design modifications that were very costly in the later life-cycle phases. Resources can be used more efficiently if the requirements across the life cycle are considered concurrently, providing results for decision-making about the system.

Figure 6-33 shows a life-cycle chart that indicates how the various general types of systems analyses fit across the life cycle. The analysis begins with a broader scope in the early phases, and narrows in scope as decisions are made and requirements become clearer. **Error! Reference source not found.** is a more specific version of the chart and addresses an example of how Operations Analysis fits into each of the life-cycle phases. This shows the evolution of ConOps from X through Z as well as specific example tools in each phase. **Error! Reference source not found.** shows that Operations Analysis should be included in the early life-cycle phases to be effective. Figure 6-35 presents a specific Spaceport example and shows how specific Operational Analysis inputs can provide analysis result outputs pertinent to the operations portion of the life cycle. It is noted that these simulations are conducted across the life cycle and updated periodically with the new data that is obtained as the project evolves.

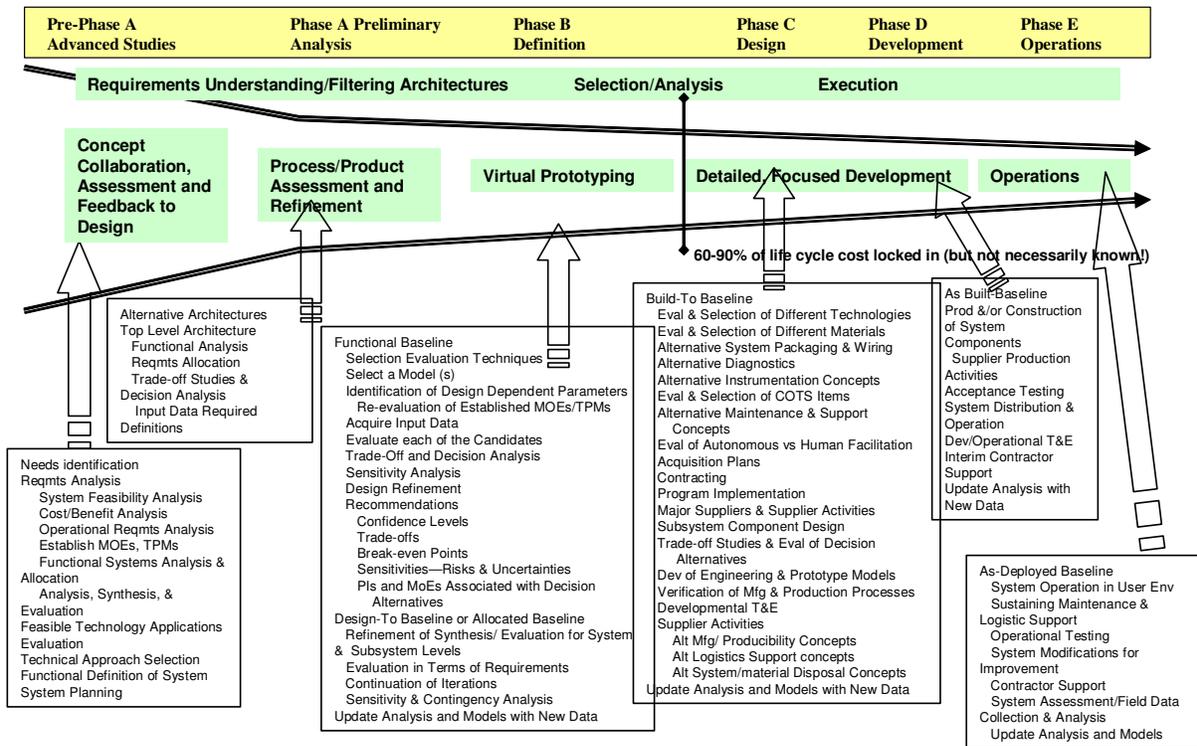


Figure 6-33 Systems Analysis across the Life Cycle

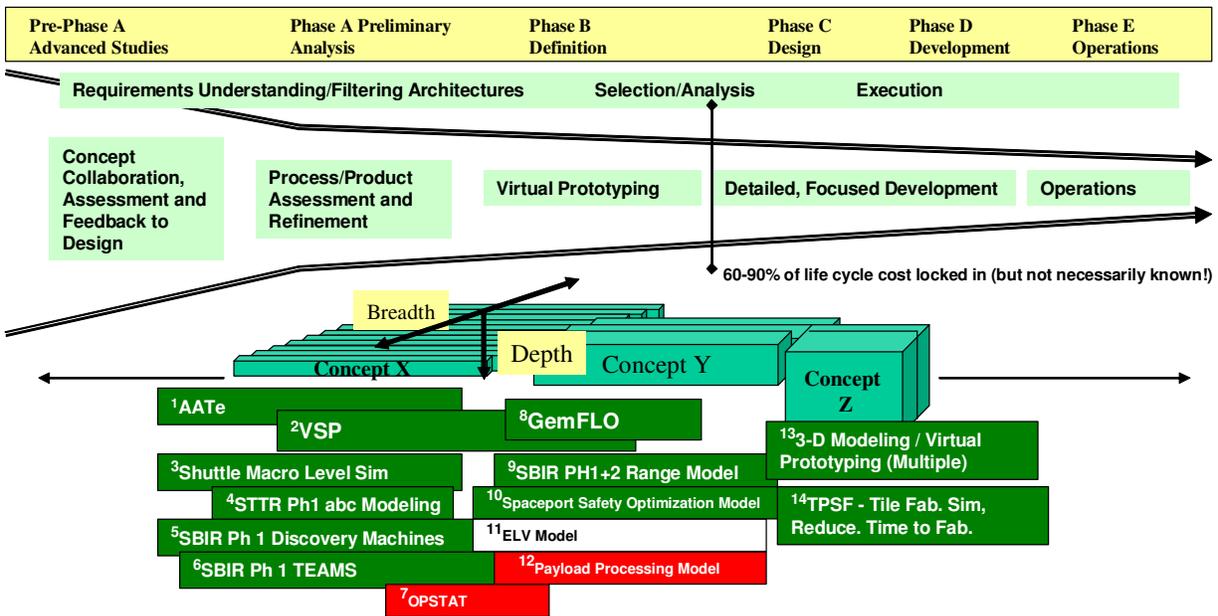


Figure 6-34 Operations Modeling and Simulation Capabilities

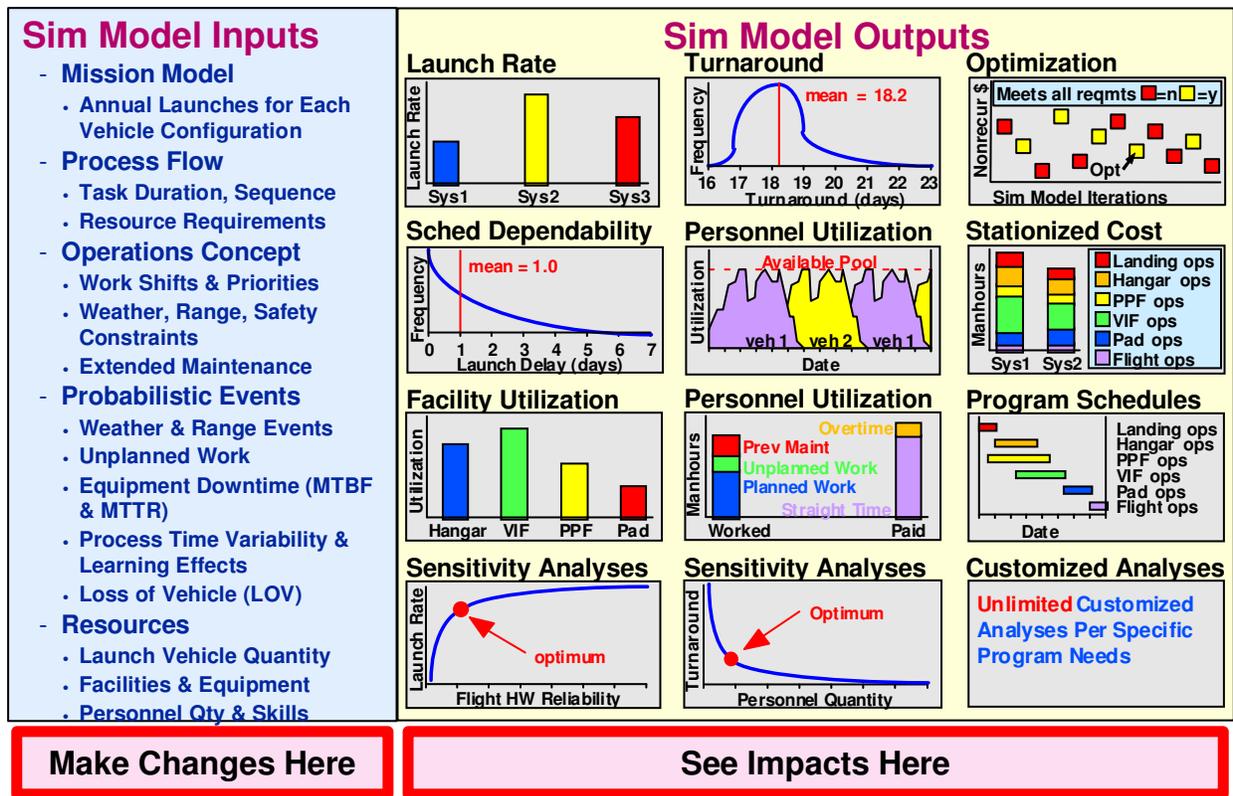


Figure 6-35 Simulation Model Analysis Techniques

During the early life-cycle phases, inputs should include a plan for collecting the quantitative and qualitative data necessary to manage contracts and improve processes and products as the project evolves. This plan should indicate the type of data necessary to determine the cause of problems, non-conformances, and anomalies, and propose corrective action to prevent recurrence. This closed-loop plan involving identification, resolution, and recurrence control system is critical to producing actual reliability that approaches predicted reliability. It should indicate the information technology infrastructure and database capabilities to provide data sorting, data mining, data analysis, and precursor management. Management of problems, nonconformances, and anomalies should begin with data collection, should be a major part of technical assessment, and should provide critical information for decision analysis.

### Functional Analysis Techniques

Functional analysis is the process of identifying, describing, and relating the functions a system must perform in order to fulfill its goals and objectives. Functional analysis is described in detail in the design solution section of this handbook.

#### 6.8.2.2 Trade Studies

The trade study process is a critical part of the systems engineering. This section discusses the steps of the process in greater detail. Trade studies help to define the emerging system at each

level of resolution. One key message of this section is that to be effective, the process requires the participation of many skills and a unity of effort to move toward an optimum system design.

Figure 6-36 shows the trade study process in simplest terms, beginning with the step of defining the system's goals and objectives, and identifying the constraints it must meet. In the early phases of the project life cycle, the goals, objectives, and constraints are usually stated in general operational terms. In later phases of the project life cycle, when the architecture and, perhaps, some aspects of the design have already been decided, the goals and objectives may be stated as performance requirements that a segment or subsystem must meet.

At each level of system resolution, the systems engineer needs to understand the full implications of the goals, objectives, and constraints in order to formulate an appropriate system solution. This step is accomplished by performing a functional analysis. Functional analysis is described in detail in the design solution section of this handbook.

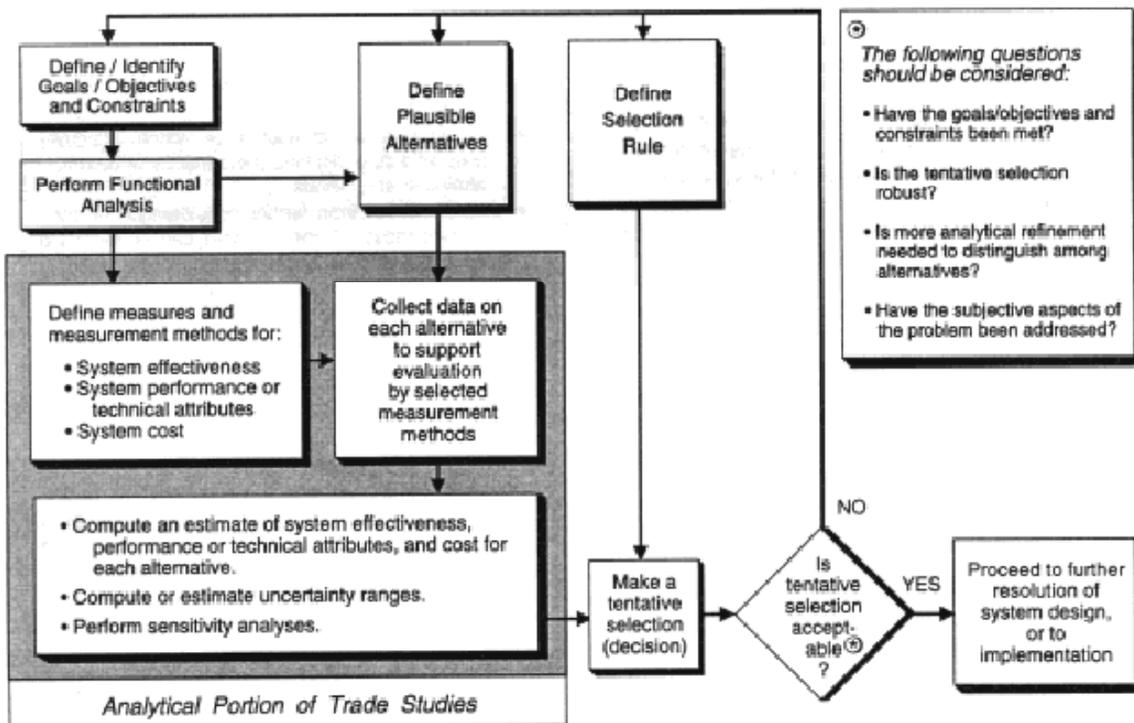


Figure 6-36 Trade Study Process

Closely related to defining the goals and objectives, and performing a functional analysis, is the step of defining the measures and measurement methods for system effectiveness (when this is practical), system performance or technical attributes, and system cost. (These variables are collectively called outcome variables, in keeping with the discussion in section 2.3. Some systems engineering books refer to these variables as decision criteria, but this term should not be confused with selection rule, described below. Sections 5.2 and 5.3 discuss the concepts of system cost and system effectiveness, respectively, in greater detail.) This step begins the

analytical portion of the trade study process, since it suggests the involvement of those familiar with quantitative methods.

For each measure, it is important to address the question of how that quantitative measure will be computed—that is, which measurement method is to be used. One reason for doing this is that this step then explicitly identifies those variables that are important in meeting the system’s goals and objectives.

Evaluating the likely outcomes of various alternatives in terms of system effectiveness, the underlying performance or technical attributes, and cost before actual fabrication and/or programming usually requires the use of a mathematical model or series of models of the system. So a second reason for specifying the measurement methods is that the necessary models can be identified.

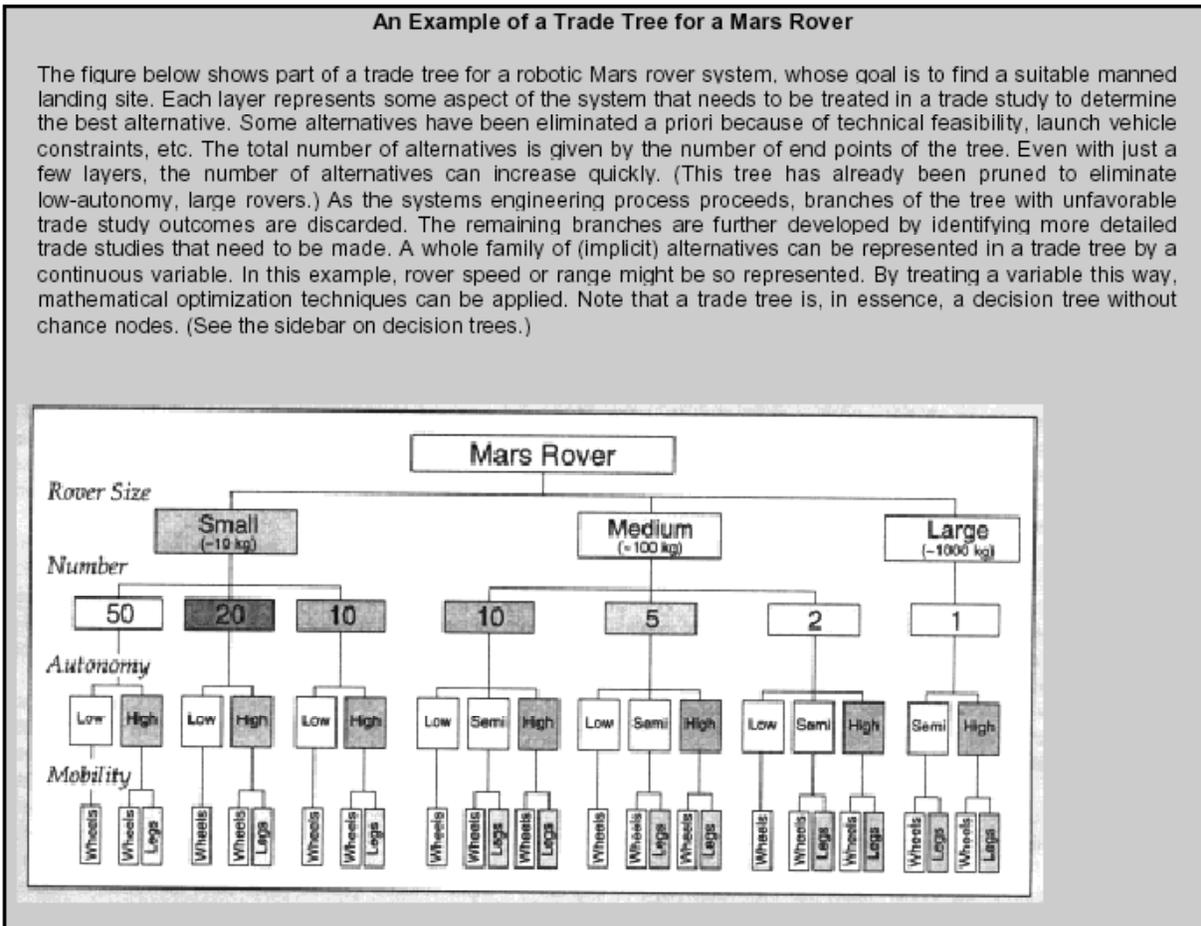
Sometimes these models are already available from previous projects of a similar nature; other times, they need to be developed. In the latter case, defining the measurement methods should trigger the necessary system modeling activities. Since the development of new models can take a considerable amount of time and effort, early identification is needed to ensure they will be ready for formal use in trade studies. Defining the selection rule is the step of explicitly determining how the outcome variables will be used to make a (tentative) selection of the preferred alternative. As an example, a selection rule may be to choose the alternative with the highest estimated system effectiveness that costs less than  $x$  dollars (with some given probability), meets safety requirements, and possibly meets other political or schedule constraints. Defining the selection rule is essentially deciding how the selection is to be made. This step is independent from the actual measurement of system effectiveness, system performance or technical attributes, and system cost.

Many different selection rules are possible. The selection rule in a particular trade study may depend on the context in which the trade study is being conducted—in particular, what level of system design resolution is being addressed. At each level of the system design, the selection rule generally should be chosen only after some guidance from the next higher level. The selection rule for trade studies at lower levels of the system design should be in consonance with the higher-level selection rule.

Defining plausible alternatives is the step of creating some alternatives that can potentially achieve the goals and objectives of the system. This step depends on understanding (to an appropriately detailed level) the system’s functional requirements and operational concept. Running an alternative through an operational time line or reference mission is a useful way of determining whether it can plausibly fulfill these requirements. (Sometimes it is necessary to create separate behavioral models to determine how the system reacts when a certain stimulus or control is applied, or a certain environment is encountered. This provides insights into whether it can plausibly fulfill time-critical and safety requirements.) Defining plausible alternatives also requires an understanding of the technologies available, or potentially available, at the time the system is needed. Each plausible alternative should be documented qualitatively in a description sheet. The format of the description sheet should, at a minimum, clarify the allocation of required system functions to that alternative’s lower-level architectural or design components (e.g., subsystems).

One way to represent the trade study alternatives under consideration is by a trade tree.

During Phase A trade studies, the trade tree should contain a number of alternative high-level system architectures to avoid a premature focus on a single one. As the systems engineering process proceeds, branches of the trade tree containing unattractive alternatives will be “pruned,” and greater detail in terms of system design will be added to those branches that merit further attention. The process of pruning unattractive early alternatives is sometimes known as doing “killer trades.” (See Figure 6-37.)



**Figure 6-37 Trade Tree**

Given a set of plausible alternatives, the next step is to collect data on each to support the evaluation of the measures by the selected measurement methods. If models are to be used to calculate some of these measures, then obtaining the model inputs provides some impetus and direction to the data collection activity. By providing data, engineers in such disciplines as reliability, maintainability, producibility, integrated logistics, software, testing, operations, and costing have an important supporting role in trade studies. The data collection activity, however, should be orchestrated by the systems engineer. The results of this step should be a quantitative description of each alternative to accompany the qualitative.

Test results on each alternative can be especially useful. Early in the systems engineering process, performance and technical attributes are generally uncertain and must be estimated. Data from breadboard and brassboard testbeds can provide additional confidence that the range of values used as model inputs is correct. Such confidence is also enhanced by drawing on data collected on related previously developed systems.

The next step in the trade study process is to quantify the outcome variables by computing estimates of system effectiveness, its underlying system performance or technical attributes, and system cost. If the needed data have been collected, and the measurement methods (for example, models) are in place, then this step is, in theory, mechanical. In practice, considerable skill is often needed to get meaningful results.

In an ideal world, all input values would be precisely known, and models would perfectly predict outcome variables. This not being the case, the systems engineer should supplement point estimates of the outcome variables for each alternative with computed or estimated uncertainty ranges. For each uncertain key input, a range of values should be estimated. Using this range of input values, the sensitivity of the outcome variables can be gauged, and their uncertainty ranges calculated. The systems engineer may be able to obtain meaningful probability distributions for the outcome variables using Monte Carlo simulation, but when this is not feasible, the systems engineer must be content with only ranges and sensitivities. Please see the Risk-Based Decision Analysis Process in section 5.8.4 for more information on uncertainty.

This essentially completes the analytical portion of the trade study process. The next steps can be described as the judgmental portion. Combining the selection rule with the results of the analytical activity should enable the systems engineer to array the alternatives from most preferred to least, in essence making a tentative selection.

This tentative selection should not be accepted blindly. In most trade studies, there is a need to subject the results to a “reality check” by considering a number of questions. Have the goals, objectives, and constraints truly been met? Is the tentative selection heavily dependent on a particular set of input values to the measurement methods, or does it hold up under a range of reasonable input values? (In the latter case, the tentative selection is said to be robust.) Are there sufficient data to back up the tentative selection? Are the measurement methods sufficiently discriminating to be sure that the tentative selection is really better than other alternatives? Have the subjective aspects of the problem been fully addressed?

If the answers support the tentative selection, then the systems engineer can have greater confidence in a recommendation to proceed to a further resolution of the system design, or to the implementation of that design. The estimates of system effectiveness, its underlying performance or technical attributes, and system cost generated during the trade study process serve as inputs to that further resolution. The analytical portion of the trade study process often provide the means to quantify the performance or technical (and cost) attributes that the system’s lower levels must meet. These can be formalized as performance requirements.

If the reality check is not met, the trade study process returns to one or more earlier steps. This iteration may result in a change in the goals, objectives, and constraints, a new alternative, or a change in the selection rule, based on the new information generated during the trade study. The

reality check may, at times, lead instead to a decision to first improve the measures and measurement methods (e.g., models) used in evaluating the alternatives, and then to repeat the analytical portion of the trade study process.

### **Controlling the Trade Study Process**

There are a number of mechanisms for controlling the trade study process. The most important one is the Systems Engineering Management Plan (SEMP). The SEMP specifies the major trade studies that are to be performed during each phase of the project life cycle. It should also spell out the general contents of trade study reports, which form part of the decision support packages (i.e., documentation submitted in conjunction with formal reviews and change requests).

A second mechanism for controlling the trade study process is the selection of the study team leaders and members. Because doing trade studies is part art and part science, the composition and experience of the teams is an important determinant of the study's ultimate usefulness. A useful technique to avoid premature focus on a specific technical designs is to include in the study team individuals with differing technology backgrounds.

Another mechanism is limiting the number of alternatives that are to be carried through the study. This number is usually determined by the time and resources available to do the study because the work required in defining additional alternatives and obtaining the necessary data on them can be considerable. However, focusing on too few or too similar alternatives defeats the purpose of the trade study process.

A fourth mechanism for controlling the trade study process can be exercised through the use (and misuse) of models. Lastly, the choice of the selection rule exerts a considerable influence on the results of the trade study process.

### **Trade Study Reports**

Trade study reports should be prepared for each trade study. At a minimum, each trade study report should identify:

- The system under analysis
- System goals and objectives (or requirements, as appropriate to the level of resolution), and constraints
- The measures and measurement methods (models) used
- All data sources used
- The alternatives chosen for analysis
- The computational results, including uncertainty ranges and sensitivity analyses performed
- The selection rule used
- The recommended alternative.

Trade study reports should be maintained as part of the system archives so as to ensure traceability of decisions made through the systems engineering process. Using a generally consistent format for these reports also makes it easier to review and assimilate them into the formal change control process.

#### *6.8.2.3 Cost-Benefit Analysis*

A cost benefit analysis (CBA) is performed to determine the advantage of one alternative over another in terms of equivalent cost or benefits. The CBA relies on the addition of positive factors and the subtraction of negative factors to determine a net result. Cost-benefit analysis maximizes

net benefits (benefits minus costs). A cost benefit analysis finds, quantifies, and adds all the positive factors. These are the benefits. Then it identifies, quantifies, and subtracts all the negatives, the costs. The difference between the two indicates whether the planned action is a preferred alternative. The real trick to doing a cost benefit analysis well is making sure you include all the costs and all the benefits and properly quantify them. A similar approach, used when a cost cap is imposed externally, is to maximize effectiveness for a given level of cost. Cost-effectiveness is a systematic quantitative method for comparing the costs of alternative means of achieving the same equivalent benefit for a specific objective. A project is cost-effective if, on the basis of life-cycle cost analysis of competing alternatives, it is determined to have the lowest costs expressed in present value terms for a given amount of benefits.

Cost-effectiveness analysis is appropriate whenever it is impractical to consider the dollar value of the benefits provided by the alternatives. This is the scenario whenever each alternative has the same life-cycle benefits expressed in monetary terms, or, each alternative has the same life-cycle affects, but dollar values cannot be assigned to their benefits. After determining the scope of the project on the basis of mission and other requirements, and having identified, quantified, and valued the costs and benefits of the alternatives, the next step is to identify the least-cost or most cost-effective alternative to achieve the purpose of the project. A comparative analysis of the alternative options or designs is often required. This is illustrated in figure 2-18. In cases in which alternatives can be defined that deliver the same benefits, it is possible to estimate the equivalent rate between each alternative for comparison. Least-cost analysis aims at identifying the least-cost project option for meeting the technical requirements. Least-cost analysis involves comparing the costs of the various technically feasible options and selecting the one with the lowest costs. Project options must be alternative ways of achieving the mission objectives. If differences in results or quality exist, a normalization procedure that takes the benefits of one option relative to another as a cost to the option that does not meet all of the mission objectives to ensure an equitable comparison. Procedures for the calculation and interpretation of the discounting factors should be made explicit, with the least-cost project being identified by comparing the total life-cycle costs of the project alternatives and calculating the equalizing factors for the difference in costs. The project with the highest equalizing factors for all comparisons is the least-cost alternative. Cost-effectiveness analysis also deals with alternative means of achieving mission requirements. However, the results may be estimated only indirectly. For example, different types of systems may be under consideration to obtain science data. The effectiveness of each alternative may be measured through obtaining science data through different methods. An example of a Cost-effectiveness analysis requires the increase in science data to be divided by the costs for each alternative. The most cost-effective method is the one that raises science data by a given amount for the least cost. If this method is chosen and applied to all similar alternatives, the same increase in science data can be obtained for the lowest cost. However, it should also be noted that the most cost effective method is not necessarily the most effective method of meeting mission objectives. Another method may be the most effective, but also cost a lot more, so it is not the most cost effective. The cost effectiveness ratios—the cost per unit increase in science data for each method—can be compared to see how much more it would cost to implement the most effective method. Which method is chosen for implementation then depends jointly on the desired mission objectives, and the extra cost involved in implementing the most effective method. There will be circumstances where project alternatives have more than one outcome. In order to assess the cost effectiveness of the different alternatives it is necessary to devise a testing system where the results for the different factors can be added

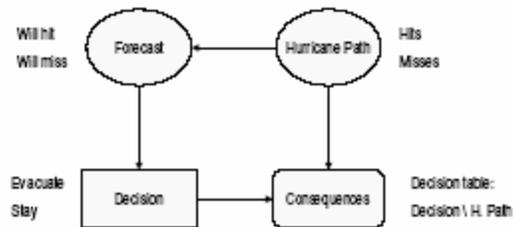
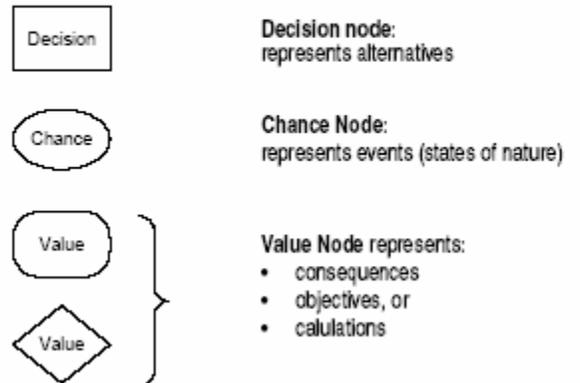
together. It is also necessary to decide on some weights for adding the different elements together, reflecting their importance in relation to the objectives of the project. Such a use of cost-effectiveness analysis is called weighted cost-effectiveness analysis. It introduces a subjective element, the weights, into the comparison of project alternatives, both to find the most cost-effective alternative and to identify the extra cost of implementing the most effective alternative.

#### 6.8.2.4 *Influence Diagrams*

An influence diagram (also called a decision network) is a compact graphical and mathematical representation of a decision state. ID was first developed in mid 70's within [http://en.wikipedia.org/wiki/Decision\\_analysis](http://en.wikipedia.org/wiki/Decision_analysis) the decision analysis community with an intuitive approach that is easy to understand, it is now adopted widely and becoming an alternative to decision tree which typically suffers from exponential growth in number of branches with each variable modeled. ID is directly applicable in team decision analysis since it allows incomplete sharing of information among team members to be modeled and solved explicitly. Its elements are:

- Decision nodes, indicating the decision inputs, and the items directly influenced by the decision outcome,
- Chance nodes, indicating factors that impact the chance outcome, and items influenced by the chance outcome,
- Value nodes, indicating factors that affect the value, and items influenced by the value, and
- Arrows, indicating the relationships among the elements.

## Elements of Influence Diagrams



**Figure 6-38 Influence Diagrams**

An influence diagram does not depict a strictly sequential process. Rather, it illustrates the decision process at a particular point in time, showing all of the elements important to the decision. The influence diagram for a particular model is not unique. The strength of influence diagrams is their ability to display the structure of a decision problem in a clear, compact form, useful both for communication and to help the analyst think clearly during problem formulation. An influence diagram can be transformed into a decision tree for quantification.

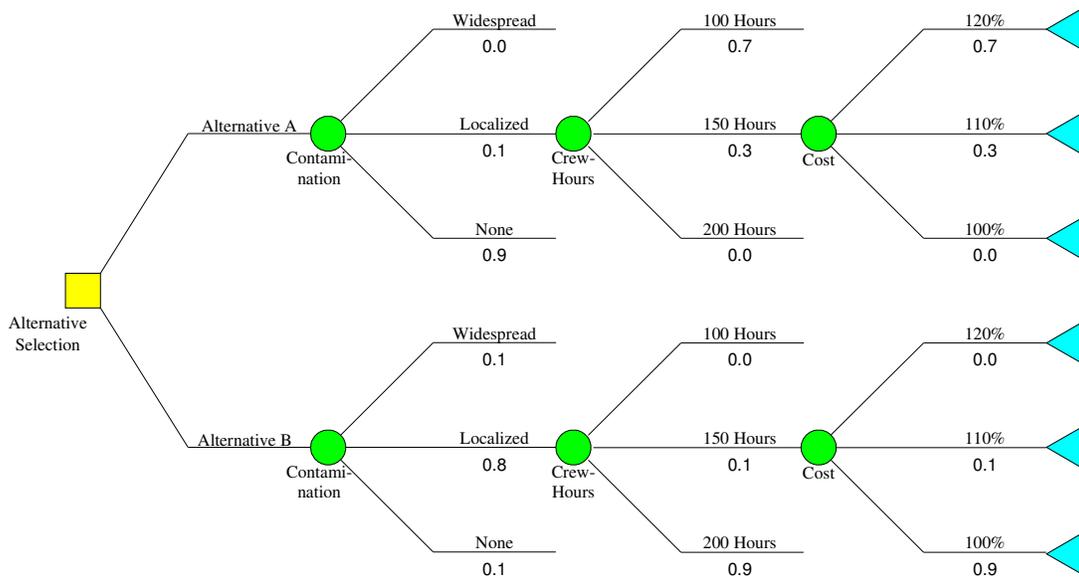
### 6.8.2.5 Decision Trees

Like the influence diagram, a decision tree portrays a decision model, but a decision tree is drawn from a point of view different from that of the influence diagram. The decision tree exhaustively works out the expected consequences of all decision alternatives by discretizing all “chance” nodes, and, based on this discretization, calculating and appropriately weighting all possible consequences of all alternatives. The preferred alternative is then identified by summing the appropriate outcome variables (MoE or expected utility) from the path end states.

A decision tree grows horizontally from left to right, with the trunk at the left. Typically, the possible alternatives initially available to the decision-maker stem from the trunk at the left. Moving across the tree, the decision-maker encounters branch points corresponding to probabilistic outcomes and perhaps additional decision nodes. Thus, the tree branches as one

reads from left to right across the tree. At the far right side of the decision tree, a vector of TPM scores is listed for each terminal branch, representing each combination of decision outcome and chance outcome. From the TPM scores, and the chosen selection rule, a preferred alternative is determined.

In even moderately complicated problems, decision trees can quickly become difficult to understand. A commonly employed strategy is to start with an equivalent influence diagram. This often aids in helping to understand the principal issues involved. Some software packages make it easy to develop an influence diagram and then, based on the influence diagram, automatically furnish a decision tree. The decision tree can be edited if this is desired. Calculations are typically based on the decision tree itself.



**Figure 6-39 Decision Tree**

### 6.8.2.6 Multi-Criteria Decision Analysis

Multi-criteria decision analysis (MCDA), is a method aimed at supporting decision makers who are faced with making numerous and conflicting evaluations. These techniques aim at highlighting the conflicts in alternatives and deriving a way to come to a compromise in a transparent process. For example, NASA may apply MCDA to help assess whether selection of one set of software tools for every NASA application is cost effective. MCDA involves a certain element of subjectiveness, the bias and position of the team implementing MCDA play a significant part in the accuracy and fairness of decisions. One of the MCDA methods is:

#### The Analytic Hierarchy Process

The analytic hierarchy process (AHP) was first developed and applied by Thomas Saaty. AHP is a multi-attribute methodology which provides a proven, effective means to deal with complex decision making and can assist with identifying and weighting selection criteria, analyzing the data collected for the criteria and expediting the decision-making process. Many different

problems can be investigated with the mathematical techniques of this approach. AHP helps capture both subjective and objective evaluation measures, providing a useful mechanism for checking the consistency of the evaluation measures and alternatives suggested by the team thus reducing bias in decision making. The analytic hierarchy process (AHP) is supported by pair-wise comparison techniques, and it supports the entire decision process. AHP is normally done in six steps:

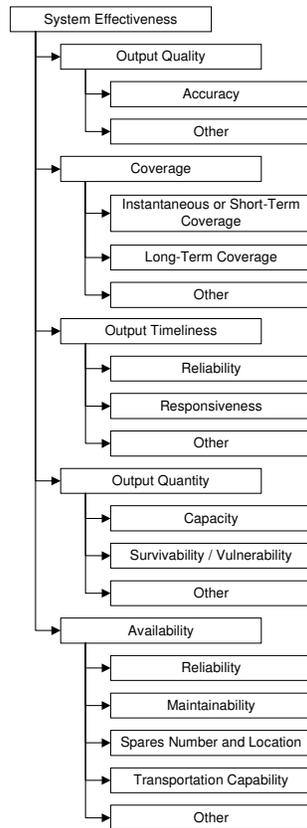
1. Describe in summary form the alternatives under consideration.
2. Develop a set of high-level objectives,
3. Decompose the high-level objective from general to specific to produce an Objectives Hierarchy,
4. Determine the relative importance of the evaluation objectives and attributes by assigning weights by engaging experts with structured process such as interviews or questionnaires,
5. Have each expert make pair-wise comparisons of the performance of each decision alternative with respect to a TPM. Repeat this for each TPM. Combine the results of these subjective evaluations mathematically using a process, or commonly available software tool, which rank the alternatives.
6. Iterate the questionnaire and AHP evaluation process until a consensus ranking of the alternative is achieved

If AHP is used only to produce the TPM weights, to be used in a PI or MoE calculation, then only the first four steps listed above are applicable.

With AHP, consensus may be achieved quickly, or several feedback rounds may be required. The feedback consists of reporting the computed ranking, for each evaluator and for the group, for each option, along with the reasons for differences in rankings, and identified areas of divergence. Experts may choose to change their judgments on TPM weights. At this point, divergent preferences can be targeted for more detailed study. AHP assumes the existence of an underlying preference vector, with magnitudes and directions that are revealed through the pair-wise comparisons. This is a powerful assumption, which may at best hold only for the participating experts. The ranking of the alternatives is the result of the experts' judgments and is not necessarily a reproducible result. For further information on AHP see references by Thomas Saaty.

### **Flexibility and Extensibility Attributes**

In some decision situations, the selection of a particular decision alternative will have implications for the long-term that are very difficult to model in the present. In such cases, it is useful to structure the problem as a series of linked decisions, with some decisions to be made in the near future and others to be made later, perhaps on the basis of information to be obtained in the meantime. There is value in delaying some decisions to the future, when additional information will be available. Some technology choices might foreclose certain opportunities that would be preserved by other choices.



**Figure 6-40 Examples of Lower-Level Performance Measures**

In these cases, it is desirable to consider attributes such as “flexibility” and “extensibility.” “Flexibility” refers to the ability to support more than one current application. “Extensibility” refers to the ability to be extended to other applications. For example, in choosing an architecture to support lunar exploration, one might consider extensibility to Mars missions. A technology choice that imposes a hard limit on mass that can be boosted into a particular orbit has less flexibility than a choice that is more easily adaptable to boost more. Explicitly adding “extensibility” and “flexibility” as attributes to be weighted and evaluated allows this issue to be addressed systematically. In such applications, “extensibility” and “flexibility” are being used as *surrogates* for certain future performance attributes. This idea is discussed more extensively by [Keeney] and [Keeney and McDaniels.]

#### 6.8.2.7 Utility Analysis

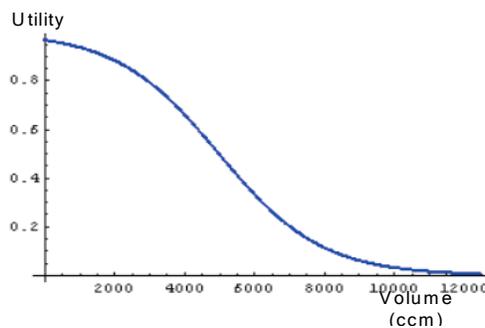
**Utility** is a measure of the relative value gained from an alternative. Given this measure, the team looks at increasing or decreasing utility, and thereby explain alternative decisions in terms of attempts to increase one's utility. The theoretical unit of measurement for utility is the util.

The utility function maps the range of the TPM into the range of associated utilities, capturing the decision-maker’s preferences and risk attitude. One could imagine simply mapping the indicated range of values linearly onto the interval [0,1] on the utility axis, but in general, this

would not capture the decision-maker's preferences. The decision-maker's attitude toward risk causes the curve to be convex (risk prone), concave (risk averse), or even some of each.

The detailed process of eliciting the utility function from the decision-maker is described in [Keeney and Raiffa]. The utility function directly reflects the decision-maker's attitude towards risk. When ranking alternatives on the basis of utility, a risk-averse decision-maker will rank an alternative with highly uncertain performance below an alternative having the same expected performance but less uncertainty. The opposite outcome would result for a risk-prone decision-maker. When the individual TPM utility functions have been assessed, it is important to check the result for consistency with the decision-maker's actual preferences (e.g., is it true that intermediate values of  $TPM_1$  and  $TPM_2$  are preferred to a high value of  $TPM_1$  and a low value of  $TPM_2$ ?).

An example of a utility function for the TPM "Volume" is shown in **Error! Reference source not found.** This measure was developed in the context of design of sensors for a space mission. Volume was a precious commodity. The implication of the graph is that low volume is good, large volume is bad, and the decision-maker would prefer a design alternative with a very well-determined volume of a few thousand cc's to an alternative with the same expected volume but large uncertainty.



**Figure 6-41 Utility Function for a "Volume" Performance Measure**

Value functions can take the place of utility functions when a formal treatment of risk attitude is unnecessary. They appear very similar to utility functions, but have one important difference. Value functions do not consider the risk attitude of the decision-maker. They do not reflect how the decision-maker compares certain outcomes to uncertain outcomes.

The assessment of a TPM's value function is relatively straightforward. The "best" end of the TPM's range is assigned a value of 1. The "worst" is assigned a value of 0. The decision-maker makes direct assessments of the value of intermediate points to establish the preference structure in the space of possible TPM values. This process is discussed in [Keeney and Raiffa]. The utility function can be treated as a value function, but the value function is not necessarily a utility function.

One way to rank alternatives is to use a multi-attribute, utility theory (MAUT) approach. Within this approach, the "expected utility" of each alternative is quantified, and alternatives are ranked based on their expected utilities. Sometimes the expected utility is referred to as performance

index (PI). An important benefit of applying this method is that it is the best way to deal with significant uncertainties when the decision-maker is not risk-neutral. Probabilistic methods are used to treat uncertainties. A downside of applying this method is the need to quantify the decision-maker's risk attitudes. Top-level system architecture decisions are natural examples of appropriate applications of MAUT.

#### 6.8.2.8 *Risk-Informed Decision Analysis Process Example*

##### **Introduction**

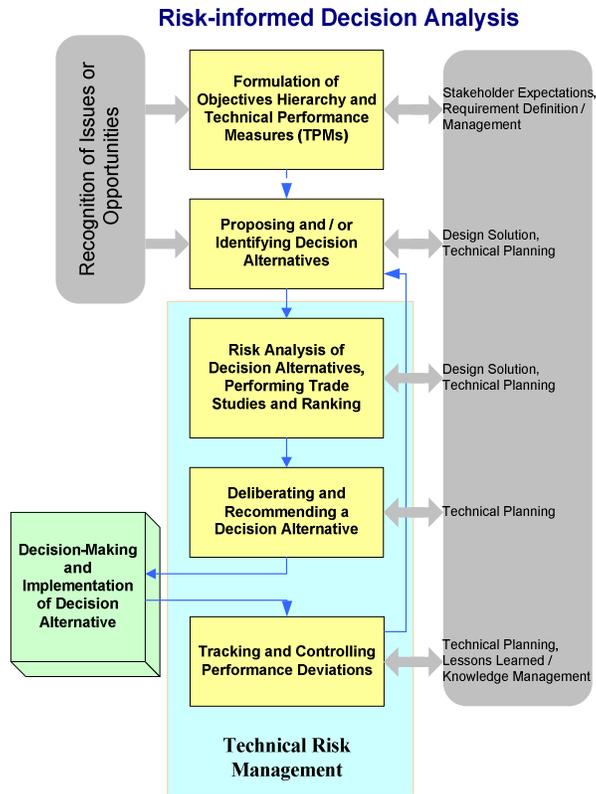
A decision matrix works for many decisions, but the decision matrix may not scale up to very complex decisions or risky decisions. For some decisions, sometimes a tool is needed to handle the complexity. The following section describe detailed decision analysis process that can be used to support a risk-based decision.

In practice, decisions are made in many different ways. Simple approaches may be useful, but it is important to recognize their limitations and upgrade to better analysis when this is warranted. Some decision-makers, when faced with uncertainty in an important quantity, determine a best estimate for that quantity, and then reason as if the best estimate were correct. This might be called the “take-your-best-shot” approach. Unfortunately, when the stakes are high, and uncertainty is significant, this best-shot approach may lead to poor decisions.

The following steps are a Risk-informed Decision Analysis Process:

1. Formulation of the Objectives Hierarchy, TPMs
2. Proposing and Identifying Decision Alternatives. Alternatives from this process are combined with the alternatives identified in the other systems engineering processes to include the design solution process, but also include verification and validation as well as production.
3. Risk Analysis of Decision Alternatives and ranking of alternatives
4. Deliberation and Recommendation of Decision Alternative
5. Follow-up tracking of the implementation of the decision

These steps support good decisions by focusing first on objectives, next on developing decision alternatives with those objectives clearly in mind and/or using decision alternatives that have been developed under other systems engineering processes. The later steps of the Decision Analysis process interrelate heavily with the Technical Risk Management process, as indicated in the figure above. These steps include risk analysis of the decision alternatives, deliberation informed by risk analysis results, and recommendation of a decision alternative to the decision-maker. Implementation of the decision is also important.



**Figure 6-42 Risk-informed Decision Analysis Process**

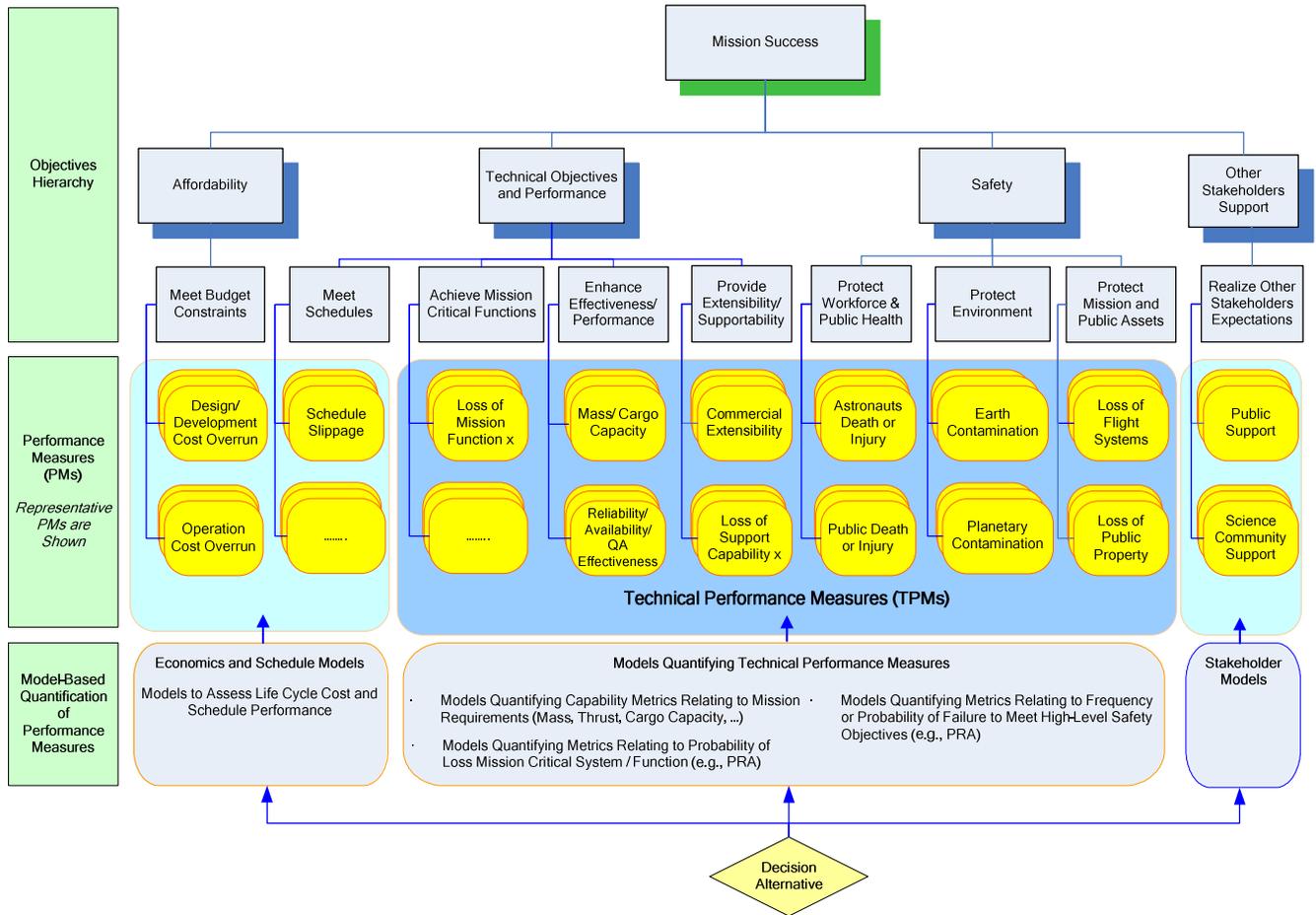
In practice, decisions are made in many different ways. Simple approaches may be useful, but it is important to recognize their limitations and upgrade to better analysis when this is warranted. Some decision-makers, when faced with uncertainty in an important quantity, determine a best estimate for that quantity, and then reason as if the best estimate were correct. This might be called the “take-your-best-shot” approach. Unfortunately, when the stakes are high, and uncertainty is significant, this best-shot approach may lead to poor decisions.

### Objectives Hierarchy/TPMs

As shown in Figure 6-42, risk-informed decision analysis starts with formulation of the objective hierarchy. Using this hierarchy, TPMs are formulated in such a way as to quantify performance of a decision with respect to the program objectives. The TPMs should have the following characteristics:

- They can support ranking of major decision alternatives,
- They are sufficiently detailed to be used directly in the risk management process,
- They are preferentially independent. This means that they contribute in distinct ways to the program goal. This property helps to ensure that alternatives are ranked appropriately.

An example of an objectives hierarchy is shown on Figure 6-43. Details will vary from program to program, but a construct like Figure 6-43 is behind the program-specific objectives hierarchy.



**Figure 6-43 An Example of an Objectives Hierarchy**

The TPMs in this figure are meant to be generically important for many missions. Depending on the mission, these TPMs are further subdivided to the point where they can be objectively measured. Not all TPMs can be measured directly. For example, safety-related TPMs are defined in terms of the probability of a consequence type of a specific magnitude (e.g., probability of any general public deaths or injuries) or the expected magnitude of a consequence type (e.g., the number of public deaths or injuries). Probability of Loss of Mission P(LOM) and Loss of Crew P(LOC) are two particularly important safety-related TPMs for manned space missions. Because an actuarial basis does not suffice for prediction of these probabilities, modeling will be needed to quantify them.

## **7.0 Special Topics**

The topics below are of special interest for enhancing the performance of the systems engineering process or constitute special considerations in the performance of systems engineering. The first section elucidates the process of how the systems engineering principles need to be applied to contracting and contractors who implement NASA processes and create NASA products. Applying Lessons Learned enhances the efficiency of the present with the wisdom of the past. Protecting the environment and the Nation's space assets are important considerations in the design and development of requirements and designs. Integrated design can enhance the efficiency and effectiveness of the design process.

## 7.1 Engineering with Contracts

### 7.1.1 Introduction, Purpose, and Scope

Historically, most successful NASA projects have depended on effectively blending project management, systems engineering, and technical expertise among NASA, contractors, and third parties. Underlying these successes are a variety of agreements (e.g., contract, memorandum of understanding, grant, cooperative agreement) between NASA organizations or between NASA and other Government agencies, government organizations, companies, universities, research laboratories, and so on. To simplify the discussions, the term “contract” is used to encompass these agreements.

This section focuses on the engineering activities pertinent to awarding a contract, managing contract performance, and completing a contract. Procurement and legal processes are covered extensively elsewhere [reference]. However, interfaces to the procurement process will be covered, since the engineering technical team plays a key role in development and evaluation of contract documentation.

Contractors and third parties perform activities that supplement (or substitute for) the NASA project technical team accomplishment of the common technical process activities and requirements. Since contractors might be involved in any part of the systems engineering life cycle, the NASA project technical team needs to know how to prepare for, perform, and complete surveillance of technical activities that are allocated to contractors.

### 7.1.2 Acquisition Strategy

Creating an acquisition strategy for a project is a collaborative effort among several NASA HQ offices that leads to approval for project execution. The program and project offices characterize the acquisition strategy in sufficient detail to identify the contracts needed to execute the strategy. Awarding contracts at the project level occurs in the context of the overall program acquisition strategy.

While this section pertains to projects where the decision has been made to have a contractor implement a portion of the project, it is important to remember that the choice between “making” a product inhouse by NASA or “buying” it from a contractor is one of the most crucial decisions in systems development. (See Section 5.1, Product Implementation.) Questions that should be considered in the “make/buy” decision include the following.

- Is the desired system a development item or more off the shelf?
- What is the relative experience between NASA and contractors?
- What are the relative importance of risk, cost, schedule and performance?
- Is there a desire to maintain an “in house” capability?

As soon as it is clear that a contract will be needed to obtain a system or service, the responsible project manager should contact the local procurement office. The contracting officer will assign a contract specialist to navigate the numerous regulatory requirements that affect NASA procurements and guide the development of contract documentation needed to award a contract. The contract specialist engages the local legal office as needed.

### *7.1.2.1 Develop an Acquisition Strategy*

The project manager, assisted by the local procurement office, first develops a project acquisition strategy or verifies the one provided. The acquisition strategy provides a business and technical management outline for planning, directing, and managing a project and obtaining products and services via contract.

In some cases, it may be appropriate to probe outside sources in order to gather sufficient information to formulate an acquisition strategy. This can be done by issuing a request for information (RFI) to industry and other parties that may have interest in potential future contracts. An RFI is a way to obtain information about technology maturity, technical challenges, capabilities, price and delivery considerations, and other market information that can influence strategy decisions.

The acquisition strategy [reference—CMMI-Acq] includes:

- Objectives of the acquisition—capabilities to be provided, major milestones;
- Acquisition approach—single step or evolutionary (incremental), single or multiple suppliers/contracts, competition or sole source, funding source(s), phases, system integration, COTS products;
- Business considerations—constraints (e.g., funding, schedule), availability of assets and technologies, applicability of commercial items versus internal technical product development;
- Risk management of acquired products or services—major risks and risk sharing with the supplier;
- Contract types—performance-based or level of effort, fixed-price or cost reimbursable;
- Contract elements—incentives, performance parameters, rationale for decisions on contract type; and
- Product support strategy—oversight of delivered system, maintenance, and improvements.

The technical team gathers data to facilitate the decisionmaking process regarding the above items. The technical team knows about issues with the acquisition approach, determining availability of assets and technologies, applicability of commercial items, issues with system integration, and details of product support. Similarly, the technical team provides corporate knowledge to identify and evaluate risks of acquiring the desired product, especially regarding the proposed contract type and particular contract elements.

### *7.1.2.2 Acquisition Life Cycle*

Contract activities are part of the broader acquisition life cycle, which comprises the phases solicitation, source selection, contract monitoring, and acceptance (See Figure 7-1.). The acquisition life cycle overlaps and interfaces with the systems engineering processes in the

project life cycle. Acquisition planning focuses on technical planning when a particular contract (or purchase) is required. (See Section 6.1, Technical Planning.) In the figure below, requirements development corresponds to the Technical Requirements Definition Process in the systems engineering engine. (See Figure 2-2.) The next four phases—solicitation, source selection, contract monitoring, and acceptance—are the phases of the contract activities. Transition to operations and maintenance represents activities performed to transition acquired products to the organization(s) responsible for operating and maintaining them (which could be contractor(s)). Acquisition management refers to project management activities that are performed throughout the acquisition life cycle by the acquiring organization.

Acquisition Management						
Acquisition Planning	Requirements Development	Solicitation	Source Selection	Contract Monitoring	Acceptance	Transition to O&M

**Figure 7-1 Acquisition Life Cycle**

O&M = Operations and Maintenance

### 7.1.2.3 NASA Responsibility for Systems Engineering

The technical team is responsible for systems engineering throughout the acquisition life cycle. The technical team contributes heavily to systems engineering decisions and results, whatever the acquisition strategy, for any combination of suppliers, contractors, and subcontractors. The technical team is responsible for systems engineering whether the acquisition strategy calls for the technical team, or a prime contractor, or some combination to perform system integration and testing of products from multiple sources.

This section provides specific guidance on how to assign responsibility when translating the technical processes onto a contract. Generally, the Technical Planning, Interface Management, Technical Risk Management, Configuration Management, Technical Data Management, Technical Assessment, and Decision Analysis processes should be implemented throughout the project by both the NASA team and the contractor. Stakeholder Expectations Definition, Technical Requirements Definition, Logical Decomposition, Design Solution Definition, Product Implementation and Integration, Product Verification and Validation, Product Transition, and Requirements Management processes are implemented by NASA or the contractor depending upon the level of the product decomposition. The following table provides guidance on how to implement the 17 processes from NPR 7123.1. The third column in the table also provides an example of implementing the processes for a sample project.

**Table 7-1 Applying the Technical Processes on Contract**

<b>NPR 7123.1 Process and Number</b>		<b>General Guidance on How to Distinguish Who—Contractor or NASA—Implements the Process</b>	<b>Example: A science mission where a contractor is building the spacecraft, NASA assigns Government-furnished property (GFP) instruments to the contractor, and NASA operates the mission.</b>
1	The Center Directors or designees establish and maintain a process to include activities, requirements, guidelines, and documentation for the definition of stakeholder expectations for the applicable WBS model.	If stakeholders are at the contractor, then the contractor should have responsibility and vice versa.	Stakeholders for the mission/project are within NASA; stakeholders for the spacecraft power subsystem are mostly at the contractor.
2	The Center Directors or designees establish and maintain a process to include activities, requirements, guidelines, and documentation for definition of the technical requirements from the set of agreed upon stakeholder expectations for the applicable WBS model.	Assignment of responsibility follows the stakeholders, e.g., if stakeholders are at the contractor, then requirements are developed by the contractor and vice versa.	NASA develops the high level requirements, and the contractor develops the requirements for the power subsystem.
3	The Center Directors or designees establish and maintain a process to include activities, requirements, guidelines, and documentation for logical decomposition of the validated technical requirements of the applicable WBS.	Follows the requirements, e.g., if requirements are developed at the contractor, then the decomposition of those requirements is implemented by the contractor and vice versa.	NASA performs the decomposition of the high level requirements, and the contractor performs the decomposition of the power subsystem requirements.
4	The Center Directors or designees establish and maintain a process to include activities, requirements, guidelines, and documentation for designing product solution definitions within the applicable WBS model that satisfy the derived technical requirements.	Follows the requirements, e.g., if requirements are developed at the contractor, then the design of the product solution is implemented by the contractor and vice versa.	NASA designs the mission/project, and the contractor designs the power subsystem.
5	The Center Directors or designees establish and maintain a process to include activities, requirements, guidelines, and documentation for implementation of a design solution definition by making, buying, or reusing an end product of the applicable WBS model.	Follows the design, e.g., if the design is developed at the contractor, then the implementation of the design is performed by the contractor and vice versa.	NASA implements (and retains responsibility for) the design for the mission/project, and the contractor does the same for the power subsystem.
6	The Center Directors or designees establish and maintain a process to include activities, requirements, guidelines, and documentation for the integration of lower level products into an end product of the applicable WBS model in accordance with its design solution definition.	Follows the design, e.g., if the design is developed at the contractor, then the integration of the design elements is performed by the contractor and vice versa.	NASA integrates the design for the mission/project, and the contractor does the same for the power subsystem.
7	The Center Directors or designees establish and maintain a process to include activities, requirements, guidelines, and documentation for verification of end products generated by the product implementation process or product integration process against their design solution definitions.	Follows the product integration, e.g., if the product integration is implemented at the contractor, then the verification of the product is performed by the contractor and vice versa.	NASA verifies the mission/project, and the contractor does the same for the power subsystem.
8	The Center Directors or designees establish and maintain a process to include activities, requirements, guidelines, and documentation for validation of end products generated by the product implementation process or product integration process against their stakeholder expectations.	Follows the product integration, e.g., if the product integration is implemented at the contractor, then the validation of the product is performed by the contractor and vice versa.	NASA validates the mission/project, and the contractor does the same for the power subsystem.

<b>NPR 7123.1 Process and Number</b>		<b>General Guidance on How to Distinguish Who—Contractor or NASA—Implements the Process</b>	<b>Example: A science mission where a contractor is building the spacecraft, NASA assigns Government-furnished property (GFP) instruments to the contractor, and NASA operates the mission.</b>
9	The Center Directors or designees establish and maintain a process to include activities, requirements, guidelines, and documentation for transitioning end products to the next higher level WBS-model customer or user.	Follows the product V&V, e.g., if the product V&V is implemented at the contractor, then the transition of the product is performed by the contractor and vice versa.	NASA transitions the mission/project to operations, and the contractor transitions the power subsystem to the spacecraft level.
10	The Center Directors or designees establish and maintain a process to include activities, requirements, guidelines, and documentation for planning the technical effort.	Assuming both NASA and the contractor have technical work to perform, then both NASA and the contractor need to plan their respective technical efforts.	NASA would plan the technical effort associated with the GFP instruments and the launch and operations of the spacecraft, and the contractor would plan the technical effort associated with the design, build, V&V, and delivery and operations of the power subsystem.
11	The Center Directors or designees establish and maintain a process to include activities, requirements, guidelines, and documentation for <b>management of requirements defined and baselined during the application of the system design processes.</b>	Follows process #2	
12	The Center Directors or designees establish and maintain a process to include activities, requirements, guidelines, and documentation for <b>management of the interfaces defined and generated during the application of the system design processes.</b>	Interfaces should be managed one level above the elements being interfaced.	The interface from the spacecraft to the project ground system would be managed by NASA, while the power subsystem to attitude control subsystem interface would be managed by the contractor.
13	The Center Directors or designees establish and maintain a process to include activities, requirements, guidelines, and documentation for <b>management of the technical risk identified during the technical effort.</b> (NPR 8000.4, <i>Risk Management Procedural Requirements</i> , is to be used as a source document for defining this process, and NPR 8705.5, <i>Probabilistic Risk Assessment (PRA) Procedures for NASA Programs and Projects</i> , provides one means of identifying and assessing technical risk.)	Technical risk management is a process that needs to be implemented by both NASA and the contractor. All elements of the project need to identify their risks and participate in the project risk management process. Deciding which risks to mitigate, when, at what cost is generally a function of NASA project management.	NASA project management should create a project approach to risk management that includes participation from the contractor. Risks identified throughout the project down to the power subsystem level and below should be identified and reported to NASA for possible mitigation.
14	The Center Directors or designees establish and maintain a process to include activities, requirements, guidelines, and documentation for <b>configuration management.</b>	Like risk management, configuration management is a process that should be implemented throughout the project by both the NASA and contractor teams.	NASA project management should create a project approach to configuration management that includes participation from the contractor. The contractor's internal CM process will have to be integrated with the NASA approach. CM needs to be implemented throughout the project down to the power subsystem level and below.
15	The Center Directors or designees establish and maintain a process to include activities, requirements, guidelines, and documentation for <b>management of the technical data generated and used in the technical effort.</b>	Like risk management and configuration management, technical data management is a process that should be implemented throughout the project by both the NASA and contractor teams.	NASA project management should create a project approach to technical data management that includes participation from the contractor. The contractor's internal technical data process will have to be integrated with the NASA approach. Management of technical data needs to be implemented throughout the project down to the power subsystem level and below.

NPR 7123.1 Process and Number		General Guidance on How to Distinguish Who—Contractor or NASA—Implements the Process	Example: A science mission where a contractor is building the spacecraft, NASA assigns Government-furnished property (GFP) instruments to the contractor, and NASA operates the mission.
16	The Center Directors or designees establish and maintain a process to include activities, requirements, guidelines, and documentation for <b>making assessments of the progress of planned technical effort and progress toward requirements satisfaction.</b>	Assessing progress is a process that should be implemented throughout the project by both the NASA and contractor teams.	NASA project management should create a project approach to assessing progress that includes participation from the contractor. Typically this would be the Project Review Plan. The contractor's internal review process will have to be integrated with the NASA approach. Technical reviews need to be implemented throughout the project down to the power subsystem level and below.
17	The Center Directors or designees establish and maintain a process to include activities, requirements, guidelines, and documentation for <b>making technical decisions.</b>	Clearly technical decisions are made throughout the project both by NASA and contractor personnel. Certain types of decisions or decisions on certain topics may best be made by either NASA or the contractor depending upon the Center's processes and the type of project.	For this example, decisions affecting high level requirements or mission success would be made by NASA and those at the lower level, e.g., the power subsystem that did not affect mission success would be made by the contractor.

### 7.1.3 Prior to Contract Award

#### 7.1.3.1 Acquisition Planning

Based on the acquisition strategy, the technical team needs to plan acquisitions and document the plan in developing the SEMP. The SEMP covers the technical team's involvement in the periods before contract award, during contract performance, and upon contract completion. Included in acquisition planning are solicitation preparation, source selection activities, contract phase-in, monitoring contractor performance, acceptance of deliverables, completing the contract, and transition beyond the contract. The SEMP focuses on interface activities with the contractor, including NASA technical team involvement with and monitoring of contracted work.

Often overlooked in project staffing estimates is the amount of time that technical team members are involved in contracting-related activities. Depending on the type of procurement, a technical team member involved in source selection could be consumed nearly full-time for 6 to 12 months. After contract award, technical monitoring consumes 30 to 50 percent, peaking at full-time when critical milestones or key deliverables arrive. Keep in mind for most contractor activities, NASA staff performs supplementary activities.

## Solicitations

The release of a solicitation to interested parties is the formal indication of a future contract. A solicitation conveys sufficient details of a Government need (along with terms, conditions, and instructions) to allow prospective contractors (or offerors) to respond with a proposal. Depending on the magnitude and complexity of the work, a draft solicitation may be issued. After proposals are received, a source evaluation board (or committee) evaluates technical and business proposals per its source evaluation plan and recommends a contractor selection to the contracting officer. The source evaluation board, led by a technical expert, includes other technical experts and a contracting specialist. The source selection process is completed when the contracting officer signs the contract.

Visit the on-line NASA Procurement Library [Procurement] for a full range of details regarding procurements and source selection. [SEB]

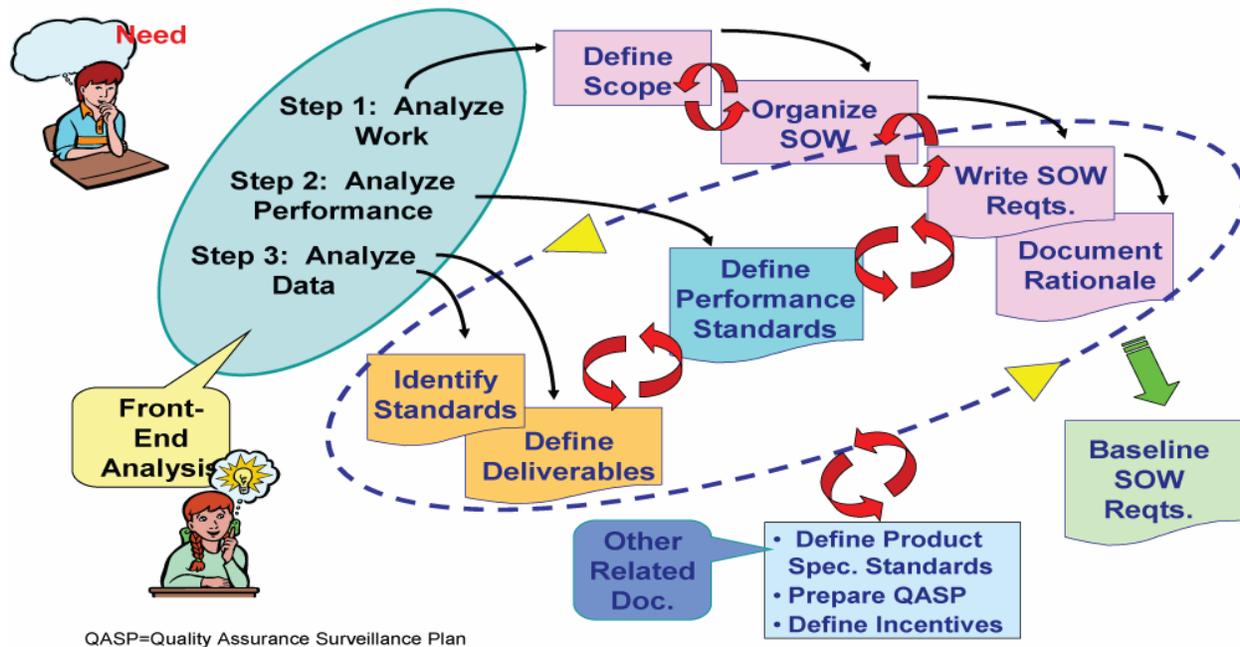
The technical team is intimately involved in developing technical documentation for the acquisition package. The acquisition package consists of the solicitation (e.g., request for proposals) and supporting documents. The solicitation contains all the documentation that is advertised to prospective contractors (or offerors). The key technical sections of the solicitation are the statement of work (SOW) (or performance work statement), technical specifications, and contract data requirements list. Other sections of the solicitation include proposal instructions and evaluation criteria. Documents that support the solicitation include a procurement schedule, source evaluation plan, Government cost estimate, and purchase request. Input from the technical team will be needed for some of the supporting documents.

It is the responsibility of the contract specialist, with input from the technical team, to ensure that the appropriate clauses are included in the solicitation. The contract specialist is familiar with requirements in the Federal Acquisition Regulations (FAR) and the NASA FAR Supplement (NFS) that will be included in the solicitation as clauses in full text form or as clauses incorporated by reference. Many of these clauses relate to public laws, contract administration, and financial management. Newer clauses address information technology security, data rights, intellectual property, new technology reporting, and similar items. The contract specialist stays abreast of updates to the FAR and NFS. As the statement of work and other parts of the solicitation mature, it is important for the contract specialist and technical team to work closely to avoid duplication of similar requirements.

### *7.1.3.2 Develop the Statement of Work*

Effective surveillance of a contractor begins with the development of the SOW. The technical team establishes the SOW requirements for the product to be developed. The SOW contains process, performance, and management requirements the contractor must fulfill during product development.

As depicted in Figure 7-2, developing the SOW requires the technical team to analyze the work, performance, and data needs to be accomplished by the contractor. The process is iterative and supports the development of other documentation needed for the contracting effort. The principal steps in the figure are discussed further in Table 7-2 below the figure.



**Figure 7-2 Contract Requirements Development Process**

**Table 7-2 Steps in the Requirements Development Process**

<b>Step 1: Analyze the Work</b>	
Define Scope	Document in the SOW that part of the project's scope that will be contracted. Give sufficient background information to orient offerors.
Organize SOW	Organize the work by products and associated activities (i.e., product WBS).
Write SOW Requirements	Include activities necessary to: <ul style="list-style-type: none"> <li>• develop products defined in the requirements specification and</li> <li>• support, manage, and oversee development of the products.</li> </ul> Write SOW requirements in the form "the Contractor shall." Write product requirements in the form "the system shall."
Document Rationale	Document separately from the SOW the reason(s) for including requirements that may be unique, unusual, controversial, political, etc. The rationale is not part of the solicitation.
<b>Step 2: Analyze Performance</b>	
Define Performance Standards	Define what constitutes acceptable performance by the contractor. Common metrics for use in performance standards include cost and schedule. For guidance on metrics to assess the contractor's performance and to assess adherence to product requirements on delivered products, refer to <i>System and Software Metrics for Performance-Based Contracting</i> [Metrics].
<b>Step 3: Analyze Data</b>	
Identify Standards	Identify standards (e.g., EIA, IEEE, ISO) that apply to deliverable work products including plans, reports, specifications, drawings, etc. Consensus standards and codes (e.g., NEC, NFPA, ASME) that apply to product development and workmanship are included in specifications.

Define Deliverables	Ensure each deliverable data item (e.g., technical data–requirements specifications, design documents; management data–plans, metrics reports) has a corresponding SOW requirement for its preparation. Ensure each product has a corresponding SOW requirement for its delivery.
---------------------	---

After a few iterations, baseline the SOW requirements and place them under configuration management. Refer to the Configuration Management Process.

Use the SOW checklist [SOWC] to help ensure that the SOW is complete, consistent, correct, unambiguous, and verifiable. Below are some key items to require in the SOW:

- Technical and management deliverables having the highest risk potential (e.g., SEMP, development and transition plans); requirements and architecture specifications; test plans, procedures and reports; metrics reports; delivery, installation, and maintenance documentation [Eslinger].
- Contractual or scheduling incentives in a contract should not be tied to the technical milestone reviews. These milestone reviews (for example, SRR, PDR, CDR, etc.) enable a critical and valuable technical assessment to be performed. These reviews have specific entrance criteria that should not be waived. The reviews should be conducted when these criteria are met, rather than being driven by a particular schedule.
- Timely electronic access to data, work products, and interim deliverables to assess contractor progress on final deliverables.
- Provision(s) to flow down requirements to sub-contractors and other team members.
- Content and format requirements of deliverables in the contract data requirements list. These requirements are specified in a data requirements document or data item description, usually as an attachment. Remember that you need to be able to edit data deliverables.
- Metrics to gain visibility into technical progress for each discipline (e.g., hardware, software, thermal, optics, electrical, mechanical). For guidance on metrics to assess the contractor’s performance and to assess adherence to product requirements on delivered products, refer to *System and Software Metrics for Performance-Based Contracting* [Metrics]. [reference Metrics here in Handbook]
- Quality incentives (defect, error count, etc.) to reduce risk of poor quality deliverables. Be careful because incentives can affect contractor behavior. For example, if you reward early detection and correction of software defects, the contractor may expend effort correcting minor defects and saving major defects for later.
- Risk management, including joint risk reviews and contractor-subcontractor approach.
- Surveillance activities (e.g., status meetings, reviews, audits, site visits) to monitor progress and production, especially access to subcontractors and other team members.

- Specialty engineering (e.g., reliability, quality assurance, cryogenics, pyrotechnics, biomedical, waste management) that is needed to fulfill standards and certifications requirements.
- Provisions to assign responsibilities between NASA and contractor according to verification, validation, or similar plans that are not available prior to award.
- Provisions to cause a contractor to disclose changing a critical process. If a process is critical to human safety (say, fabrication or certification), require the contractor to obtain approval from the contracting officer before a different process is implemented.

If you neglect to require something in the SOW, it can be costly to add it later.

The contractors must supply a SEMP that specifies their systems engineering approach for requirements development, technical solution definition, design realization, product evaluation, product transition, and technical planning, control, assessment, and decision analysis. It is best to request a preliminary SEMP in the solicitation. The source evaluation board can use the SEMP to evaluate the offeror's understanding of the requirements, as well as the offeror's capability and capacity to deliver the system. After contract award, the technical team can eliminate any gaps between the project's SEMP and the contractor's SEMP that could affect smooth execution of the integrated set of common technical processes.

Often a technical team has experience developing technical requirements, but little or no experience developing SOW requirements. If you give the contractor a complex set of technical requirements, but neglect to include sufficient performance measures and reporting requirements, you will have difficulty monitoring progress and determining product and process quality. Understanding performance measures and reporting requirements will enable you to ask for the appropriate data or reports that you intend to use.

Traditionally, NASA contracts require contractors to satisfy requirements in NASA policy directives, NASA Procedural Requirements, NASA standards, and similar documents. These documents are almost never written in language that can be used directly in a contract. Too often, these documents contain requirements that do not apply to contracts. So, before the technical team boldly goes where so many have gone before, it is a smart idea to understand what the requirements mean and if they apply to contracts. The requirements that apply to contracts need to be written in a way that is suitable for contracts.

### *7.1.3.3 Task Order Contracts*

Sometimes, the technical team can obtain engineering products and services through an existing task order contract. The technical team develops a task order SOW and interacts with the contracting officer's technical representative to issue a task order. Preparing the task order SOW is simplified because the contract already establishes baseline requirements for execution. First time users need to understand the scope of the contract and the degree to which delivery and reporting requirements, performance metrics, incentives, and so forth are already covered. Task contracts offer quick access (days or weeks instead of months) to engineering services for studies, analyses, design, development, and testing and to support services for configuration management, quality assurance, maintenance, and operations. Once a task order is issued, the

technical team performs engineering activities associated with managing contract performance and completing a contract (discussed later) as they apply to the task order.

#### *7.1.3.4 Surveillance Plan*

The surveillance plan defines the monitoring of the contractor effort and is developed at the same time as the SOW. The technical team works with mission assurance personnel, generally from the local Safety and Mission Assurance organization, to prepare the surveillance plan for the contracted effort. Sometimes mission assurance is performed by technical experts on the project. In either case, mission assurance personnel should be engaged from the start of the project. Prior to contract award, the surveillance plan is written at a general level to cover the Government's approach to perceived programmatic risk. After contract award, the surveillance plan describes in detail inspection, testing, and other quality-related surveillance activities that will be performed to ensure the integrity of contract deliverables, given the current perspective on programmatic risks. [FAR 46.103(a)] [NFS 1846.401(a)]

Recommended items in the surveillance plan include:

- Review key deliverables within the first 30 days to ensure adequate startup of activities.
- Conduct contractor/subcontractor site visits to monitor production or assess progress.
- Evaluate effectiveness of the contractor's systems engineering processes.

Drafting the surveillance plan when the SOW is developed promotes the inclusion of key requirements in the SOW that enable activities in the surveillance plan. For example, in order for the technical team to conduct site visits to monitor production of a subcontractor, then the SOW must include a requirement that permits site visits, combined with a requirement for the contractor to flow down requirements that directly affect subcontractors.

#### *7.1.3.5 Writing Proposal Instructions and Evaluation Criteria*

Once the technical team has written the SOW, the Government cost estimate, and the preliminary surveillance plan and updated the SEMP, the solicitation can be developed. Authors of the solicitation must understand the information that will be needed to evaluate the proposals and write instructions to obtain specifically needed information. In a typical source selection, the source selection board evaluates the offerors' understanding of the requirements, management approach, and cost and their relevant experience and past performance. This information is required in the business and technical proposals. (This section discusses only the technical proposal.) The solicitation also gives the evaluation criteria that the source evaluation board will use. This section corresponds one-for-one to the items requested in the proposal instructions section.

### Source Evaluation Board

One or more members of the technical team serve as members of the source evaluation board. They participate in the evaluation of proposals following applicable NASA and Center source selection procedures. Because source selection is so important, the procurement office works closely with the source evaluation board to ensure that the source selection process is properly executed. The source evaluation board develops a source evaluation plan that describes the evaluation factors, and the method of evaluating the offerors' responses. Unlike decisions made by systems engineers early in a product life cycle, source selection decisions must be carefully managed in accordance with regulations governing the fairness of the selection process.

State instructions clearly and correctly. The goal is to obtain enough information to have common grounds for evaluation. The challenge becomes how much information to give the offerors. If you are too prescriptive, the proposals may look too similar. Be careful not to level the playing field too much, otherwise discriminating between offerors will be difficult. Because the technical merits of a proposal compete with non-technical items of similar importance (e.g., cost), the technical team must wisely choose discriminators to facilitate the source selection.

The source evaluation board evaluates non-technical (business) and technical items. Items may be evaluated by themselves, or in the context of other technical or non-technical items. Table 7-3 shows technical items to request from offerors and the evaluation criteria with which they correlate.

**Table 7-3 Proposal Evaluation Criteria**

Item	Evaluation Criteria
Preliminary contractor SEMP.	How well the plan can be implemented given the resources, processes, and controls stated. Look at completeness (how well it covers all SOW requirements), internal consistency, and consistency with other proposals items. The SEMP should cover all resources and disciplines needed to meet product requirements, etc.
Process descriptions, including subcontractor's (or team member's) processes.	Effectiveness of processes and compatibility of contractor and subcontractor processes (e.g., responsibilities, decision-making, problem resolution, reporting).
Artifacts (documents) of relevant work completed. Such documentation depicts the probable quality of work products an offeror will provide on your contract. Artifacts provide evidence (or lack) of systems engineering process capability.	Completeness of artifacts, consistency among artifacts on a given project, consistency of artifacts across projects, conformance to standards.
Engineering methods and tools.	Effectiveness of the methods and tools.
Process and product metrics.	How well the offeror measures performance of its processes and quality of its products.
Subcontract management plan (may be part of contractor SEMP).	Effectiveness of subcontract monitoring and control and integration/separation of risk management and configuration management.
Phase-in plan (may be part of contractor SEMP).	How well the plan can be implemented given the existing workload of resources.

### *Evaluation Considerations*

The following are important to consider when evaluating proposals:

- Give adequate weight to evaluating the capability of disciplines that could cause mission failure (e.g., hardware, software, thermal, optics, electrical, mechanical).
- Conduct a pre-award site visit of production/test facilities that are critical to mission success.
- Distinguish between “pretenders” (good proposal writers) and “contenders” (good performing organizations). Pay special attention to how process descriptions match relevant experience and past performance. While good proposals can indicate good future performance, lesser quality proposals usually predict lesser quality future work products and deliverables.
- Assess the contractor’s SEMP and other items submitted with the proposal based on evaluation criteria that include quality characteristics (e.g., complete, unambiguous, consistent, verifiable, and traceable).

The cost estimate that the technical team performs as part of the Technical Planning Process supports evaluation of the offerors’ cost proposals, helping the source evaluation board determine the realism of the offerors’ technical proposals. [See Technical Planning.] The source evaluation board can determine “whether the estimated proposed cost elements are realistic for the work to be performed; reflect a clear understanding of the requirements; and are consistent with the unique methods of performance and materials described in the offeror’s technical proposal.” [FAR 15.404-1(d) (1)]

#### *7.1.3.6 Selection of COTS Products*

When COTS products are given as part of the technical solution in a proposal, it is imperative that the selection of a particular product be evaluated and documented by applying the Decision Analysis Process. Bypassing this task or neglecting to document the evaluation sufficiently could lead to a situation where NASA cannot support its position in the event of a vendor protest. [S06012]

#### *7.1.3.7 Acquisition-Unique Risks*

Table 7-4 identifies a few risks that are unique to acquisition along with ways to manage them from an engineering perspective. Bear in mind, legal and procurement aspects of these risks are generally covered in contract clauses.

**Table 7-4 Risks in Acquisition**

Risk	Mitigation
Supplier goes bankrupt prior to delivery	The source selection process is the strongest weapon. Select a supplier with a proven track record, solid financial position, and stable work force. As a last resort, the Government may take possession of any materials, equipment, and facilities on the work site necessary for completing the work in-house or via another contract.
Supplier acquired by another supplier with different policies	Determine differences between policies before and after the acquisition. If there is a critical difference, then consult with the procurement and legal offices. Meet with the supplier and determine if the original policy will be honored at no additional cost. If the supplier balks, then follow the advice from legal.
Deliverables include software to be developed	Include an experienced software manager on the technical team. Monitor the contractor's adherence to software development processes. Discuss software progress, issues, and quality at technical interchange meetings.
Deliverables include COTS products (especially software)	<p>Understand the quality of the product.</p> <ul style="list-style-type: none"> <li>• Look at test results. When test results show a lot of rework to correct defects, then users will probably find more defects.</li> <li>• Examine problem reports. These show whether or not users are finding defects after release.</li> <li>• Evaluate user documentation</li> <li>• Look at product support</li> </ul>
Products depend on results from models or simulations	Establish the credibility and uncertainty of results. Determine depth and breadth of practices used in verification and validation of the model or simulation. Understand the quality of software upon which the model or simulation is built. For more information, refer to NASA-STD-(I)-7009, <i>Standard for Models and Simulations</i> .
Budget changes prior to delivery of all products (and contract was written without interim deliverables)	<p>Options include:</p> <ul style="list-style-type: none"> <li>• Remove deliverables or services from the contract scope in order to obtain key products</li> <li>• Relax the schedule in exchange for reduced cost</li> <li>• Accept deliverables "as is"</li> </ul> <p>To avoid this situation, include electronic access to data, work products, and interim deliverables to assess contractor progress on final deliverables in the SOW.</p>
Contractor is a specialty supplier with no experience in a particular engineering discipline. For example, the contractor produces cryogenic systems that use alarm monitoring software from another supplier; however, the contractor does not have software expertise.	<p>Mitigate risks of COTS product deliverables as discussed earlier. If the contract is for delivery of a modified COTS product or custom product, then include provisions in the SOW to cover the following:</p> <ul style="list-style-type: none"> <li>• Supplier support (beyond product warranty) that includes sub-supplier support</li> <li>• Version upgrade/replacement plans</li> <li>• Surveillance of sub-supplier</li> </ul> <p>If the product is inexpensive, simply purchasing spares may be more cost effective than adding surveillance requirements.</p>

When the technical team completes the activities prior to contract award, they will have an updated SEMP, the Government cost estimate, a SOW, and a preliminary surveillance plan. Once the contract is awarded, the technical team begins technical oversight.

#### **7.1.4 During Contract Performance**

##### *7.1.4.1 Performing Technical Surveillance*

Surveillance of a contractor's activities and/or documentation is performed to demonstrate fiscal responsibility, ensure crew safety and mission success, and determine award fees for extraordinary (or penalty fees for substandard) contract execution. Prior to or outside of a contract award, a less formal agreement may be made for the Government to be provided with information for a trade study or engineering evaluation. Upon contract award, it may become necessary to monitor the contractor's adherence to contractual requirements more formally. (For a greater understanding of surveillance requirements, see *NPR 8735.2, Management of Government Quality Assurance Functions for NASA Contracts*.)

Under the authority of the contracting officer, the technical team performs technical surveillance as established in the NASA SEMP. The technical team assesses technical work productivity, evaluates product quality, and conducts technical reviews of the contractor. (Refer to the Technical Assessment process.) Some of the key activities are discussed below.

- **Develop NASA-contractor technical relationship.** At the contract kick-off meeting, set expectations for technical excellence throughout the execution of the contract. Highlight the requirements in the contract SOW that are the most important. Discuss the quality of work and products to be delivered against the technical requirements. Mutually agree on the format of the technical reviews and how to resolve misunderstandings, oversights, and errors.
- **Conduct technical interchange meetings.** Start early in the contract period and meet periodically with the contractor (and subcontractors) to confirm that the contractor has a correct and complete understanding of the requirements and operational concepts. Establish day-to-day NASA-contractor technical communications.
- **Control and manage requirements.** Almost inevitably, new or evolving requirements will affect a project. When changes become necessary, the technical team needs to control and manage changes and additions to requirements proposed by either NASA or the contractor. (See Requirements Management Process.) Communication changes to any project participants that the changes will affect. Any changes in requirements that affect contract cost, schedule, or performance must be conveyed to the contractor through a formal contract change. Consult the contracting officer's technical representative.
- **Evaluate systems engineering processes.** Evaluate the effectiveness of defined systems engineering processes. Conduct audits and reviews of the processes. Identify process deficiencies and offer assistance with process improvement.
- **Evaluate work products.** Evaluate interim plans, reports, specifications, drawings, processes, procedures, and similar artifacts that are created during the systems engineering effort.

- **Monitor contractor performance against key metrics.** Monitoring contractor performance extends beyond programmatic metrics to process and product metrics. (See technical performance measures, Section 6.7, Technical Assessment.) These metrics depend on acceptable product quality. For example, “50 percent of design drawings completed” is misleading if most of them have defects (e.g., incorrect, incomplete, inconsistent). The amount of work to correct the drawings affects cost and schedule. It is useful to examine reports that show the amount of contractor time invested in product inspection and review.
- **Conduct technical reviews.** Assess contractor progress and performance against requirements through technical reviews. (See Section 6.7, Technical Assessment).
- **Verify and validate products.** Verify and validate the functionality and performance of products before delivery and prior to integration with other system products. To ensure that a product is ready for system integration or to enable further system development, perform verification and validation as early as practical. (See Sections 5.3 and 5.4, Product Verification and Product Validation.)

#### 7.1.4.2 *Evaluating Work Products*

Work products and deliverables share common attributes that can be used to assess quality. Additionally, relationships among work products and deliverables can be used to assess quality. Some key attributes that help determine quality of work products are listed below:

- Satisfies content and format requirements
- Understandable
- Complete
- Consistent (internally and externally): terminology is consistent. An item is called the same thing throughout the documents.
- Traceable

Table 7-5 shows some typical work products from the contractor and key attributes with respect to other documents that can be used as evaluation criteria.

**Table 7-5 Typical Work Product Documents**

Work Product	Evaluation Criteria
SEMP	Describes activities and products required in the SOW. The SEMF is not complete unless it describes (or references) how each activity and product in the SOW will be accomplished.
Software management plan	Consistent with the SEMF and related project plans. Describes how each software-related activity and product in the SOW will be accomplished. Development approach is feasible.
System design	Covers the technical requirements and operational concepts. System can be implemented.
Software design	Covers the technical requirements and operational concepts. Consistent with hardware design. System can be implemented.
Installation plans	Covers all user site installation activities required in the SOW. Presents a sound approach. Shows consistency with the SEMF and related project plans.
Test plans	Covers qualification requirements in the SOW. Covers technical requirements. Approach is feasible.
Test procedures	Test cases are traceable to technical requirements
Transition plans	Describes all transition activities required in the SOW. Shows consistency with the SEMF and related project plans
User documentation	Sufficiently and accurately describes installation, operation, or maintenance (depending on the document) for the target audience.
Drawings and documents (general)	Complies with content and format requirements specified in the SOW.

*7.1.4.3 Issues with Contract-Subcontract Arrangements*

In the ideal world, a contractor manages its subcontractors, each subcontract contains all the right requirements, and resources are adequate. In the real world, the technical team deals with contractors and subcontractors who are motivated by profit, (sub)contracts with missing or faulty requirements, and resources that are consumed more quickly than expected. These and other factors cause or influence two key issues in subcontracting:

- limited or no oversight of subcontractors
- limited access to or unable to obtain subcontractor data

These issues are exacerbated when they apply to second (or lower tier) subcontractors. Table 7-6 looks at these issues more closely along with potential resolutions.

**Table 7-6 Contract-Subcontract Issues**

Issue	Resolution
Oversight of subcontractor is limited because requirement(s) missing from contract	The technical team gives the SOW requirement(s) to the contracting officer who adds the requirement(s) to the contract and negotiates the change order, including additional costs to NASA. The contractor then adds the requirement(s) to the subcontract and negotiates the change order with the subcontractor. If the technical team explicitly wants to perform oversight, then the SOW should indicate what the contractor, its subcontractors, and team members are required to do and provide.
Oversight of subcontractor is limited because requirement(s) not flowed down from contractor to subcontractor	<p>It is the contractor’s responsibility to satisfy the requirements of the contract. If the contract includes provisions to flow down requirements to subcontractors, then the technical team can request the contracting officer to direct the contractor to execute the provisions. The contractor may need to add requirements and negotiate cost changes with the subcontractor. If NASA has a cost-plus contract, then expect the contractor to bill NASA for any additional costs incurred. If NASA has a fixed price contract, then the contractor will absorb the additional costs or renegotiate cost changes with NASA.</p> <p>If the contract does not explicitly include requirements flow-down provisions, the contractor is responsible for performing oversight.</p>
Oversight of second tier subcontractor is limited because requirement(s) not flowed down from subcontractor to second tier subcontractor	<p>This is similar to the previous case, but more complicated. Assume that the contractor flowed down requirements to its subcontractor, but the subcontractor did not flow down requirements to the second tier subcontractor. If the subcontract includes provisions to flow down requirements to lower tier subcontractors, then the technical team can request the contracting officer to direct the contractor to ensure that subcontractors execute the flow-down provisions to their subcontractors.</p> <p>If the subcontract does not explicitly include requirements flow-down provisions, the subcontractor is responsible for performing oversight of lower tier subcontractors.</p>
Access to subcontractor data is limited or not provided because providing the data is not required in the <i>contract</i>	The technical team gives the SOW requirement(s) to the contracting officer who adds the requirement(s) to the contract and negotiates the change order, including additional costs to NASA. The contractor then adds the requirement(s) to the subcontract and negotiates the change order with the subcontractor. If the technical team explicitly wants direct access to subcontractor data, then the SOW should indicate what the contractor, its subcontractors, and team members are required to do and provide.
Access to subcontractor data is limited or not provided because providing the data is not required in the <i>subcontract</i>	It is the contractor’s responsibility to obtain data (and data rights) necessary to satisfy the conditions of its contract, including data from subcontractors. If the technical team needs direct access to subcontractor data, then follow the previous case to add flow-down provisions to the contract so that the contractor will add requirements to the subcontract.

Scenarios other than those above are possible. Resolutions might include reducing contract scope or deliverables in lieu of cost increases or sharing information technology in order to obtain data.

Even with the adequate flow-down requirements in (sub)contracts, legal wrangling may be necessary to entice contractors to satisfy the conditions of their (sub)contracts.

Activities during contract performance will generate an updated surveillance plan, minutes documenting meetings, change requests, and contract change orders. Processes will be assessed, deliverables and work products evaluated, and results reviewed.

### **7.1.5 Contract Completion**

The contract comes to completion with the delivery of the contracted products, services, or systems and their enabling products or systems. Along with the product, as-built documentation must be delivered and operational instructions including user manuals.

#### *7.1.5.1 Acceptance of Final Deliverables*

Throughout the contract period, the technical team reviews and accepts various work products and interim deliverables identified in the contract data requirements list and schedule of deliverables. The technical team also participates in milestone reviews to finalize acceptance of deliverables. At the end of the contract, the technical team ensures that each technical deliverable is received and that its respective acceptance criteria are satisfied. The technical team records the acceptance of deliverables against the contract data requirements list and the schedule of deliverables. These documents serve as an inventory of items and services to be accepted. Although rejections and omissions are infrequent, the technical team needs to take action in such a case. Good data management and configuration management practices facilitate the effort.

Acceptance criteria include:

- Product verification and validation completed successfully. The technical team performs or oversees verification and validation of products, integration of products into systems, and system verification and validation.
- Technical data package is current (as-built) and complete.
- Transfer of certifications, spare parts, warranties, etc. is complete.
- Transfer of software products, licenses, data rights, intellectual property rights, etc. is complete.
- Technical documentation required in contract clauses is complete (e.g., new technology reports).

It is important for NASA personnel and facilities to be ready to receive final deliverables. Key items to have prepared include:

- A plan for support and to transition products to operations
- Training of personnel
- Configuration management system in place
- Responsibilities for troubleshooting, repair, and maintenance are allocated

### *7.1.5.2 Transition Management*

Before the contract was awarded, a product support strategy was developed as part of the acquisition strategy. The product support strategy outlines preliminary notions regarding integration, operations, maintenance, improvements, decommissioning, and disposal. Later, after the contract is awarded, a high-level transition plan that expands the product support strategy is recorded in the SEMP. Details of product/system transition are subsequently documented in one or more transition plans. Elements of transition planning are discussed in the Product Transition Process. (See section 5.5).

The transition plan(s) must clearly indicate responsibility for each action (NASA or contractor). Also, the contract SOW must have included a requirement that the contractor will execute responsibilities assigned in the transition plan (usually on a cost-reimbursable basis).

Frequently, NASA (or NASA jointly with a prime contractor) is the system integrator on a project. In this situation, multiple contractors (or subcontractors) will execute their respective transition plan(s). NASA is responsible for developing and managing a system integration plan that incorporates inputs from each transition plan. The provisions that were written in the SOW months or years earlier accommodate the transfer of products and systems from the contractor(s) to NASA.

### *7.1.5.3 Transition to Operations and Support*

The successful transition of systems to operations and support, which includes maintenance and improvements, depends on clear transition criteria that the stakeholders agree on. The technical team participates in the transition, providing continuity for the customer, especially when a follow-on contract is involved. When the existing contract is used, the technical team conducts a formal transition meeting with the contractor. Alternately, the transition may involve the same contractor under a different contract arrangement (e.g., modified or new contract). Or the transition may involve a different contractor than the developer, using a different contract arrangement.

The key benefits of using the existing contract are that the relevant stakeholders are familiar with the contractor and that the contractor knows the products and systems involved. Ensure that the contractor and other key stakeholders understand the service provisions (requirements) of the contract. This meeting may lead to contract modifications in order to amend or remove service requirements that have been affected by contract changes over the years.

Seeking to retain the development contractor under a different contract can be beneficial. Although it takes time and resources to compete the contract, it permits NASA to evaluate the contractor and other offerors against operations and support requirements only. The incumbent contractor has personnel with development knowledge of the products and systems, while service providers specialize in optimizing cost and availability of services. In the end, the incumbent may be retained under a contract that focuses on current needs (not several years ago), or else a motivated service provider will work hard to understand how to operate and maintain the systems. If a follow-on contract will be used, consult the local procurement office and exercise the steps that were used to obtain the development contract. Assume that the amount of calendar time to award a follow-on contract will be comparable to the time to award the

development contract. Also consider that the incumbent may be less motivated upon losing the competition.

Some items to consider for follow-on contracts during the development of SOW requirements:

- Staff qualifications
- Operation schedules, shifts, staffing levels
- Maintenance profile (e.g., preventive, predictive, run-to-fail)
- Maintenance and improvement opportunities (e.g., schedule, turnaround time)
- Historical data for similar efforts
- Performance-based work

The transition to operations and support represents a shift from the delivery of products to the delivery of services.

Service contracts focus on the contractor's performance of activities, rather than development of tangible products. Consequently, performance standards reflect customer satisfaction and service efficiency, such as:

- Customer satisfaction ratings
- Efficiency of service
- Response time to a customer request
- Availability (e.g., of system, website, facility)
- Time to perform maintenance action
- Planned versus actual staffing levels
- Planned versus actual cost
- Effort and cost per individual service action
- Percent decrease in effort and cost per individual service action

For more examples of standards to assess the contractor's performance, refer to *System and Software Metrics for Performance-Based Contracting* [Metrics].

#### 7.1.5.4 *Decommissioning and Disposal*

Contracts offer a means to achieve the safe and efficient decommissioning and disposal of systems and products that require specialized support systems, facilities, and trained personnel, especially when hazardous materials are involved. Consider these needs during development of the acquisition strategy and solidify them before the final design phase. Determine how many contracts will be needed across the product's life cycle.

Some items to consider for decommissioning and disposal during the development of SOW requirements:

- Handling and disposal of waste generated during the fabrication and assembly of the product
- Reuse and recycling of materials to minimize the disposal and transformation of materials
- Handling and disposal of materials used in the product's operations

- End-of-life decommissioning and disposal of the product
- Cost and schedule to decommission and dispose of the product, waste, and unwanted materials
- Metrics to measure decommissioning and disposal of the product.
- Metrics to assess the contractor's performance. (Refer to *System and Software Metrics for Performance-Based Contracting* [Metrics].)

For guidelines regarding disposal, refer to the *Systems Engineering Handbook: A "What To" Guide for all SE Practitioners* [INCOSE].

#### *7.1.5.5 Final Evaluation of Contractor Performance*

In preparation for closing out a contract, the technical team gives input to the procurement office regarding the contractor's final performance evaluation. Although the technical team has performed periodic contractor performance evaluations, the final evaluation offers a means to document good and bad performance that continued throughout the contract. Since the evaluation is retained in a database, it can be used as relevant experience and past performance input during a future source selection process.

This phase of oversight is complete with the closeout or modification of the existing contract, award of the follow-on contract, and an operational system. Oversight continues with follow-contract activities.

## 7.2 Integrated Design Facilities

### 7.2.1 Introduction

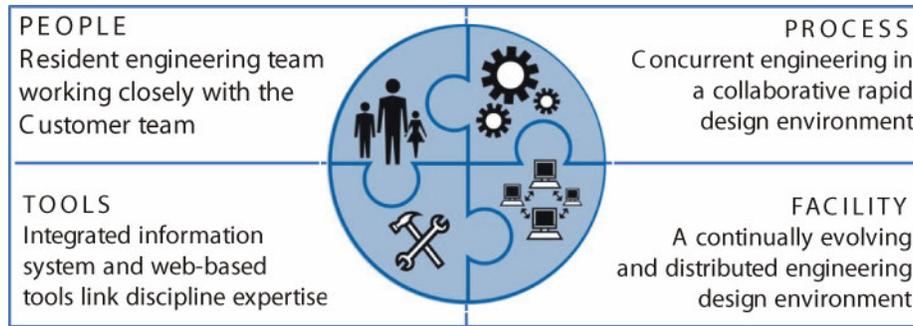
Concurrent Engineering (CE) and integrated design is a systematic approach to integrated product development that emphasizes response to customer expectations and embodies team values of cooperation, trust, and sharing. The objective of CE is to reduce the product development cycle time through a better integration of activities and processes. Parallelism is the prime concept in reducing design lead time and concurrent engineering becomes the central focus. Large intervals of parallel work on different parts of the design are synchronized by comparatively brief exchanges between teams to produce consensus and decisions.<sup>4</sup> CE has become a widely accepted concept and is regarded as an excellent alternative approach to a sequential engineering process.

This section addresses the specific application of CE and integrated design practiced at NASA in Capability for Accelerated Concurrent Engineering (CACE) environments. CACE is comprised of four essential components: people, process, tools, and facility. The CACE environment typically involves the collocation of an in-place leadership team and core multidisciplinary engineering team working with a customer team using well defined processes in a dedicated collaborative, concurrent engineering facility with specialized tools. The engineering and collaboration tools are connected by the facility's integrated infrastructure. The teams work synchronously for a short period of time in a technologically intensive physical environment to complete an instrument or mission design. CACE is most often used to design space instruments and payloads or missions including orbital configuration; hardware such as spacecraft, landers, rovers, probes, or launchers; data and ground communication systems; other ground systems; and mission operations. But the CACE process applies beyond strict instrument and/or mission conceptual design.

Most NASA centers have a CACE facility. NASA CACE is built upon a people/process/tools/facility (PPTF) paradigm that enables the accelerated production of a high quality engineering design concepts in a concurrent, collaborative, rapid design environment. (See Figure 7-3.)

---

<sup>4</sup> [from Miao, Y. & J.M. Haake (1998). Supporting Concurrent Design by Integrating Information Sharing and Activity Synchronization, in Proceedings of the 5<sup>th</sup> ISPE International Conference on Concurrent Engineering Research and Applications (CE98), p. 165-174.]



**Figure 7-3 CACE People/Process/Tools/Facility Paradigm**

Although CACE at NASA is based on a common philosophy and characteristics, specific CACE implementation varies in many areas. These variations include level of engineering detail, information infrastructure, knowledge base, areas of expertise, engineering staffing approach, administrative and engineering tools, type of facilitation, roles and responsibilities within CACE team, roles and responsibilities across CACE and customer teams, activity execution approach, and duration of session. While primarily used to support early life-cycle phases such as pre-Formulation and Formulation, the CACE process has demonstrated applicability across the full project life cycle.

### **7.2.2 CACE Overview and Importance**

CACE design techniques can be an especially effective and efficient method of generating a rapid articulation of concepts, architectures, and requirements.

The CACE approach provides an infrastructure for brainstorming and bouncing ideas between the engineers and customer team representatives, which routinely results in a high quality product that directly maps to the customer needs. The collaboration design paradigm is so successful because it enables a radical reduction in decision latency. In a non-CACE environment, questions, issues, or problems may take several days to resolve. If a design needs to be changed or a requirement re-evaluated, significant time may pass before all engineering team members get the information or customer team members can discuss potential requirement changes. These delays introduce the possibility, following initial evaluation, of another round of questions, issues, and changes to design and requirements, adding further delays.

The tools, data, and supporting information technology infrastructure within CACE provide an integrated support environment that can be immediately utilized by the team. The necessary skills and experience are gathered and are resident in the environment to synchronously complete the design. In a collaborative environment, questions can be answered immediately or key participants can explore assumptions and alternatives with the customer team or other design team members and quickly reorient the whole team when a design change occurs. The collaboration triggers the creativity of the engineers and helps them close the loop and rapidly converge on their ideas. Since the mid-1990s, the CACE approach has been successfully used at several NASA Centers as well as at commercial enterprises to dramatically reduce design development time and costs when compared to traditional methods.

CACE customers include NASA programs and projects, scientists, and technologists as well as other Government agencies, civil and military, Federal laboratories, and universities. CACE products and services include:

- Full end-to-end designs including system/subsystem concepts, requirements, and tradeoffs,
- Focused efforts assessing specific architecture subelements and tradeoffs,
- Independent assessments of customer-provided reports, concepts, and costs,
- Roadmapping support, and
- Technology and risk assessments.

As integrated design has become more accepted, collaborative engineering design efforts expanded from the participation of one or more Centers in a locally executed activity; to geographically distributed efforts across a few NASA Centers with limited scope and participation; to true OneNASA efforts with participation from many NASA integrated design teams addressing broad, complex architectures.

The use of geographically distributed CACE teams is a powerful engineering methodology to achieve lower risk and more creative solutions by factoring in the best skills and capabilities across the Agency. Using a geographically distributed process must build upon common CACE elements while considering local CACE facility differences and the differences in the local Center cultures.

### ***7.2.3 CACE Purpose and Benefits***

The driving forces behind the creation of NASA's early CACE environments were increased systems engineering efficiency and effectiveness. More specifically, the early CACE environments addressed the need for:

- Generating more conceptual design studies at reduced cost and schedule
- Creating a reusable process within dedicated facilities using well-defined tools
- Developing a database of mission requirements and designs for future use
- Developing mission generalists from a pool of experienced discipline engineers
- Infusing a broader systems engineering perspective across the organization

Additional resulting strategic benefits across NASA included:

- Core competency support (e.g., developing systems engineers, maturing and broadening of discipline engineers, training environment, etc.)
- Sensitizing the customer base to end-to-end issues and implications of requirements upon design
- Testbed environment for improved tools and processes

- Environment for forming partnerships
- Technology development and roadmapping support
- Improved quality and consistency of conceptual designs
- One NASA environment that enables cooperative rather than competitive efforts between NASA organizations

#### **7.2.4 CACE Staffing**

A management or leadership team, a multidisciplinary engineering team, a customer team and a facility support team are all vital elements in achieving a successful CACE activity.

A CACE team consists of a cadre of engineers, each representing a different discipline or specialty engineering area along with a lead systems engineer and a team lead or facilitator. As required, the core engineering team is supplemented with specialty and/or nonstandard engineering skills to meet unique customer needs. These supplementary engineering capabilities can be obtained either from the local Center or from an external source. The team lead coordinates and facilitates the CACE activity and interacts with the customers to ensure that their objectives are adequately captured and represented. Engineers are equipped with techniques and software used in their area of expertise and interact with the team lead, other engineers, and the customer team to study the feasibility of a proposed solution and produce a design for their specific subsystem.

A CACE operations manager serves as the Center advocate and manager, maintaining an operational capability, providing initial coordination with potential customers through final delivery of CACE product, and infusing continuous process and product improvement as well as evolutionary growth into the CACE environment to ensure its continued relevance to the customer base.

A CACE facility support team maintains and develops the information infrastructure to support CACE activities.

#### **7.2.5 CACE Process**

The CACE process starts with a customer requesting engineering support from CACE management. CACE management establishes that the customer's request is within the scope of the team capabilities and availability and puts together a multidisciplinary engineering team under the leadership of a team lead and lead systems engineer collaborating closely with the customer team. The following sections briefly describe the three major CACE activity phases: (1) planning and preparation, (2) execution, and (3) wrap-up.

##### **7.2.5.1 Planning and Preparation**

Once a customer request is approved and team lead chosen, a planning meeting is scheduled. The key experts attending the planning meeting may include the CACE manager, a team lead, and a systems engineer as well as key representatives from the customer team. Interactions with the customer team and their active participation in the process are integral to the successful planning,

preparation, and execution of a concurrent design session. Aspects addressed include establishing the activity scope, schedule, and costs; a general agreement on the type of product to be provided; and the exit criteria and metrics. Agreements reached at the planning meeting are documented and distributed for review and comment.

Products from the planning and preparation phase include the identification of activities required by the customer team, the CACE team, or a combination of both teams, as well as the definition of the objectives, the requirements, the deliverables, the estimated budget, and the proposed schedule. Under some conditions, followup coordination meetings are scheduled that include the CACE team lead, the systems engineer(s), and a subset of the remaining team members, and customer representatives, as appropriate. The makeup of participants is usually based on the elements that have been identified as the activity drivers and any work identified that needs to be done before the actual design activity begins.

During the planning and preparation process, the customer-provided data and the objectives and activity plan are reviewed, and the scope of the activity is finalized. A discussion is held of what activities need to be done by each of the customer and the design teams. For example, for planning a mission design study, the customer identifies the mission objectives by defining the measurement objectives and the instrument specifications, as applicable, and identifying the top level requirements. A subset of the CACE engineering team may perform some preliminary work before the actual study (e.g., launch vehicle performance trajectory analysis, thrust and navigation requirements, the entry, descent and landing profile, optical analysis, mechanical design, etc.) as identified in the planning meetings to further accelerate the concurrent engineering process in the study execution phase. The level of analysis in this phase is a function of many things, including the level of maturity of the incoming design, the stated goals and objectives of the engineering activity, engineer availability, and CACE scheduling.

#### *7.2.5.2 Activity Execution Phase*

A typical activity or study begins with the customer presentation of the overall mission concept and instrument concepts, as applicable, to the entire team. Additional information provided by the customers includes the team objectives, the science and technology goals, the initial requirements for payload, spacecraft and mission design, the task breakdown between providers of parts or functions, top challenges and concerns, and the approximate mission timeline. This information is often provided electronically in a format accessible to the engineering team and is presented by the customer representatives at a high level. During this presentation, each of the subsystems engineers focuses on the part of the overall design that is relevant to their subsystem. The systems engineer puts the high level system requirements into the systems spreadsheets and/or a database that is used through out the process to track engineering changes. These data sources can be projected on the displays to keep the team members synchronized and the customers aware of the latest developments.

The engineering analysis is performed iteratively with the CACE team lead and systems engineer playing key roles to lead the process. Thus issues are quickly identified, so consensus on tradeoff decisions and requirements redefinition can be achieved while maintaining momentum. The customer team actively participates in the collaborative process (e.g., trade studies, requirement relaxation, clarifying priorities) contributing to the rapid development of an acceptable product.

Often, there are breakout sessions, or sidebars, in which part of the team discusses a particular tradeoff study. Each subsystem has a set of key parameters that are used for describing its design. Because of the dependencies between the various subsystems, each discipline engineer needs to know the value of certain parameters related to other subsystems. These parameters are shared via the CACE information infrastructure. Often, there are conflicting or competing objectives for various subsystems. Many tradeoff studies, typically defined and led by the team systems engineer, are conducted between subsystem experts immediately as issues occur. Most of the communication between team members is face-to-face or live via video- or teleconference. Additional subject matter experts are consulted as required. In the CACE environment, subsystems that need to interact extensively are clustered in close proximity to facilitate the communication process between the experts.

The team iterates on the requirements and each subsystem expert refines or modifies design choices as schedule allows. This process continues until an acceptable solution is obtained. There may be occasions where it is not possible to iterate to an acceptable solution prior to the scheduled end of the activity. In those cases, the available iterated results are documented and form the basis of the delivered product.

In each iteration, activities such as the following take place, sometimes sequentially and other times in parallel. The subsystem experts of science, instruments, mission design, and ground systems collaboratively define the science data strategy for the mission in question. The telecom, ground systems, and command and data handling (C&DH) experts develop the data-return strategy. The attitude control systems (ACS), power, propulsion, thermal, and structure experts iterate on the spacecraft design and the configuration expert prepares the initial concept. The systems engineer interacts with all discipline engineers to ensure that the various subsystem designs fit into the intended system architecture. Each subsystem expert provides design and cost information and the cost expert estimates the total cost for the mission.

While design activity typically takes only days or weeks with final products available within weeks after study completion, longer-term efforts take advantage of the concurrent, collaborative environment to perform more detailed analyses than those performed in the shorter duration CACE exercises.

### *7.2.5.3 Activity Wrap-Up*

After the completion of a CACE study, the product is delivered to the customer. In some CACE environments, the wrap-up of the product is completed with minimal additional resources: the engineers respond to customer feedback by incorporating additional refinements or information emphasizing basic clean-up. In other CACE environments, significant time is expended to format the final report and review it with the customer to ensure that their expectations have been addressed adequately.

Some CACE environments have standardized their wrap-up activities to address the customer feedback and develop products that are structured and uniform across different ranges of efforts.

As part of activity followup, customer feedback is requested on processes, whether the product met their needs, and whether there are any suggested improvements. This feedback is factored back into the CACE environment as part of a continuous improvement process.

## **7.2.6 CACE Engineering Tools and Techniques**

Engineering tools and techniques vary within and across CACE environments in several technical aspects (e.g., level of fidelity, level of integration, generally available commercial applications versus custom tools versus customized knowledge-based Excel spreadsheets, degree of parametric design and/or engineering analysis). For example, mechanical design tools range from white-board discussions to note pad translations to computer-aided design (CAD) to 3D rapid design prototyping.

Important factors in determining which tools are appropriate to an activity include the purpose and duration of the activity, the engineers' familiarity or preference, the expected product, the local culture, and the evolution of the engineering environment. Factors to be considered in the selection of CACE tools and engineering techniques should also include flexibility, compatibility with CACE environment and process, and value and ease of use for the customer after the CACE activities.

Engineering tools may be integrated into the CACE infrastructure, routinely provided by the supporting engineering staff, and/or utilized only on an activity-by-activity basis, as appropriate. As required, auxiliary engineering analysis outside of the scope of the CACE effort can be performed external to the CACE environment and imported for reference and incorporation into the CACE product.

## **7.2.7 CACE Facility, Information Infrastructure, and Staffing**

Each CACE instantiation is unique to the Center, program, or project that it services. While the actual implementations vary, the basic character does not. Each implementation concentrates on enabling engineers, designers, team leads, and customers to be more productive during concurrent activities and communication. This section focuses on three aspects of this environment: the facility, the supporting information infrastructure, and the staff required to keep the facility operational.

### **7.2.7.1 Facility**

The nature of communication among discipline specialists working together simultaneously creates a somewhat chaotic environment. Although it is the duty of the team lead to maintain order in the environment, the facility itself has to be designed to allow the participants to maintain order and remain on task while seeking to increase communication and collaboration. To do this effectively requires a significant investment in infrastructure resources.

The room needs sufficient space to hold active participants from the disciplines required, customer representatives, and observers. CACE managers encourage observers to show potential future CACE users the value of active CACE sessions.

It is also important to note that the room will get reconfigured often. Processes and requirements change, and the CACE facility must change with that. The facility could appear to an onlooker as a work in progress. Tables, chairs, computer workstations, network connections, electrical supplies, and visualization systems will continually be assessed for upgrades, modification, or elimination.

CACE requirements in the area of visualization are unique. When one subject matter expert wants to communicate to either a group of other discipline specialists or to the whole group in general, the projection system needs to be able to switch to different engineering workstations. When more than one subject matter expert wants to communicate with different groups, multiple projection systems need to be able to switch. This can typically require 3-6 projection systems with switching capability from any specific workstation to any specific projector. In addition, multiple projection systems switchable to the engineering workstations need to be mounted so that they can be viewed without impacting other activities in the room or so that the entire group can be refocused as required during the session. The ease of this reconfiguration is one measure of the efficacy of the environment.

#### *7.2.7.2 Information Infrastructure*

A CACE system not only requires a significant investment in the facility but relies heavily on the information infrastructure. Information infrastructure requirements can be broken down into three sections: hardware, software, and network infrastructure.

The hardware portion of the information infrastructure used in the CACE facility is the most transient element in the system. The computational resources, the communication fabric, servers, storage media, and the visualization capabilities benefit from rapid advances in technology. A CACE facility must be able to take advantage of the economy produced by those advances and must also be flexible enough to take advantage of the new capabilities.

One of the major costs of a CACE infrastructure is software. Much of the software currently used by engineering processes is modeling and simulation, usually produced by commercial software vendors. Infrastructure software to support exchange of engineering data, to manage study archive, and to track, administer, and manage facility activities is integral to CACE success. One of the functions of the CACE manager is to determine how software costs can be paid along with what software should be the responsibility of the participants and customers.

The network infrastructure of a CACE facility is critical. Information flowing between workstations, file servers, and visualization systems in real time requires a significant network infrastructure. In addition, the network infrastructure enables collaboration with outside consultants, external discipline experts, and intra-Center collaboration. The effective use of the network infrastructure requires a balance between network security and collaboration and as such will always be a source of modification, upgrade, and reconfiguration. A natural extension of this collaboration is the execution of geographically distributed CACE efforts; therefore it is essential that a CACE facility have the tools, processes, and communications capabilities to support such distributed studies.

#### *7.2.7.3 Facility Support Staff Responsibilities*

A core staff of individuals is required to maintain an operational CACE environment. The responsibilities to be covered include end-to-end CACE operations and the management and administration of the information infrastructure.

CACE information infrastructure management and administration includes computer workstation configuration, network system administration, documentation development, user help service, and software support to maintain infrastructure databases, tools, and websites.

## 7.2.8 CACE Products

CACE products are applicable across project life-cycle phases and can be clearly mapped to the various outputs associated with the systems engineering activities such as requirements definition, trade studies, decision analysis, and risk management. CACE products from a typical design effort include a requirements summary with driving requirements identified; system and subsystem analysis; functional architectures and data flows; mass/power/data rack ups; mission design and ConOps; engineering trades and associated results; technology maturity levels; issues, concerns, and risks; parametric and/or grass-roots cost estimates; engineering analyses, models, and applicable tools to support potential future efforts; and a list of suggested future analyses.

CACE product format and content vary broadly both within and across CACE environments. The particular CACE environment, the goals/objectives of the supported activity, whether the activity was supported by multiple CACE teams or not, the customer's ultimate use, and the schedule requirements are some aspects that factor into the final product content and format. A primary goal in the identification and development of CACE products and in the packaging of the final delivery is to facilitate their use after the CACE activity.

Products include in-study results presentation, PowerPoint packages, formal reports and supporting CAD models and engineering analysis. Regardless of format, the CACE final products typically summarize the incoming requirements, study goal expectation, and study final results.

CACE environment flexibility enables support activities beyond that of a traditional engineering design study (e.g., independent technical reviews, cost validation, risk and technology assessments, roadmapping, requirements review). Product contents for such activities might include feasibility assessment, technical recommendations, risk identification, recosting, technology infusion impact and implementation approach, and architectural options.

In addition to formal delivery of the CACE product to the customer team, the final results and planning data are archived within the CACE environment for future reference and for inclusion in internal CACE cross-study analyses.

## 7.2.9 CACE Best Practices

This section contains general CACE best practices for a successful CACE design activity. Three main topic areas—people, process, technologies—are applicable to both local and geographically distributed activities. Many lessons learned about the multi-CACE collaboration activities were learned through the NASA Exploration Design Team (NEDT) effort, a OneNASA multi-Center distributed collaborative design activity performed during FY05.

### 7.2.9.1 People

- a. **Training:** Individuals working in CACE environments benefit from specialized training. This training should equip individuals with the basic skills necessary for efficient and effective collaboration. Training should include what is required technically as well as orientation to the CACE environment and processes.

- b. **Characteristics:** Collaborative environment skills include being flexible, working with many unknowns, and willingness to take risks. Ability and willingness to think and respond in the moment is required as well as the ability to work as part of a team and to interact directly with customer representatives to negotiate requirements and to justify design decisions. Supporting engineers also need the ability to quickly and accurately document their final design as well as present this design in a professional manner. In addition, the CACE team leads or facilitators should have additional qualities to function well in a collaborative design environment. These include organizational and people skills, systems engineering skills and background, and broad general engineering knowledge.

#### 7.2.9.2 *Process and Tools*

- a. **Customer Involvement:** Managing customer expectations is the number one factor in positive study outcome. It is important to make the customers continuously aware of the applications and limitations of the CACE environment and to solicit their active participation in the collaborative environment.
- b. **Adaptability:** The CACE environments must adapt processes depending on study type and objectives, as determined in negotiations prior to study execution. In addition to adapting the processes, engineers with appropriate engineering and collaborative environment skills must be assigned to each study.
- c. **Staffing:** Using an established team has the benefit of the team working together and knowing each other and the tools and processes. A disadvantage is that a standing army can get “stale” and not be fluent with the latest trends and tools in their areas of expertise. Supporting a standing army full time is also an expensive proposition and often, not possible. A workable compromise is to have a full-time (or nearly full-time) leadership team complemented by an engineering team. This engineering team could be composed of engineers on rotational assignments or long-term detail to the team, as appropriate. An alternate paradigm is to partially staff the engineering team with personnel provided through the customer team.
- d. **Tools and Data Exchange:** In general, each engineer should use the engineering tools with which he or she is most familiar to result in an effective and efficient process. The CACE environment should provide an information infrastructure to integrate resulting engineering parameters.
- e. **Decision Process:** Capturing the decisionmaking and design rationale is of great interest and of value to CACE customers as well as being a major challenge in the rapid engineering environment. The benefit of this is especially important as a project progresses and makes the CACE product more valuable to the customer. Further along in the life cycle of a mission or instrument, captured decisions and design rationale are more useful than a point-design from some earlier time.
- f. **Communication:** CACE environments foster rapid communication between the team members. Because of the fast paced environment and concurrent engineering activities, keeping the design elements “in synch” is a challenge. This challenge can be addressed by proactive systems engineers, frequent tag-ups, additional systems engineering support and the use of appropriate information infrastructure tools.

- g. **Standards across CACE environments:** Establishing minimum requirements and standard sets of tools and techniques across the NASA CACE environment would facilitate multi-Center collaborations.
- h. **Planning:** Proper planning and preparation are crucial for efficient CACE study execution. Customers wanting to forgo the necessary prestudy activity or planning and preparation must be aware of and accept the risk of a poor or less-than-desired outcome.

#### 7.2.9.3 Facility

- a. **Communication Technologies:** The communication infrastructure is the backbone of the collaborative CACE environment. Certain technologies should be available to allow efficient access to resources external to a CACE facility. It is important to have “plug and play” laptop capability, for example. Multiple phones should be available to the team and cell phone accessibility is desirable.
- b. **Distributed Team Connectivity:** Real-time transfer of information for immediate access between geographically distributed teams or for multi-Center activities can be complicated due to firewall and other networking issues. Connectivity and information transfer methods should be reviewed and tested before study execution.

### 7.3 Human Factors Engineering

The discipline of human factors (HF) is devoted to the study, analysis, design and evaluation of human-machine interfaces and human organizations, with an emphasis on human capabilities and limitations as they impact system operation. In applied HF, practitioners are primarily concerned with drawing on the HF knowledge base and applying it to design.

People represent critical components in many systems, as operators, installers, maintainers, and other system users. In the world of human-system interaction, there are three avenues for improving performance and reducing error: 1) personnel selection, 2) interface design, and 3) training. Most effective performance improvement involves all three avenues. People can be highly selected for the work they are to perform and the environment they are to perform it in. Second, equipment and systems can be designed to be easy to use, error resistant, and quickly learned. Third, people can be trained to proficiency on their required tasks. Astronauts are highly selected and highly trained, but the first generation Space Shuttle would not generally be considered elegant design from a crew usage point of view. Becoming an automobile driver is an almost entirely non-selective process. It does, however, require significant initial training, and automobiles are continually being redesigned to the benefit of the driver.

Human factors focuses on those aspects where people interface with the system. It considers all personnel who must interact with the system, not just the operator; deals with organizational systems as well as hardware; and examines all types of interaction, not just hardware or software interfaces. The role of the HF specialist is to advocate for the human component and to ensure that the design of hardware, software, tasks, and environment is compatible with the sensory, perceptual, cognitive, and physical attributes of those interacting with the system. The HF specialist should elucidate why human-related issues or features should be included in analyses, design decisions, or tests and explain how design options will affect human performance in ways that impact total system performance and/or cost. As system complexity grows, the potential for conflicts between requirements increases. Sophisticated human-system interfaces create conflicts such as the need to create systems which are easy for novices to learn while also being efficient for experts to use. The HF specialist recognizes these tradeoffs and constraints and provides guidance on balancing these competing requirements. The domain of application is anywhere there are concerns regarding human and organizational performance, error, safety and comfort. The goal is always to inform and improve the design.

What distinguishes a human factors professional is the particular knowledge and methods used, the domain of employment, and the goal of the work. HF professionals specialize in the knowledge of behavior, both general and specific. There are many academic specialties concerned with applying knowledge of human behavior. These include psychology, cognitive science, cognitive psychology, sociology, economics, instructional system development, education, physiology, industrial psychology, and industrial engineering. Project and/or process managers should consult with their local Chief Engineer's Office to get advice and recommendations on specific human factors personnel that would be appropriate for their particular activity.

It is important to have HF personnel on the team throughout all the systems engineering common technical processes so that they can construct the specific HF analysis techniques and tests customized to the specific process or project. Not only do the HF specialists help in the

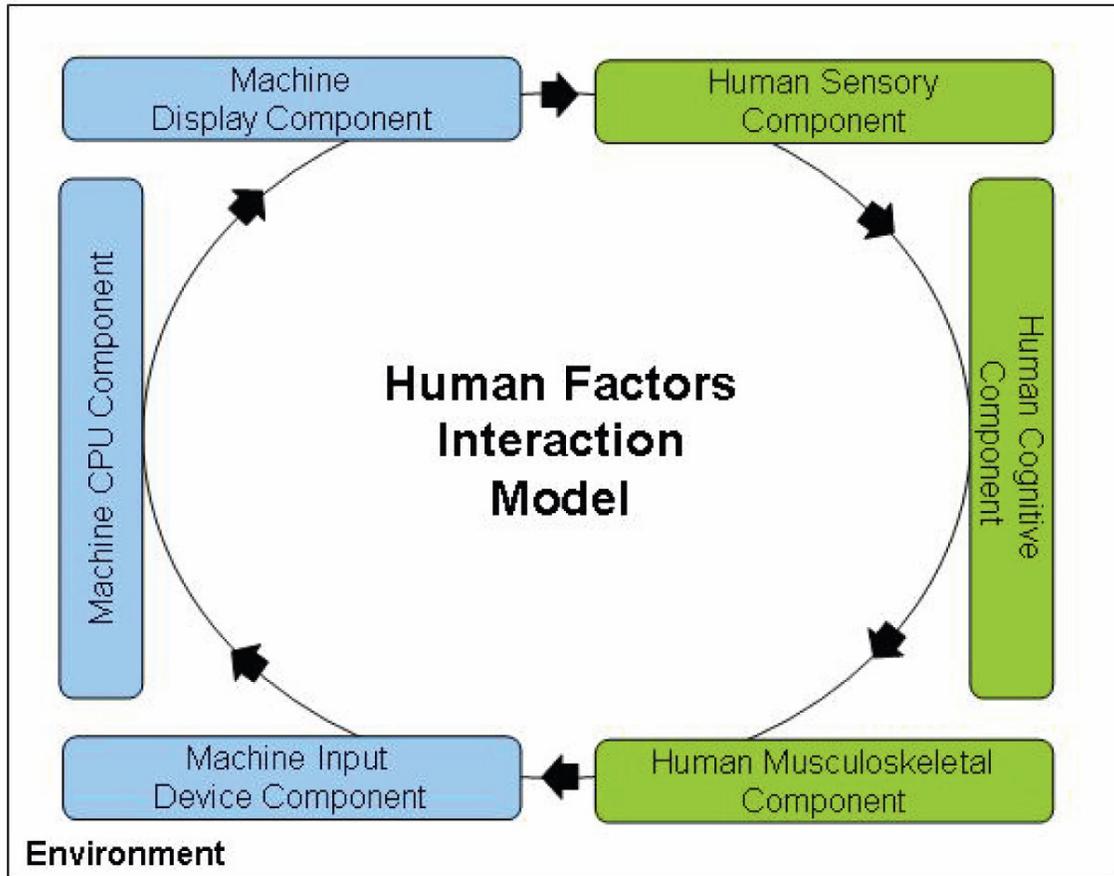
development of the end items in question, but they should also be used to make sure that the verification test and completeness techniques are compatible and accurate for humans to undertake. For example, to design a test for V&V, HF engineers can ensure the test gets done in a way where humans can understand the output and make the judgment call they need to. The technical team could build a test to assert how something works, but the process for that test may make bad assumptions about how humans perceive and verify things. Participation early in the process is especially important. Entering the system design process early ensures that human systems requirements are “designed in” rather than corrected later. Sometimes the results of analyses performed later call for a re-examination of earlier analyses. For example, functional allocation typically must be refined as design progresses because of technological breakthroughs, unforeseen technical difficulties in design or programming, or task analysis may indicate that some tasks assigned to humans exceed human capabilities under certain conditions.

During requirements definition, HF engineers ensure that HF-related goals and constraints are included in the overall plans for the system. The HF practitioner must identify the HF-related issues, design risks, and tradeoffs pertinent to each human-system component, and document these as part of the project’s requirements so they are adequately addressed during the design phase. For stakeholder expectation definition from the human factors perspective, the stakeholders include not only those who are specifying the system to be built, but also those who will be utilizing the system when it is put into operation. This approach yields requirements generated from the top down—what the system is intended to accomplish, and from the bottom up—how the system is anticipated to function. It is critical that the human factors practitioner contribute to the concepts of operations. The expectations of the role of the human in the system, the types of tasks the human is expected to perform, underlie all the hardware and software requirements. The difference between a passive passenger and an active operator will drive major design decisions. The number of crewmembers will drive subsequent decisions about habitable volume and storage, and about crew time available for operations and maintenance. HF practitioners ensure appropriate system design that defines the environmental range in which the system will operate and any factors which impact the human components. Many of these factors will need to accommodate human, as well as machine tolerances. The requirements may need to specify acceptable atmospheric conditions, including temperature, pressure, composition, and humidity, for example. The requirements might also address acceptable ranges of acoustic noise, vibration, acceleration, and gravitational forces, and the use of protective clothing. The requirements may also need to accommodate adverse or emergency conditions outside the range of normal operation.

### ***7.3.1 Basic Human Factors Model***

A key to conducting human and organizational analysis, design, and testing is to have an explicit framework that relates and scopes the work in question. The following model identifies the boundaries and the components involved in assessing human impacts.

The Human Factors Interaction Model (Figure 7-4) provides a reference point of items to be aware of in planning, analyzing, designing, testing, operating, and maintaining systems. Detailed checklists should be generated and customized for the particular system under development. The model presented in this module is adapted from David Meister’s, *Human Factors: Theory and Practice*, 1971, and is one depiction of how humans and systems interact. Environmental influences on that interaction have been added. The model illustrates a typical information flow between the “human” and “machine” components of a system.



**Figure 7-4 Human Factors Interaction Model.**

### **7.3.2 Human Factors Analysis and Evaluation Techniques**

Table 7-7 below provides a set of techniques for human and organizational analysis and evaluation to ensure that appropriate human and organizational factors have been considered and accommodated. The major methods used by trained human factors professionals analyze systems, provide data about human performance, make predictions about human-system performance, and evaluate if the human-machine system performance meets design criteria. Most methods involve judgment and so are highly dependent on the skill and expertise of the analyst. In addition, both experienced and inexperienced operators provide valuable information about the strengths and weaknesses of old systems and how the new system might be used.

These methods are used through all phases of system design but with increasing specificity and detail as development progresses. While human factors principles are often researched and understood at a generic level, their application is only appropriate when tailored to fit the design phase. Each type of analysis yields different kinds of information and so are not interchangeable. The outputs or products of the analyses go into specification documents (operational need document, concept of operations, system requirements document, etc.) and formal review processes (e.g., operational requirements review, system-requirements review, system-design review, preliminary design review, critical-design review, production-readiness review, product improvement review).

The list shown in Table 7-7 is not exhaustive. The main point is to show examples that demonstrate the scope and usefulness of common methods used to evaluate system design and development.

**Table 7-7 Human and Organizational Analysis Techniques**

Process	Human/Individual Analysis	Additional Organizational Analysis
<b>A. Operational Analysis</b>		
Definition	Analysis of projected operations.	
Purpose	Obtain information about situations or events that may confront operators and maintainers using the new system. Systems engineers or operations analysts have typically done operational analyses. Trained human factors professionals should also be members of the analysis team to capture important operator or maintainer activities.	
Inputs	RFPs, planning documents, system requirements documents, and expert opinion	
Process	Consult the systems engineer and projected users to extract implications for operators and maintainers.	Assess interactions and logistics between individuals and organizations. Evaluate operations under different types of organizations, structures, or distributions.
Outputs	Detailed scenarios (for nominal operations, hard and soft failures, and emergencies) including consequences; verbal descriptions of events confronting operators and maintainers; anticipated operations (list of feasible operations and those that may overstress the system); assumptions; constraints that may affect system performance; environments; list of system operation and maintenance requirements.	New or adjusted workflows to compensate for organizational impacts as appropriate.
<b>B. Similar Systems Analysis</b>		
Definition	Examine previous systems or systems in use for information useful for the new system.	
Purpose	To obtain lessons learned and best practices useful in planning for a new system. Experiences gained from systems in use is valuable information that should be capitalized on.	
Inputs	Structured observations, interviews, questionnaires, activity analysis, accident/incident reports, maintenance records, and training records.	
Process	Obtain data on the operability, maintainability, and number of people required to staff the system in use. Identify skills required to operate and maintain the system and training required to bring operators to proficiency. Obtain previous data on HF design problems and problems encountered by previous users of the previous system or system in use.	Identify the existing system's organizational hierarchies and management distribution schemes (centralized vs. decentralized).
Outputs	Identification of environmental factors that may affect personnel; preliminary assessments of workload and stress levels; assessment of skills required and their impact on selection, training and design; estimates of future staffing and manpower requirements; identification of operator and maintainer problems to avoid; assessment of desirability and consequences of reallocation of systems functions.	Evaluation of the configuration's impact to performance and its potential risks.
<b>C. Critical Incident Study</b>		
Definition	Identification of sources of difficulties for operators or maintenance or in the operational systems (or simulations of them).	
Purpose	To analyze and hypothesize sources of errors and difficulties in a system. This is particularly useful when a system has been operational and difficulties are observed or suspected, but the nature and severity of those difficulties is not known.	
Inputs	Operator/maintainer accounts of accidents, near-accidents, mistakes, and near-mistakes.	
Process	Interview large numbers of operators/maintainers; categorize incidents/accidents; use HF knowledge and experience to hypothesize sources of difficulty and how each one could be further studied; mitigate or redesign to eliminate difficulty.	Trace difficulties between individuals and organizations and map associated responsibilities and process assignments.

Process	Human/Individual Analysis	Additional Organizational Analysis
Outputs	Sources of serious HF difficulties in the operation of a system or its maintenance with suggested solutions to those difficulties.	Identification of potential gaps or disconnects based on the mapping.
<b>D. Functional Flow Analysis</b>		
Definition	A structured technique for determining system requirements. Decomposes the sequence of functions or actions that a system must perform.	
Purpose	Provides a sequential ordering of functions that will achieve system requirements and a detailed checklist of system functions that must be considered in assuring that the system will be able to perform its intended mission. These functions are needed for the solution of tradeoff studies and determinations of their allocation among operators, equipment, software, or some combination of them. Decision-action analysis is often used instead of a functional flow analysis when the system requires binary decisions (e.g., software oriented).	
Inputs	Operational analyses, analyses of similar systems, activity analyses.	
Process	Top-level functions are progressively expanded to lower levels containing more and more detailed information. If additional elaboration is needed about information requirements, sources of information, potential problems, and error-inducing features, for example, then an action-information analysis is also performed.	Map functional flows to associated organizational structures.
Outputs	Functional flow diagrams.	Identification of any logistics or responsibility gaps based on the integrated map.
<b>E. Action-Information Analysis</b>		
Definition	Elaborates each function or action in functional flows or decision-action diagrams by identifying the information that is needed for each action or decision to occur. This analysis is often supplemented with sources of data, potential problems, and error-inducing features associated with each function or action.	
Purpose	Provides more detail before allocating functions to agents.	
Inputs	Data from the analysis of similar systems, activity analyses, critical incident studies, functional flow and decision-action analyses, and comments and data from knowledgeable experts.	
Process	Each function or action identified in functional flows or decision-action analyses is elaborated.	Map associated components (function, action, decisions) to the responsible organizational structures.
Outputs	Detailed lists of information requirements for operator-system interfaces, early estimates of special personnel provisions likely to be needed, support requirements, and lists of potential problems and probable solutions. Often produces suggestions for improvements in design of hardware, software, or procedures.	Identification of any logistics or responsibility gaps based on the integrated map.
<b>F. Functional Allocation</b>		
Definition	Procedure for assigning each system function, action, and decision to hardware, software, operators, maintainers, or some combination of them.	
Purpose	To help identify user skill needs and provide preliminary estimates of staffing, training, and procedures requirements and workload assessments. Functional flows and decision-action analyses do not identify the agent (person or machine) that will execute the functions.	
Inputs	Functional flow analyses, decision-action analyses, action-information analyses, past engineering experience with similar systems, state-of-the-art performance capabilities of machines and software, and store of known human capabilities and limitations.	
Process	Identify and place to the side all those functions that must be allocated to personnel or equipment for reasons of safety, limitations of engineering technology, human limitations, or system requirements; list the remaining functions—those that could be either performed manually or by some combination of personnel and equipment; prepare descriptions of implementation; establish weighting criteria for each design alternative; and compare alternative configurations in terms of their effectiveness in performing the given function according to those criteria.	After initial assignment configuration is completed, evaluate allocations against relevant organizational norms, values, and organizational interfaces for logistics and management impacts.
Outputs	Allocations of system functions to hardware, software, operators,	List of potential impacts with

Process	Human/Individual Analysis	Additional Organizational Analysis
	maintainers, or some combination of them. Task analyses are then performed on those functions allocated to humans.	recommended modifications in either functions or management or both.
<b>G. Task Analysis</b>		
Definition	Method for producing an ordered list of all the things people will do in a system.	
Purpose	To develop input to all the analyses that come next. A subsequent timeline analysis chart can provide the temporal relationship among tasks—sequences of operator or maintainer actions, the times required for each action, and the time at which each action should occur.”	
Inputs	Data from all the methods above supplemented with information provided by experts who have had experience with similar systems.	
Process	Trained human factors professional and subject matter experts list and describe all tasks, subdividing them into subtasks with the addition of supplementary information.	Group all tasks assigned to a given organization and evaluate the range of skills, communications, and management capabilities required. Evaluate new requirements against existing organization’s SoPs, norms, and values.
Outputs	Ordered list of all the tasks people will perform in a system. Details on information requirements, evaluations and decisions that must be made, task times, operator actions, and environmental conditions.	Identify group level workloads, management impacts, and training requirements.
<b>H. Fault Tree Analysis</b>		
Definition	Determines those combinations of events that could cause specific system failures, faults, or catastrophes. Fault tree and FMEA are concerned with errors.	
Purpose	Anticipate mistakes that operators or maintainers might make and try to design against those mistakes. Limitation for human factors are that each event must be described in terms of only two possible conditions, and it is extremely difficult to attach exact probabilities to human activities.	
Inputs	All outputs of the methods described above, supplemented with data on human reliability.	
Process	Construct a tree with symbols (logic gates) to represent events and consequences and describe the logical relationship between events.	Anticipate mistakes and disconnects that may occur in the workflow between individuals and organizations including un-anticipated interactions between standard organization SoPs and possible system events.
Outputs	Probabilities of various undesirable workflow related events, the probable sequences that would produce them and the identification of sensitive elements that could reduce the possibility or probability of a mishap.	Probabilities of various undesirable workflow related events arranged by organizational interface points for the workflows, the probable sequences that would produce them, and the identification of sensitive elements that could reduce the possibility or probability of a mishap.
<b>I. Failure Modes and Effects Analysis</b>		
Definition	Effective method of reducing or eliminating error-inducing features in a system.	
Purpose	Deduce the consequences for system performance of a failure in one or more components (operators and maintainers) and the probabilities of those consequences occurring.	
Inputs	“All outputs of methods described above, supplemented with data on human reliability.”	
Process	Analyst identifies the various errors operators or maintainers may make in carrying out subtasks or functions. Estimates are made of the probabilities or frequencies of making each kind of error. The consequences of each kind of error are deduced by tracing its effects through a functional flow diagram to its final outcome.	Identify possible organizational entities and behaviors (i.e., political) involved with the system. Estimate the probabilities of occurrence and the impact of consequence.
Outputs	List of human failures that would have critical effects on system operation, the probabilities of system or subsystem failures due to human errors, and identification of those human tasks or actions that should be modified or	List organizational behaviors that would have a critical effect on the system operation, the probabilities of the system

Process	Human/Individual Analysis	Additional Organizational Analysis
	replaced to reduce the probability of serious system failures.	or subsystem failures due to the organizational behaviors, and those organizational values, culture, or actions/SoPs that should be modified or replaced to reduce the probability of serious system failures.
<b>J. Link Analysis</b>		
Definition	A design activity (not an analysis) for arranging the physical layout of instrument panels, control panels, workstations, or work areas to meet certain objectives.	
Purpose	To consider the most efficient and the effectiveness of the physical layout of the human-machine interface.	
Inputs	Data from activity and task analysis and observations of functional or simulated systems.	
Process	"List all personnel and items. Estimate frequencies of linkages between items, operators, or items and operators. Estimate the importance of each link. Compute frequency-importance values for each link. Starting with the highest link values, successively add items with lower link values and readjust to minimize linkages. Fit the layout into the allocated space. Evaluate the new layout against the original objectives."	Assess links at individual vs. organizational levels.
Outputs	Recommended layouts of panels, workstations, or work areas.	Adjusted layouts of panels, workstations, or work areas based on optimum individual and organizational performance priorities.
<b>K. Simulation</b>		
Definition	Basic engineering or human factors method to predict performance of systems. Includes usability testing and prototyping.	
Purpose	To predict the performance of systems, or parts of systems that do not exist, or to allow users to experience and receive training on systems or parts of systems that are complex, dangerous, or expensive.	
Inputs	Hardware, software, functions, and tasks elucidated in task analysis, operating procedures.	
Process	Users perform typical tasks on models or mockups prepared to incorporate some or all of the inputs.	Assess individual performance within possible organizational models.
Outputs	Predictions about system performance; assessment of workloads; evaluation of alternative configurations; evaluation of operating procedures; training; identification of accident- or error-provocative situations and mismatches between personnel and equipment.	Predict system performance under varied organizational conditions; assess workloads; evaluate alternative configurations and operating procedures; train; identify accident- or error-provocative situations and mismatches between personnel and equipment.
<b>L. Controlled Experimentation</b>		
Definition	Highly controlled and structured version of simulation with deliberate manipulation of some variables.	
Purpose	To answer one or more hypothesis and narrow number of alternatives used in simulation.	
Inputs	From any or all methods listed thus far.	
Process	Select experimental design; identify dependent, independent, and controlled variables; set up test, apparatus, facilities and tasks; prepare test protocol and instructions; select subjects; run tests; and analyze results statistically.	Scale to organizational levels where appropriate and feasible.
Outputs	Quantitative statements of the effects of some variables on others and differences between alternative configurations, procedures, or environments.	Quantitative statements of the effects of some organizational variables on others and differences between alternative organizational configurations, procedures, or environments.
<b>M. Operational Sequence Analysis</b>		
Definition	Powerful technique used to simulate systems.	
Purpose	To permits visualization of interrelationships between operators and	

Process	Human/Individual Analysis	Additional Organizational Analysis
	operators and equipment, identify interface problems, and explicitly identify decisions that might otherwise go unrecognized. Less expensive than mockups, prototypes, or computer programs that attempt to serve the same purpose.	
Inputs	Data from all above listed methods.	
Process	Diagram columns show timescale, external inputs, operators, machines, external outputs. Flow of events (actions, functions, decisions) is then plotted from top to bottom against the timescale using special symbology.	After initial analysis is complete, group results by responsible organizations.
Outputs	Time-based chart showing the functional relationships among system elements, the flow of materials or information, the physical and sequential distribution of operations, the inputs and outputs of subsystems, the consequences of alternative design configurations, potential sources of human difficulties.	Assessment of range of skills, communication processes, and management capabilities required. Evaluation of performance under various organizational structures.
<b>N. Workload Assessment</b>		
Definition	Procedure for appraising operator and crew task loadings or the ability of personnel to carry out all assigned tasks in the time allotted or available.	
Purpose	To keep operator workloads at reasonable levels and to ensure that workloads are distributed equitably among operators.	
Inputs	Task time, frequency, and precision data are obtained from many of the above listed methods supplemented with judgments and estimates from knowledgeable experts.	
Process	The DOD-KDBK-763 recommends a method that estimates the time required to perform a task divided by the time available or allotted to perform it. There are three classes of methods: performance measures, physiological measures, and subjective workloads either during or after an activity.	Same as above.
Outputs	Quantitative assessments of estimated workloads for particular tasks at particular times.	Same as above.
<b>O. Situational Awareness</b>		
Definition	Procedure for appraising operator and crew awareness of tasks and current situation.	
Purpose	To maintain operator and maintainer awareness to maintain safety and efficiency.	
Inputs	All of the above listed analyses.	
Process	Different methods have been proposed including: Situation Awareness Rating Technique (SART), Situation awareness behaviorally anchored rating scale (SABARS), Situation Awareness Global Assessment Technique (SAGAT), Situation Awareness Verification and Analysis Tool (SAVANT).	Collect organizational decisionmaking structures and processes and map the organization's situational awareness profiles.
Outputs	Quantitative estimates of situational awareness for particular tasks at particular times.	Identification of possible gaps, disconnects, and shortfalls.
<b>P. Performance Modeling</b>		
Definition	Computational process for predicting human behavior based on current cognitive research.	
Purpose	To predict human limitations and capabilities before prototyping.	
Inputs	All of the above listed analyses.	
Process	Input results from above analyses. Input current relevant environmental and machine parameters. Can be interleaved with fast-time simulation to obtain frequency of error types.	Scale as appropriate to relevant organizational behaviors.
Outputs	Interrelationships between operators and operators and equipment and identification of interface problems and decisions that might otherwise go unrecognized.	Interrelationships between individuals and organizations and identification of organizational interface problems and decisions that might otherwise go unrecognized.

## **7.4 Environmental, Nuclear Safety, Planetary Protection, and Asset Protection Policy Compliance**

### ***7.4.1 National Environmental Policy Act and Executive Order 12114 Environmental Effects Abroad of Major Federal Actions***

#### *7.4.1.1 National Environmental Policy Act*

The National Environmental Policy Act (NEPA) declares the basic national policy for protecting the human environment. NEPA sets the Nation's goals for enhancing and preserving the environment. NEPA also provides the procedural requirements to ensure compliance by all Federal agencies. NEPA compliance can be a critical path item in project or mission implementation. NEPA requires all Federal agencies to consider, before an action is taken, environmental values in the planning of actions and activities that may have a significant impact upon the quality of the human environment. NEPA directs agencies to consider alternatives to their proposed activities. In essence, NEPA requires NASA decision makers to integrate the NEPA process into early planning to ensure appropriate consideration of environmental factors, along with technical and economic ones. NEPA is also an environmental disclosure statute. It requires that available information be adequately addressed and made available to the NASA decision makers in a timely manner so he or she can consider the environmental consequences of the proposed action or activity before taking final action. Environmental information must also be made available to the public as well as to other Federal, State, and local agencies. NEPA does not require that the proposed action or activity be free of environmental impacts, be the most environmentally benign of potential alternatives, or be the most environmentally wise decision. NEPA requires the decision-maker to consider environmental impacts as one factor in the decision to implement an action.

NASA activities are implemented through specific Sponsoring Entities, such as NASA HQ, NASA Centers (including Component Facilities), Mission Directorates, Program, or Mission Support Offices. The lead officials for these entities, the Officials-in-Charge, have the primary responsibility for ensuring that the NEPA process is integrated into their organizations' project planning activities before the Sponsoring Entities implement activities and actions. The Sponsoring Entities also are responsible for ensuring that records management requirements are met. NEPA functions are not performed directly by lead officials. Each NASA Center has an Environmental Management Office (EMO), which is usually delegated the responsibility for implementing NEPA. The EMO performs the primary or working-level functions of the NEPA process, such as evaluating proposed activities, developing and/or reviewing and approving required documentation, advising project managers, and signing environmental decision documents on projects and programs having little or no environmental impact. However, ultimate responsibility for complying with NEPA and completing the process in a timely manner lies with the program or project manager. Since the EMO provides essential functional support to the Sponsoring Entity, and because its implementation responsibilities are delegated, the term Sponsoring Entity will be used throughout to include the implementing NEPA organization at any NASA facility. In cases where the Sponsoring Entity needs to be further defined, it will be specifically noted. For proposals made by tenants or entities using services or facilities at a NASA Center or Component Facility, the Sponsoring Entity shall be that Center or, if such authority is delegated to the Component Facility, the Component Facility.

NEPA compliance documentation must be completed before project planning reaches a point where NASA's ability to implement reasonable alternatives is effectively precluded (i.e., before hard decisions are made regarding project implementation). Environmental planning factors should be integrated into the Pre-Phase A concept study phase when a broad range of alternative approaches is being considered. In the Phase A concept development stage, decisions are made that could affect the Phase B preliminary design stage. At a minimum, an environmental evaluation should be initiated in the Phase A concept development stage. During this stage, the responsible project manager will have the greatest latitude in making adjustments in the plan to mitigate or avoid important environmental sensitivities and in planning the balance of the NEPA process to avoid unpleasant surprises later in the project cycle which may have schedule and/or cost implications. Before completing the NEPA process, no NASA official can take an action that would (1) affect the environment or (2) limit the choice of reasonable alternatives.

Accommodating environmental requirements early in project planning ultimately conserves both budget and schedule. Further detail regarding NEPA compliance requirements for NASA programs and projects can be found in *NPR 8580.1, Implementing The National Environmental Policy Act And Executive Order 12114*.

#### *7.4.1.2 Executive Order 12114 Environmental Effects Abroad of Major Federal Actions*

Executive Order (EO) 12114 was issued "solely for the purpose of establishing internal procedures for Federal agencies to consider the significant effects of their actions on the environment outside the United States, its territories and possessions." The EO also specifically provided that its purpose is to enable the decisionmakers of the Federal agencies to be informed of pertinent environmental considerations, and factor such considerations in their decisions; however, such decisionmakers must still take into account considerations such as foreign policy, national security, and other relevant special circumstances.

The NASA Office of the General Counsel, or designee, is the NASA point of contact and official NASA representative on any matter involving EO 12114. Accordingly, any action by, or any implementation or legal interpretation of EO 12114 requires consultations with and the concurrence of the designee of the Office of the General Counsel. The Sponsoring Entity and local Environmental Management Office (EMO) contemplating an action which would have global environmental effects or effects outside the territorial jurisdiction of the U.S. must notify the NASA Headquarters/Environmental Management Division (HQ/EMD). The HQ/EMD will, in turn, coordinate with the Office of the General Counsel, the Assistant Administrator for External Relations, and other NASA organizations as appropriate; and assist the Sponsoring Entity to develop a plan of action, (such a plan is subject to the concurrence of the Office of the General Counsel). Further detail regarding EO 12114 compliance requirements for NASA Programs and Projects can be found in NPR 8580.1.

#### *7.4.2 Presidential Directive/National Security Council Memorandum #25*

NASA has procedural requirements for characterizing and reporting potential risks associated with a planned launch of radioactive materials into space, on launch vehicles and spacecraft, during normal or abnormal flight conditions. Procedures and levels of review and analysis required for nuclear launch safety approval vary with the quantity of radioactive material planned for use and potential risk to the general public and the environment. Specific details

concerning these requirements can be found in *NPR 8715.3, NASA General Safety Program Requirements*.

For any U.S. space mission involving the use of radioisotope power systems, radioisotope heating units, nuclear reactors, or a major nuclear source, launch approval must be obtained from the Office of the President per Presidential Directive/National Security Council Memorandum No. 25 (PD/NSC-25), “Scientific or Technological Experiments with Possible Large-Scale Adverse Environmental Effects and Launch of Nuclear Systems into Space,” paragraph 9 as amended May 8, 1996. The approval decision is based on an established and proven review process that includes an independent evaluation by an ad hoc Interagency Nuclear Safety Review Panel (INSRP) comprised of representatives from NASA, the Department of Energy (DOE), the Department of Defense, and the Environmental Protection Agency, with an additional technical advisor from the Nuclear Regulatory Commission. The process begins with development of a launch vehicle databook (i.e., a compendium of information describing the mission, launch system, and potential accident scenarios including their environments and probabilities). DOE uses the databook to prepare a Preliminary Safety Analysis Report (PSAR) for the space mission. In all, three safety analysis reports (SARs) are typically produced and submitted to the mission’s INSRP—the PSAR, an updated SAR (draft final SAR), and a final SAR (FSAR). The DOE project office responsible for providing the nuclear power system develops these documents.

The ad hoc INSRP conducts its nuclear safety/risk evaluation and documents their results in a nuclear Safety Evaluation Report (SER). The SER contains an independent evaluation of the mission radiological risk. The DOE uses the SER as its basis for accepting the SAR. If the DOE Secretary formally accepts the SAR-SER package, it is forwarded to the NASA Administrator for use in the launch approval process.

NASA distributes the SAR and SER to the other cognizant government agencies involved in the INSRP, and solicits their assessment of the documents. After receiving responses from these agencies, NASA conducts internal management reviews to address the SAR and SER and any other nuclear safety information pertinent to the launch. If the NASA Administrator recommends proceeding with the launch, then a request for nuclear safety launch approval is sent to the Director of the Office of Science and Technology Policy (OSTP) within the Office of the President.

NASA Headquarters is responsible for implementing this process for NASA missions. It has traditionally enlisted JPL to assist in this activity. DOE supports the process by analyzing the response of RPS hardware to the different accident scenarios identified in the databook, and prepares a PRA of the potential radiological consequences and risks to the public and the environment for the mission. NASA KSC is responsible for overseeing development of databooks, and traditionally uses JPL to characterize accident environments and integrate databooks. Both KSC and JPL subcontractors provide information relevant to supporting the development of databooks. The development team ultimately selected for a mission would be responsible for providing payload descriptions, describing how the nuclear hardware integrates into the spacecraft, describing the mission, and supporting NASA KSC and JPL in their development of databooks.

Mission Directorate Associate Administrators, Center Directors, and program executives involved with the control and processing of radioactive materials for launch into space must ensure that basic designs of vehicles, spacecraft, and systems utilizing radioactive materials

provide protection to the public, the environment, and users such that radiation risk resulting from exposures to radioactive sources are as low as reasonably achievable. Nuclear safety considerations must be incorporated from the Pre-Phase A concept study stage throughout all project stages to ensure that the overall mission radiological risk is acceptable. All space flight equipment (including medical and other experimental devices) that contain or use radioactive materials must be identified and analyzed for radiological risk. Site-specific ground operations and radiological contingency plans must be developed commensurate with the risk represented by the planned launch of nuclear materials. Contingency planning, as required by the National Response Plan, includes provisions for emergency response and support for source recovery efforts. *NPR 8710.1, Emergency Preparedness Program*, and *NPR 8715.2, NASA Emergency Preparedness Plan Procedural Requirements*, address the NASA emergency preparedness policy and program requirements.

### **7.4.3 Planetary Protection**

The U.S. is a signatory to the United Nation's Treaty of Principles Governing the Activities of States in the Exploration and Use of Outer Space, Including the Moon and Other Celestial Bodies. Known as the "Outer Space" treaty, it states in part (Article IX) that exploration of the Moon and other celestial bodies shall be conducted "so as to avoid their harmful contamination and also adverse changes in the environment of the Earth resulting from the introduction of extraterrestrial matter." NASA policy (*NPD 8020.7, Biological Contamination Control for Outbound and Inbound Planetary Spacecraft*) specifies that the purpose of preserving solar system conditions is for future biological and organic constituent exploration. This NPD also establishes the basic NASA policy for the protection of the Earth and its biosphere from planetary and other extraterrestrial sources of contamination. The general regulations to which NASA flight projects must adhere are set forth in *NPR 8020.12, Planetary Protection Provisions for Robotic Extraterrestrial Missions*. Different requirements apply to different missions, depending on which solar system object is targeted or encountered and the spacecraft or mission type (flyby, orbiter, lander, sample-return, etc.). For some bodies (such as the Sun, Moon, and Mercury), there are minimal planetary protection requirements. Current requirements for the outbound phase of missions to Mars and Europa, however, are particularly rigorous. Table 7-8 shows the current planetary protection categories, while Table 7-9 provides a brief summary of their associated requirements.

**Table 7-8 Planetary Protection Mission Categories**

Planet Priorities	Mission Type	Category	Example
Not of direct interest for understanding the process of chemical evolution. No protection of such planets is warranted (no requirements)	Any	I	Lunar missions
Of significant interest relative to the process of chemical evolution, but only a remote chance that contamination by spacecraft could jeopardize future exploration.	Any	II	Stardust (outbound) Genesis (outbound) Cassini
Of significant interest relative to the process of chemical evolution and/or the origin of life or for which scientific opinion provides a significant chance of contamination which could jeopardize a future biological experiment.	Flyby, Orbiter	III	Odyssey MGS MRO
	Lander, Probe	IV	MER Phoenix Europa Explorer MSR (outbound)
Any solar system body	Unrestricted Earth return	V	Stardust (return) Genesis (return)
	Restricted Earth return	V	MSR (return)

**Table 7-9 Summarized Planetary Protection Requirements**

Mission Category	Summarized Requirements
I	Certification of category
II	Avoidance of accidental impact by spacecraft and launch vehicle. Documentation of final disposition of launched hardware.
III	Stringent limitations on the probability of impact. Requirements on orbital lifetime or requirements for microbial cleanliness of spacecraft
IV	Stringent limitations on the probability of impact and/or the contamination of the object. Microbial cleanliness of landed hardware surfaces directly established by bioassays.
V	Outbound requirements per category of a lander mission to the target. Detailed restricted Earth return requirements will depend on many factors, but will likely include sterilization of any hardware that contacted the target planet before its return to Earth, and the containment of any returned sample.

At the core, planetary protection is a systems engineering activity; the effort cuts across multiple WBS elements and failure to adopt and incorporate a viable planetary protection approach during the early planning phases will add cost and complexity to the mission. Planning for planetary protection begins in Phase A, during which feasibility of the mission is established. Prior to the end of Phase A, the project manager must send a letter to the Planetary Protection Officer (PPO)

stating the mission type and planetary targets, and requesting that the mission be assigned a planetary protection category.

Prior to the Preliminary Design Review (PDR), at the end of Phase B, the project manager must submit to the NASA PPO a Planetary Protection Plan detailing the actions that will be taken to meet the requirements. The project's progress and completion of the requirements are reported in a Planetary Protection Pre-Launch Report submitted to the NASA PPO for approval. The approval of this report at the Flight Readiness Review (FRR) constitutes the final Planetary Protection approval for the project and must be obtained for permission to launch. An update to this report, the Planetary Protection Post-Launch Report, is prepared to report any deviations from the planned mission due to actual launch or early mission events. For sample return missions only, additional reports and reviews are required: prior to launch toward the Earth, prior to commitment to Earth reentry, and prior to the release of any extraterrestrial sample to the scientific community for investigation. Finally, at the formally declared end-of-mission, a Planetary Protection End-of-Mission Report is prepared. This document reviews the entire history of the mission in comparison to the original Planetary Protection Plan, and documents the degree of compliance with NASA's planetary protection requirements. This document is typically reported on by the NASA PPO at a meeting of the Committee on Space Research (COSPAR) to inform other spacefaring nations of NASA's degree of compliance with international planetary protection requirements.

#### **7.4.4 *Space Asset Protection***

The terrorist attacks on the World Trade Center in New York and on the Pentagon on September 11th, 2001 have created an atmosphere for greater vigilance on the part of government agencies to ensure that sufficient security is in place to protect their personnel, physical assets and information, especially those assets that contribute to the political, economic and military capabilities of the United States. Current trends in technology proliferation, accessibility to space, globalization of space programs and industries, commercialization of space systems and services, and foreign knowledge about U.S. space systems increases the likelihood that vulnerable US space systems may come under attack, particularly vulnerable systems. The ability to restrict or deny freedom of access to and operations in space is no longer limited to global military powers. The reality is that there are many existing capabilities to deny, disrupt or physically destroy orbiting spacecraft and the ground facilities that command and control them. Knowledge of U.S. space system's functions, locations and physical characteristics, as well as the means to conduct counter-space operations is increasingly available on the international market. Nations or groups hostile to the U.S. either possess or can acquire the means to disrupt or destroy our space systems by attacking our satellites in space, their communications nodes on the ground and in space, the ground nodes that command these satellites or process their data and the commercial infrastructure that supports a space system's operations.

##### **7.4.4.1 *Protection Policy***

The new National Space Policy authorized by the President on August 31, 2006, states that space capabilities are vital to the nation's interests and that the US will "take those actions necessary to protect its space capabilities". The policy also gives responsibility for space situational awareness (SSA) to the Secretary of Defense (SECDEF). In that capacity the SECDEF will conduct space situational awareness for civil space capabilities and operations, particularly human space flight activities. SSA provides an in-depth knowledge and understanding of the

threats posed to U.S., allied, and coalition space systems by adversaries and the environment, and is essential in developing and employing protection measures. Therefore, NASA's space asset protection needs will drive the requirements that the NASA levies on the Department of Defense (DOD) for space situational awareness.

#### *7.4.4.2 Goal*

The overall space asset protection goal for NASA is to support sustained mission assurance through the reduction of susceptibilities and the mitigation of vulnerabilities, relative to risk, and within fiscal constraints.

#### *7.4.4.3 Scoping*

Space asset protection involves the planning and implementation of measures to protect NASA space assets from intentional or unintentional disruption, exploitation or attack, whether natural or man-made. It is essential that protection is provided for all segments of a space system (ground, communications/information, space and launch) and covers the entire life cycle of a project. Space asset protection includes aspects of personnel, physical, information, communications, information technology, and operational security, as well as counter-intelligence activities. The role of the systems engineer is to integrate security competencies with space systems engineering and operations expertise to develop mission protection strategies consistent with payload classifications as defined in *NPR 8705.4, Risk Classification for NASA Payloads*.

#### *7.4.4.4 Protection Planning*

Systems engineers use protection planning processes and products (which include engineering trade studies and cost/benefit analyses) to meet NASA's needs for acquiring, fielding and sustaining secure and uncompromised space systems. Project Protection Plans (PPP) are single source documents that coordinate and integrate protection efforts and prevent inadvertent/uncontrolled disclosure of sensitive program information. Protection plans provide project management personnel (project manager, project scientist, mission systems engineer, operations manager, user community, etc.) with an overall view of the valid threats to a space system (both hostile and environmental), identifies infrastructure vulnerabilities and proposes security countermeasures to mitigate risks and enhance survivability of the mission. A outline for a typical PPP can be found in appendix 7-4.

## 7.5 Use of Metric System

The decision whether a project or program could or should implement the System Internationale (SI), often called the “metric” system, requires consideration of a number of factors—cost, technical, risk and other programmatic aspects.

First is the understanding that there is a national goal of establishing the metric system as the preferred system of weights and measures for U.S. trade and commerce. [Metric Conversion Act of 1975], [Omnibus Trade and Competitiveness Act of 1988] [find reference]. NASA has developed *NPD 8010.2, Use of the SI (Metric) System of Measurement in NASA Programs*, which implements SI and provides specific requirements and responsibilities for NASA.

However, a second factor to consider is that there are possible exceptions to the required implementation approach. Both Executive Order (EO) 12770 and NPD 8010.2 allow exceptions and, because full SI implementation may be difficult, allow the use of “hybrid” systems. Consideration of the following factors will have a direct impact on the implementation approach and use of exceptions by the program or project.

Programs or projects must do analysis during the early life-cycle phases when the design solutions are being developed, to identify where SI is feasible or recommended and where exceptions will be required. A major factor to consider is the capability to actually produce or provide metric-based hardware components. Results and recommendations from these analyses must be presented by SRR for approval.

In planning program or project implementation to produce metric-based systems, issues to be addressed should include the following:

- Interfaces with heritage components (e.g., valves, pyrotechnic devices, etc.) built to English-based units:
  - Necessitate conversion from English to SI and/or interface to English-based hardware.
  - Team should review design implementation to ensure there is no certification impact with heritage hardware or identify and plan for any necessary recertification efforts.
- Dimensioning and tolerancing:
  - Can result in parts which do not fit.
  - Rounding errors have occurred when converting units from one unit system to the other.
  - The team may require specific additional procedures, steps, and drawing QA personnel when converting units.
- Tooling:
  - Not all shops have full metric tooling (e.g., drill bits, taps, end mills, reamers, etc.).
  - The team needs to inform potential contractors of intent to use SI and obtain feedback as to potential impacts.

- Fasteners and miscellaneous parts:
  - High-strength fastener choices and availability are more limited in metric sizes.
  - Bearings, pins, rod ends, bushings, etc. are readily available in English with minimal lead times.
  - Team needs to ascertain availability of acceptable SI-based fasteners in the timeframe needed.
- Reference material:
  - Some key aerospace reference materials are built only in English units, e.g., MIL-HDBK-5 (metallic material properties), and values will need to be converted when used.
  - Other key reference materials or commercial databases are built only in SI units.
  - The team needs to review the reference material to be used and ensure acceptable conversion controls are in place, if necessary.
- Corporate knowledge:
  - Many engineers presently “think” in English units, i.e., can relate to pressure in PSI, can relate to material strength in KSI, can relate to a tolerance of .003 inches, etc.
  - However, virtually all engineers coming out of school in this day and era presently “think” in SI units and have difficulty relating to English-based units such as slugs (for mass) and would require retraining with attendant increase in conversion errors.
  - The team needs to be aware of their program or project-specific knowledge in English and SI-units and obtain necessary training and experience.
- Industry practices:
  - Certain industries work exclusively in English units, and sometimes have their own jargon associated with English material properties. The parachute industry falls in this category, e.g., “600-lb braided Kevlar line.”
  - Other industries, especially international suppliers, may work exclusively in metric units such as “30-mm-thick raw bar stock.”
  - The team needs to be aware of these unique cases and ensure both procurement and technical design and integration have the appropriate controls to avoid errors.
- Program or project controls:
  - The team needs to consider, early in the SE process, what program- or project-specific risk management controls (such as configuration management steps) are required. This will include such straightforward concerns as the conversion(s) between system elements that are in English units and those in SI units or other, more complex, issues.

Several NASA projects have taken the approach of using both systems, which is allowed by NPD 8010.2. For example, the Mars soil drill project designed and developed their hardware

using English-based components, while accomplishing their analyses using SI-based units. Other small-scale projects have successfully used a similar approach.

For larger or more dispersed projects or programs, a more systematic and complete risk management approach may be needed to successfully implement an SI-based system. Such things as standard conversion factors (e.g., from pounds to kilograms) should be documented, as should standard SI nomenclature. Many of these risk management aspects can be found in such documents as the National Institute of Standards and Technology (NIST) Special Publications (SP) 811, *Guide for the Use of the International System of Units (SI)* and the DoD *Guide for Identification and Development of Metric Standards*.

Until the Federal Government and the aerospace industrial base are fully converted to an SI-based unit system, the various NASA programs and projects will have to address their own level of SI implementation on a case-by-case basis. It is the responsibility of each NASA program and project management team, however, to comply with all laws and EOs while still maintaining a reasonable level of risk for cost, schedule, and performance.

## 7.6 Selecting Engineering Design Tools

NASA utilizes cutting-edge design tools and techniques to create the advanced analyses, designs, and concepts required to develop unique aerospace products, spacecraft and science experiments. The diverse nature of the design work generated and overseen by NASA requires use of a broad spectrum of robust electronic tools such as Computer-Aided Design (CAD) tools and Computer-Aided Systems Engineering (CASE) tools. Based on the distributed and varied nature of NASA projects, selection of a single suite of tools from only one vendor to accomplish all design tasks is not practical. However, opportunities to improve standardization of design policy, processes, and tools remain a focus for continuous improvement activities at all levels within the Agency.

These guidelines serve as an aid to help in the selection of appropriate tools in the design and development of aerospace products and space systems and when selecting tools that affect multiple Centers.

### 7.6.1 Program and Project Considerations

When selecting a tool to support a program or project, all of the upper level constraints and requirements must be identified early in the process. Pertinent information from the project that affects the selection of the tools will include the urgency, schedule, resource restrictions, extenuating circumstances, and constraints. A tool that does not support meeting the program master schedule or is too costly to be bought in sufficient numbers will not satisfy the project manager's requirements. For example, a tool that requires extensive modification and training that is inconsistent with the master schedule should not be selected by the technical team. If the activity to be undertaken is an upgrade to an existing project, legacy tools and availability of trained personnel are factors to be considered.

### 7.6.2 Policy and Processes

When selecting a tool, one must consider the applicable policies and processes at all levels, including those at the Center level, within programs and projects, and at other Centers when a program or project is a collaborative effort. In the following discussion the term "organization" will be used to represent any controlling entity that establishes policy and/or process for the use of tools in the design or development of NASA products. In other words, "organization" can mean the user's Center, another collaborating Center, a program, a project, in-line engineering groups, or any combination of these entities.

Policies and processes affect many aspects of a tool's functionality. First and foremost, there are policies that dictate how designs are to be formally or informally controlled within the organization. These policies address configuration management processes that must be followed as well as the type of data object that will be formally controlled (e.g., drawings or models). Clearly this will affect the types of tools that will be used and how their designs will be annotated and controlled.

The Information Technology (IT) policy of the organization also needs to be considered. Data security and export control (e.g., International Traffic in Arms Regulation (ITAR)) policies are two important IT policy considerations that will influence the selection of a particular design tool.

The policy of the organization may also dictate requirements on the format of the design data that is produced by a tool. A specific format may be required for sharing information with collaborating parties. Other considerations are the organizations' quality processes, which control the versions of the software tools as well as their verification and validation. There are also policies on training and certifying users of tools supporting critical flight programs and projects. This is particularly important when the selection of a new tool results in the transition from a legacy tool to a new tool. Therefore, the quality of the training support provided by the tool vendor is an important consideration in the selection of any tool.

Also, if a tool is being procured to support a multi-Center program or project, then program policy may dictate which tool must be used by all participating Centers. If Centers are free to select their own tool in support of a multi-Center program or project, then consideration of the policies of all the other Centers must be taken into account to ensure compatibility between Centers.

### ***7.6.3 Collaboration***

The design process is highly collaborative due to the complex specialties that must interact to achieve a successful integrated design. Tools are an important part of a successful collaboration. To successfully select and integrate tools in this environment requires a clear understanding of the intended user community size, functionality required, and knowledge of tools to be used. These factors will dictate the number of licenses, hosting capacity, tool capabilities, and training required. The sharing of common models across a broad group requires mechanisms for advancing the design in a controlled way. Effective use of data management tools can help control the collaborative design by requiring common naming conventions, markings, and design techniques to ensure compatibility among distributed design tools.

### ***7.6.4 Design Standards***

Depending on the specific domain or discipline there may be industry and Center-specific standards that must be followed, particularly when designing hardware. This can be evident in the design of a mechanical part, where a mechanical CAD (MCAD) package selected to model the parts must have the capability to meet specific standards, such as model accuracy, dimensioning and tolerancing, the ability to create different geometries, and the capability to produce annotations describing how to build and inspect the part. However these same issues must be considered regardless of the product.

### ***7.6.5 Existing IT Architecture***

As with any new tool decision, an evaluation of defined Agency and Center IT architectures should be made that focuses on compatibility with and duplication of existing tools. Typical architecture considerations would include data management tools, middleware or integration infrastructure, network transmission capacity, design analysis tools, manufacturing equipment, approved hosting, and client environments.

While initial focus is typically placed on current needs, the scalability of the tools and the supporting IT infrastructure should be addressed too. Scalability applies to both the number of users and capacity of each user to successfully use the system over time.

### **7.6.6 Tool Interfaces**

Information interfaces are ubiquitous, occurring whenever information is exchanged.

This is particularly characteristic of any collaborative environment. It is here that inefficiencies arise, information is lost, and mistakes are made. There may be an organizational need to interface with other capabilities and/or analysis tools and understanding the tools used by the design teams with which your team interfaces and how the outputs of your team drives other downstream design functions is critical to ensure compatibility of data.

For CASE tools, users are encouraged to select tools that are compatible with the Object Management Group System Modeling Language (SysML) standard. SysML is a version of the UML modeling language that has been specifically developed for systems engineering.

### **7.6.7 Interoperability and Data Formats**

Interoperability is an important consideration when selecting tools. The tools must represent the designs in formats that are acceptable to the end user of the data. It is important that any selected tool include associative data exchange and industry-standard data formats. As the Agency increasingly engages in multi-Center programs and projects, the need for interoperability among different tools, and different versions of the same tool, becomes even more critical. True interoperability reduces human error and the complexity of the integration task, resulting in reduced cost, increased productivity, and a quality product.

When considering all end users needs, it is clear that interoperability becomes a difficult challenge. Three broad approaches each with their own strengths and weaknesses are:

- Have all employees become proficient in a variety of different tool systems and the associated end use applications. While this provides a broad capability it may not be practical or affordable.
- Require interoperability among whatever tools are used, i.e., requiring that each tool be capable of transferring model data in a manner that can be easily and correctly interpreted by all the other tools. Considerable progress has been made in recent years in the standards for the exchange of model data. While this would be the ideal solution for many, standard data formats that contain the required information for all end users do not yet exist.
- Dictate that all participating organizations use the same version of the same tool.

### **7.6.8 Backward Compatibility**

On major programs and projects which span several years it is often necessary to access design data that is more than 3 to 5 years old. However, access to old design data can be extremely difficult and expensive, either because tool vendors end their support or later versions of the tool can no longer read the data. Strategies for maintaining access include special contracts with vendors for longer support, archiving design data in neutral formats, continuous migration of archives into current formats, and recreating data on demand. Organizations should select the strategy that works best for them, after a careful consideration of the cost and risk.

### **7.6.9 Platform**

While many tools will run on multiple hardware platforms, some perform better in specific environments or are only supported by specified versions of operating systems. In the case of open-source operating systems many different varieties are available that may not fully support the intended tools. If the tool being considered requires a new platform, the additional procurement cost and administration support costs should be factored in.

### **7.6.10 Tool Configuration Control**

Tool configuration control is a tradeoff between responsive adoption of the new capabilities in new versions and smooth operation across tool chain components. This is more difficult with heterogeneous (multiple vendor) tool components. An annual or biannual block upgrade strategy requires significant administrative effort. On the other hand, the desktop diversity resulting from user-managed upgrade timing also increases support requirements.

### **7.6.11 Security/Access Control**

Special consideration should be given to the sensitivity and required access of all design data. Federal Government and Agency policy requires the assessment of all tools to ensure appropriate security controls are addressed to maintain the integrity of the data.

### **7.6.12 Training**

Most of the major design tools have similar capabilities that will not be new concepts to a seasoned designer. However, each design tool utilizes different techniques to perform design functions and each contains some unique tool sets that will require training. The more responsive vendors will provide follow-up access to instructors and onsite training with liberal distribution of training materials and worked examples. The cost and time to perform the training and time for the designer to become proficient can be significant and should be carefully factored in when making decisions on new design tools.

The disruptive aspect of training is an important consideration in adapting to a different tool. Before transitioning to a new tool, an organization must consider the schedule of deliverables to major programs and projects. Can commitments still be met in a timely fashion? It is suggested that organizations implement a phase-in approach to a new tool, where the old tool is retained for some time to allow people to learn the new tool and become proficient in its use. The transition of a fully functional and expert team using any one system, to the same team fully functional using another system, is a significant undertaking. Some overlap between the old tool and the new tool will ensure flexibility in the transition and ensure that the program and project work proceeds uninterrupted.

### **7.6.13 Licenses**

Licenses provide and control access to the various modules or components of a product or product family. Consideration of the license scheme should be taken into account while selecting a tool package. Licenses are sometimes physical, like a hardware key that plugs into a serial or parallel port, or software that may or may not require a whole infrastructure to administer. Software licenses may be floating (able to be shared on many computers on a first-come, first-

served basis) or locked (dedicated to a particular computer). A well-thought-out strategy for licenses must be developed in the beginning of the tool selection process. This strategy must take into consideration program and project requirements and constraints as well as other factors such as training and use.

#### ***7.6.14 Stability of Vendor and Customer Support***

As in the selection of any support device or tool, vendor stability is of great importance. Given the significant investment in the tools (directly) and infrastructure (indirectly), it is important to look at the overall company stability to ensure the vendor will be around to support the tools. Maturity of company products, installed user base, training, and financial strength can all provide clues to the company's ability to remain in the marketplace with a viable product. In addition, a responsive vendor provides customer support in several forms. A useful venue is a web-based user-accessible knowledge base that includes resolved issues, product documentation, manuals, white papers, and tutorials. Live telephone support can be valuable for customers who don't provide support internally. An issue resolution and escalation process involves customers directly in prioritizing and following closure of critical issues. Onsite presence by the sales team and application engineers, augmented by post-sales support engineers, can significantly shorten the time to discovery and resolution of issues and evolving needs.

## **Appendix A: Acronyms**

**[to come]**

## **Appendix B: Glossary**

[to come]

## **Appendix 4-2a: How to Write a Good Requirement**

### **Use of Correct Terms**

“Shall” = requirement

“Will” = facts or declaration of purpose

“Should” = goal

### **Editorial Checklist**

#### **Personnel Requirement**

- The requirement is in the form “responsible party shall perform such and such.” A requirement must state who shall (do, perform, provide, weigh, or other verb) followed by a description of what.

#### **Product Requirement**

- The requirement is in the form “product ABC shall XYZ.” A requirement must state “The product shall” (do, perform, provide, weigh, or other verb) followed by a description of what, i.e., use active rather than passive voice.
- The requirement uses consistent terminology to refer to the product and its lower-level entities.
- Complete with tolerances for qualitative/performance values (e.g., Less than, greater than or equal to, plus or minus, 3 sigma root sum squares)?
- Free of implementation? (Requirements should state WHAT is needed, NOT HOW to provide it, i.e., state the problem not the solution. Ask, “Why do you need the requirement?” The answer may point to the real requirement.)
- Free of descriptions of operations? (Is this a need the product must satisfy or an activity involving the product?” Sentences like “The operator shall...” are almost always operational statements not requirements.)

#### **Example Product Requirements:**

- The system shall operate at a power level of...
- The software shall acquire data from the...
- The structure shall withstand loads of...
- The hardware shall have a mass of...

#### **General Goodness Checklist**

- The requirement is grammatically correct.
- The requirement is free of typos, misspellings, and punctuation errors.

- The requirement complies with the project’s template and style rules.
- The requirement is stated positively (as opposed to negatively, i.e., “shall not”).
- The use of “To Be Determined” (TBD) values should be minimized. It is better to use a best estimate for a value and mark it “To Be Resolved” (TBR) with the rationale along with what must be done to eliminate the TBR, who is responsible for its elimination, and by when it must be eliminated.
- The requirement is accompanied by an intelligible rationale, including any assumptions. Can you validate (concur with) the assumptions? Assumptions must be confirmed before baselining.
- The requirement is located in the proper section of the document (e.g., not in an appendix)?

## **Requirements Validation Checklist**

### *CLARITY*

1. Are the requirements clear and unambiguous? (Are all aspects of the requirement understandable and not subject to misinterpretation? Is the requirement free from indefinite pronouns (this, these) and ambiguous terms (e.g., “as appropriate,” “etc.,” “and/or,” “but not limited to”)?)
2. Are the requirements concise and simple?
3. Do the requirements express only one thought per requirement statement, a standalone statement as opposed to multiple requirements in a single statement or a paragraph that contains both requirements and rationale?

### *COMPLETENESS*

1. Are requirements stated as completely as possible? Have all incomplete requirements been captured as TBDs or TBRs and a complete listing of them maintained with the requirements?
2. Are any requirements missing? For example have any of the following requirements areas been overlooked: functional, performance, interface, environment (development, manufacturing, test, transport, storage, operations), facility (manufacturing, test, storage, operations), transportation (among areas for manufacturing, assembling, delivery points, within storage facilities, loading), training, personnel, operability, safety, security, appearance and physical characteristics, and design.
3. Have all assumptions been explicitly stated?

### *COMPLIANCE*

1. Are all requirements at the correct level (e.g., system, segment, element, subsystem)?

2. Are requirements free of implementation specifics? (Requirements should state what is needed, not how to provide it.)
3. Are requirements free of descriptions of operations? (Don't mix operation with requirements: update the Concept of Operations instead.)

#### *CONSISTENCY*

1. Are the requirements stated consistently without contradicting themselves or the requirements of related systems?
2. Is the terminology consistent with the user and sponsor's terminology? With the project glossary?
3. Is the terminology consistently used through out the document?
4. Are the key terms included in the project's glossary?

#### *TRACEABILITY*

1. Are all requirements needed? Is each requirement necessary to meet the parent requirement? Is each requirement a needed function or characteristic? Distinguish between needs and wants. If it is not necessary, it is not a requirement. Ask, "What is the worst that could happen if the requirement was not included?"
2. Are all requirements (functions, structures, and constraints) bidirectionally traceable to higher level requirements or mission or system-of-interest scope (i.e., need(s), goals, objectives, constraints, or concept of operations)?
3. Is each requirement stated in such a manner that it can be uniquely referenced (e.g., each requirement is uniquely numbered) in subordinate documents?

#### *CORRECTNESS*

1. Is each requirement correct?
2. Is each stated assumption correct? Assumptions must be confirmed before the document can be baselined.
3. Are the requirements technically feasible?

#### *FUNCTIONALITY*

1. Are all described functions necessary and together sufficient to meet mission and system goals and objectives?

#### *PERFORMANCE*

1. Are all required performance specifications and margins listed? (e.g., consider timing, throughput, storage size, latency, accuracy and precision).

2. Is each performance requirement realistic?
3. Are the tolerances overly tight? Are the tolerances defensible and cost-effective? Ask, "What is the worst thing that could happen if the tolerance was doubled or tripled?"

#### *INTERFACES*

1. Are all external interfaces clearly defined?
2. Are all internal interfaces clearly defined?
3. Are all interfaces necessary, sufficient, and consistent with each other?

#### *MAINTAINABILITY*

1. Have the requirements for system maintainability been specified in a measurable, verifiable manner?
2. Are requirements written so that ripple effects from changes are minimized (i.e., requirements are as weakly coupled as possible)?

#### *RELIABILITY*

1. Are clearly defined, measurable, and verifiable reliability requirements specified?
2. Are there error detection, reporting, handling, and recovery requirements?
3. Are undesired events (e.g., single event upset, data loss or scrambling, operator error) considered and their required responses specified?
4. Have assumptions about the intended sequence of functions been stated? Are these sequences required?
5. Do these requirements adequately address the survivability after a software or hardware fault of the system from the point of view of hardware, software, operations personnel and procedures?

#### *VERIFIABILITY/TESTABILITY*

1. Can the system be tested, demonstrated, inspected, or analyzed to show that it satisfies requirements? Can this be done at the level of the system at which the requirements is stated? Does a means exist to measure the accomplishment of the requirement and compliance verified? Can the criteria for verification be stated?
2. Are the requirements stated precisely to facilitate specification of system test success criteria and requirements?
3. Are the requirements free of unverifiable terms (e.g., flexible, easy, sufficient, safe, ad hoc, adequate, accommodate, user-friendly, useable, when required, if required, appropriate, fast, portable, light-weight, small, large, maximize, minimize, sufficient, robust, quickly, easily, clearly, other "ly" words, other "ize" words)?

### *DATA USAGE*

1. Where applicable, are “don’t care” conditions truly “don’t care”? (“Don’t care” values identify cases when the value of a condition or flag is irrelevant, even though the value may be important for other cases.) Are “don’t care” conditions values explicitly stated? (Correct identification of “don’t care” values may improve a design’s portability.)

## Appendix 4-2b: Requirements Verification Matrix

*When developing requirements, it is important to identify an approach for verifying the requirements. This appendix provides the matrix that defines how all the requirements are verified. Only “shall” requirements should be included in these matrices. The matrix should identify each “shall” by unique identifier and be definitive as to the source, i.e., document from which the requirement is taken. This matrix could be divided into multiple matrices (e.g., one per requirements document) to delineate sources of requirements depending on the project. The example is shown to provide suggested guidelines for the minimum information that should be included in the verification matrix.*

Reqm't No. <sup>1</sup>	Docu-ment <sup>2</sup>	Para-graph <sup>3</sup>	Shall Statement <sup>4</sup>	Verification Success Criteria <sup>5</sup>	Verif. Method <sup>6</sup>	Facil-ity or Lab <sup>7</sup>	Phas-e <sup>8</sup>	Accept-ance Requirement? <sup>9</sup>	Pre-Flight Accept-ance? <sup>10</sup>	Per-form. Org <sup>11</sup>	Results <sup>12</sup>
P-1	xxx	3.2.1.1 Capability: Support Uplinked Data (LDR)	The System X shall provide a max. ground-to-station uplink of...	1. System X locks to forward link at the min and max data rate tolerances. 2. System X locks to the forward link at the min and max operating frequency tolerances	Test	xxx	5			xxx	TPS xxxx
P-i	xxx	Other Para-graphs	Other “shalls” in PTRS	Other criteria	xxx	xxx	xxx			xxx	Memo xxx
S-i or other unique designator	xxxxx (Other Specs, ICDs, etc.)	Other Para-graphs	Other “shalls” in Specs, ICDs, etc	Other criteria	xxx	xxx	xxx			xxx	Report xxx

<sup>1</sup> Unique identifier for each System X requirement

<sup>2</sup> Document number the System X requirement is contained within

<sup>3</sup> Paragraph number of the System X requirement

<sup>4</sup> Text (within reason) of the System X requirement, i.e., the “shall”

<sup>5</sup> Success Criteria for the System X requirement

<sup>6</sup> Verification Method for the System X requirement (Analysis, Inspection, Demonstration, or Test)

<sup>7</sup> Facility or laboratory used to perform the verification/validation

<sup>8</sup> Phase in which the verification/validation will be performed:

1. Pre-declared Development
2. Formal Box-level Functional
3. Formal Box-level Environmental
4. Formal System-level Environmental
5. Formal System-level Functional
6. Formal end-to-end Functional
7. Integrated Vehicle Functional
8. On-orbit Functional

<sup>9</sup> Indicate whether this requirement is also verified during initial acceptance testing of each unit

<sup>10</sup> Indicate whether this requirement is also verified during any pre-flight or recurring acceptance testing of each unit

<sup>11</sup> Organization responsible for performing the verification

<sup>12</sup> Indicate documents that contain the objective evidence that requirement was satisfied

ID	DESCRIPTION	REQUIREMENT	TRACED FROM	PERFORMANCE	MARGIN	Comments	REF
M1	Mission Orbit	575 +/-15 km Sun-synchronous dawn-dusk orbit	S3, S11, P3	Complies	NA	Pegasus XL with HAPS provides required launch injection dispersion accuracy	F.2.c
M2	Launch Vehicle	Pegasus XL with HAPS	P2, P4	Complies	NA		F.2.c
M3	Observatory Mass	The NEXUS Observatory total mass shall not exceed 241 kg.	M1, M2	192.5 kg	25.20%		F.5.b
M4	Data Acquisition Quality	The NEXUS mission shall deliver 95% data with better than 1 in 100,000 BER.	P1	Complies	NA	Standard margins and systems baselined, formal system analysis to be completed by PDR	F.7
M5	Communication Band	The mission shall use S-band SQPSK at 5 Mbps for spacecraft downlink and 2 kbps uplink.	S12, P4	Complies	NA	See SC27, SC28 and G1, G2	F.3.f, F.7
M7	Tracking	MOC shall use NORAD two line elements for observatory tracking	P4	Complies	NA		F.7
M8	Data Latency	Data Latency shall be less than 72 hours	P12	Complies	NA		F.7
M9	Daily Data Volume	Accommodate average daily raw science data volume of 10.8 Gbits	P1, S12	Complies	12%	Margin based on funded ground contacts	F.3.e, F.7
M10	Ground Station	The Mission Shall be Compatible With the Rutherford Appleton Laboratory Ground Station and the Poker Flat Ground Station	P1	Complies	NA		F.7
M11	Orbital Debris (Casualty Area)	Design NEXUS observatory for demise upon reentry with < 1/10,000 probability of injury	P3	1/51,000	400%	See Orbital Debris Analysis in Appendix M-6	F.2.e, App.6
M12	Orbital Debris (Lifetime)	Design NEXUS observatory for re-entry <25 years after end of mission	P3	< 10 years	15 years	See Orbital Debris Analysis in Appendix M-6	F.2.e, App.6

Table 13

## Appendix 4-2c: Creating the Validation Plan (Including Validation Requirements Matrix)

*When developing requirements, it is important to identify a validation approach for how additional validation evaluation, testing, analysis or other demonstrations will be performed to ensure customer/sponsor satisfaction. This validation plan should include a validation matrix with the elements in the example below. The final column in the matrix below uses a display product as a specific example.*

Validation Product #	Activity	Objective	Valid. Method	Facility or Lab	Phase	Perform. Org	Results
Unique identifier for validation product	Describe evaluation by the customer/sponsor that will be performed	What is to be accomplished by the customer/sponsor evaluation	Validation Method for the System X requirement (Analysis, Inspection, Demonstration, or Test)	Facility or laboratory used to perform the validation	Phase in which the verification/validation will be performed <sup>1</sup>	Organization responsible for coordinating the validation activity	Indicate the objective evidence that validation activity occurred
1	The customer/sponsor will evaluate the candidate displays	1. Ensure legibility is acceptable 2. Ensure overall appearance is acceptable	<i>test</i>	xxx	Phase A	xxx	

<sup>1</sup>Example:

1. During product selection process
2. Prior to final product selection (if COTS) or prior to PDR
3. Prior to CDR
4. During box-level Functional
5. During System-level Functional
6. During end-to-end Functional
7. During Integrated Vehicle Functional
8. During On-orbit Functional

## **Appendix 4-3: Functional, Timing, and State Analysis**

### **Functional Flow Block Diagrams**

Functional analysis can be performed using various methods, one of which is functional flow block diagrams (FFBDs). FFBDs define the system functions and depict the time sequence of functional events. They identify “what” must happen and do not assume a particular answer to “how” a function will be performed. They are functionally oriented, not solution oriented.

FFBDs are made up of functional blocks each of which represents a definite, finite, discrete action to be accomplished. The functional architecture is developed using a series of leveled diagrams to show the functional decomposition and display the functions in their logical, sequential relationship. A consistent numbering scheme is used to label the blocks. The numbers establish identification and relationships that carry through all the diagrams and facilitate traceability from the lower levels to the top level. Each block in the first/top level diagram can be expanded to a series of functions in the second level diagram, and so on (see Figure 1). Lines connecting functions indicate function flow and not lapsed time or intermediate activity. Diagrams are laid out so that the flow direction is generally from left to right. Arrows are often used to indicate functional flows. The diagrams show both input (Transfer To OPS Orbit) and output (Transfer To STS Orbit), thus facilitating the definition of interfaces and control process.

Each diagram contains a reference to other functional diagrams to facilitate movement between pages of the diagrams. Gates are used: “AND”, “OR”, “Go/noGo”, sometimes with enhanced functionality to include XOR, Iteration (IT), Repetition (RP), Loop (LP). A circle is used to denote a summing gate and is used when AND/OR is present. “AND” is used to indicate parallel functions and all conditions must be satisfied to proceed (i.e., concurrency). “OR” is used to indicate that alternative paths can be satisfied to proceed (i.e., selection). “G” and “bar G” are used to denote “go” and “no go” conditions. These symbols are placed adjacent to lines leaving a particular function to indicate alternative paths. For examples of the above see Figure 2 and 3.

Enhanced Functional Flow Block Diagrams (EFFBDs) provide data flow overlay to capture data dependencies. EFFBDs (as shown in Figure 4) represent: 1) functions, 2) control flows, and 3) data flows. An EFFBD specification of a system is complete enough that it is executable as a discrete event model, capable of dynamic, as well as static, validation. EFFBDs provide freedom to use either control constructs or data triggers or both to specify execution conditions for the system functions. EFFBDs graphically distinguish between triggering and non-triggering data inputs. Triggering data is required before a function can begin execution. Triggers are actually data items with control implications. In Figure 4, the data input shown with a green background and with the double-headed arrows is a triggering data input. The non-triggering data inputs are shown with gray backgrounds and with single-headed arrows. An EFFBD must be enabled by: (A) the completion of the function(s) preceding it in the control construct and (B) triggered, if trigger data is identified, before it can execute. For example: ‘1. Serial Function’ must complete and ‘Data 3’ must be present before ‘3. Function in Concurrency’ can execute. It should be noted that the ‘External Input’ data into ‘1. Serial Function’ and the ‘External Output’ data from ‘6. Output Function’ should not be confused with the functional input and output for these functions,

which are represented by the input and the output arrows respectively. Data flows are represented as elongated ovals whereas functions are represented as rectangular boxes.

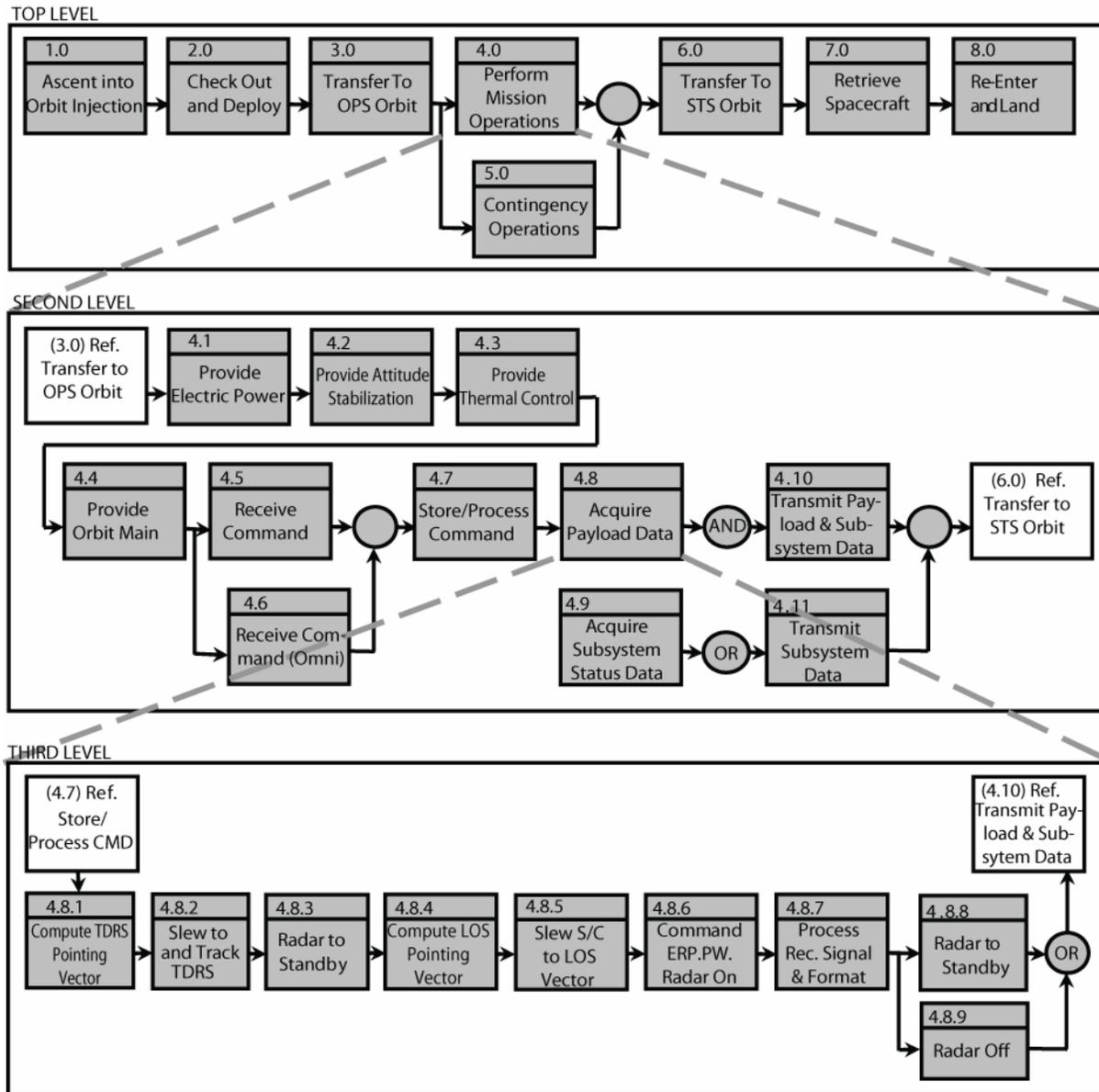
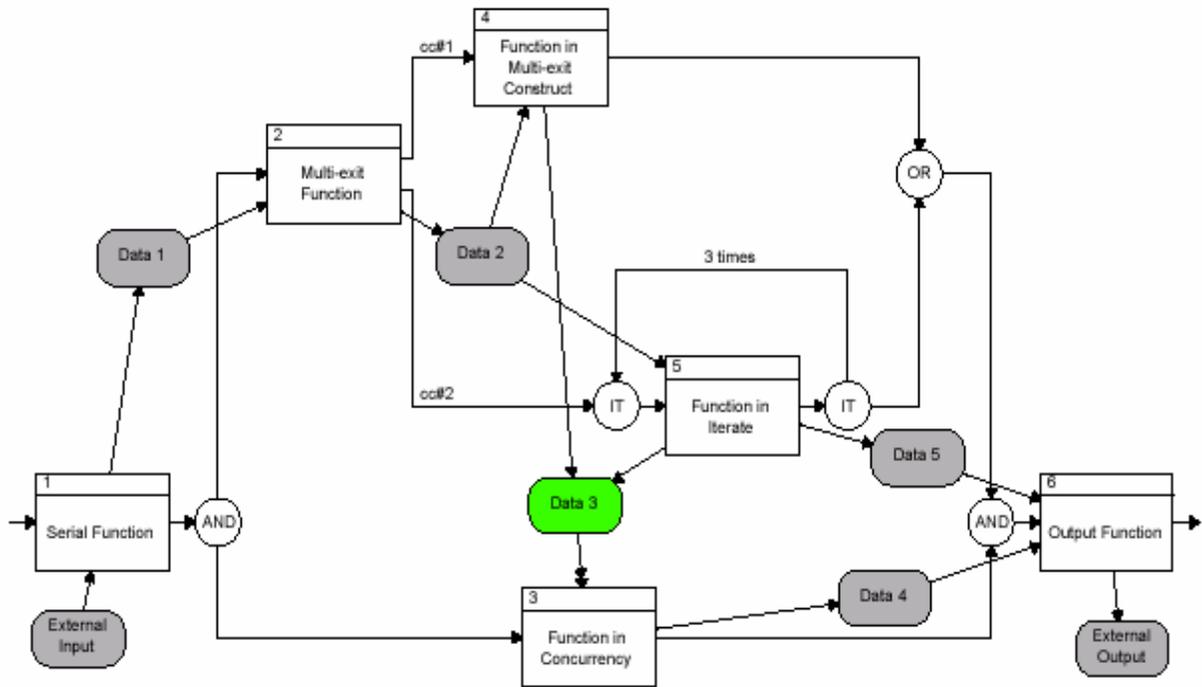


Figure 1: FFBD Flow Down





**Figure 4: Enhanced Functional Flow Block Diagram—Example 3**

Functional analysis looks across all life-cycle processes. Functions required to deploy a system are very different from functions required to operate and ultimately dispose of the system. Preparing FFBDs for each phase of the life cycle as well as the transition into the phases themselves is necessary to draw out all the requirements. These diagrams are used both to develop requirements and to identify profitable. The functional analysis also incorporates alternate and contingency operations, which improve the probability of mission success. The flow diagrams provide an understanding of total operation of the system, serve as a basis for development of operational and contingency procedures, and pinpoint areas where changes in operational procedures could simplify the overall system operation. This organization will eventually feed into the WBS structure and ultimately drive the overall mission organization and cost. In certain cases, alternate FFBDs may be used to represent various means of satisfying a particular function until data are acquired, which permits selection among the alternatives. For more information on FFBDs and EFFBDs see David Long's *Relationships between Common Graphical Representations in Systems Engineering*.

### Requirements Allocation Sheets

Requirements Allocation Sheets document the connection between allocated functions, allocated performance and the physical system. They provide traceability between Technical Requirements Definition functional analysis activities and Logical Decomposition and Design Solution Definition activities and maintain consistency between them, as well as show disconnects. Fill in the Requirements Allocation Sheet by performing the following:

1. Include the functions and function numbers from the FFBDs,

2. Allocate functional performance requirements and design requirements to the appropriate function(s) (many requirements may be allocated to one function or one requirement may be allocated to many functions),
3. All system level requirements must be allocated to a function to ensure the system meets all system requirements (functions without allocated requirements should be eliminated as unnecessary activities),
4. Allocate all derived requirements to the function that spawned the requirement,
5. Identify the physical equipment, configuration item (CI), facilities, and specifications that will be used to meet the requirements,

Figure 5 provides an example of a Requirements Allocation Sheet. The reference column to the far right indicates the function numbers from the FFBDs.

ID	DESCRIPTION	REQUIREMENT	TRACED FROM	PERFORMANCE	MARGIN	Comments	REF
M1	Mission Orbit	575 +/-15 km Sun-synchronous dawn-dusk orbit	S3, S11, P3	Complies	NA	Pegasus XL with HAPS provides required launch injection dispersion accuracy	F.2.c
M2	Launch Vehicle	Pegasus XL with HAPS	P2, P4	Complies	NA		F.2.c
M3	Observatory Mass	The NEXUS Observatory total mass shall not exceed 241 kg.	M1, M2	192.5 kg	25.20%		F.5.b
M4	Data Acquisition Quality	The NEXUS mission shall deliver 95% data with better than 1 in 100,000 BER.	P1	Complies	NA	Standard margins and systems baselined, formal system analysis to be completed by PDR	F.7
M5	Communication Band	The mission shall use S-band SQPSK at 5 Mbps for spacecraft downlink and 2 kbps uplink.	S12, P4	Complies	NA	See SC27, SC28 and G1, G2	F.3.f, F.7
M7	Tracking	MOC shall use NORAD two line elements for observatory tracking	P4	Complies	NA		F.7
M8	Data Latency	Data Latency shall be less than 72 hours	P12	Complies	NA		F.7
M9	Daily Data Volume	Accommodate average daily raw science data volume of 10.8 Gbits	P1, S12	Complies	12%	Margin based on funded ground contacts	F.3.e, F.7
M10	Ground Station	The Mission Shall be Compatible With the Rutherford Appleton Laboratory Ground Station and the Poker Flat Ground Station	P1	Complies	NA		F.7
M11	Orbital Debris (Casualty Area)	Design NEXUS observatory for demise upon reentry with < 1/10,000 probability of injury	P3	1/51,000	400%	See Orbital Debris Analysis in Appendix M-6	F.2.e, App.6
M12	Orbital Debris (Lifetime)	Design NEXUS observatory for re-entry <25 years after end of mission	P3	< 10 years	15 years	See Orbital Debris Analysis in Appendix M-6	F.2.e, App.6

Table 13

**Figure 5: Requirements Allocation Sheet**

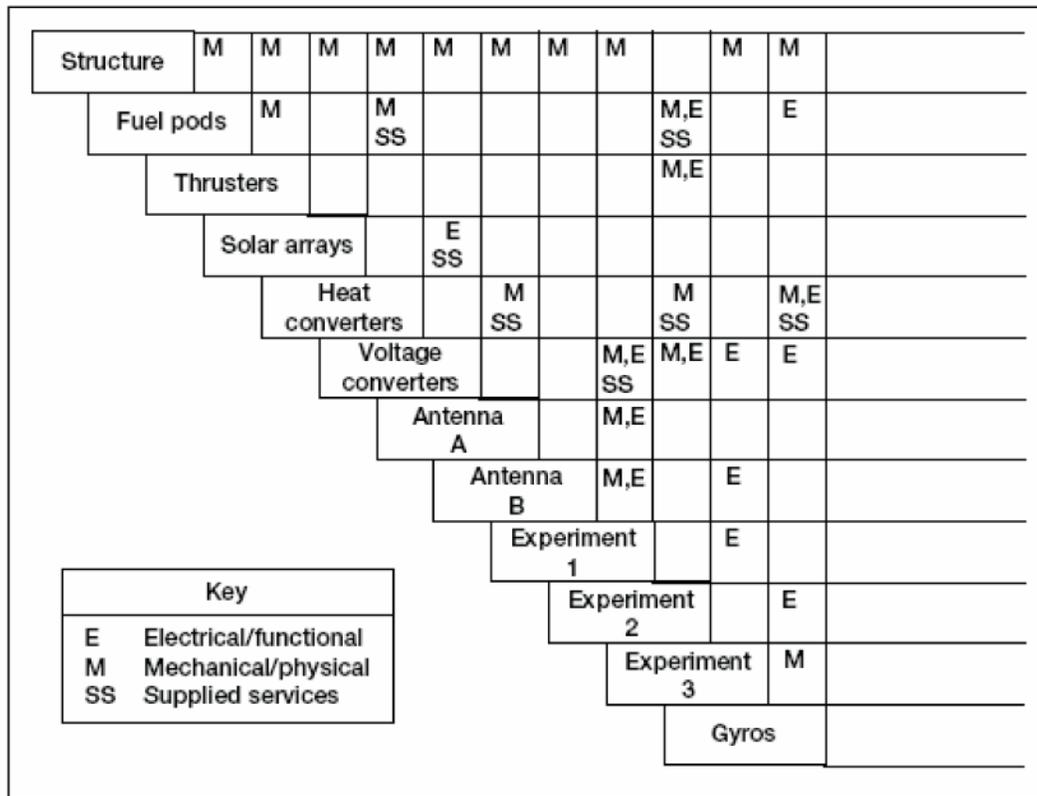
[For a reference on Requirements Allocation Sheets see *Systems Engineering Fundamentals Guide*, Defense Acquisition University, Fort Belvoir, VA, January 2001]

## N2 Diagrams

An N-Squared (N2) diagram is a matrix representation of functional and/or physical interfaces between elements of a system at a particular hierarchical level. The N2 diagram has been used extensively to develop data interfaces, primarily in the software areas. However, it can also be used to develop hardware interfaces as shown in Figure 6 below. The system components are placed on the diagonal. The remainder of the squares in the N x N matrix represent the interfaces. The square at the intersection of a row and a column contains a description of the interface between the two components represented on that row and that column. For example, the

solar arrays have a mechanical (M) interface with the structure and an electrical (E) interface and supplied service (SS) interface with the voltage converters. Where a blank appears there is no interface between the respective components.

The N2 chart can be taken down into successively lower levels to the hardware and software component functional levels. In addition to defining the data that must be supplied across the interface, by showing the data flows the N2 chart pinpoints areas where conflicts could arise in interfaces, and highlights input and output dependency assumptions and requirements.



**Figure 6: N-Squared Diagram for Orbital Equipment**

(Entries not polarized.) [From NASA Reference Publication 1370, Training Manual for Elements of Interface Definition and Control, January 1997]

### Timing Analysis

There are several methods for visualizing the complex timing relationships in a system. Two of the more important ones are the timing diagram and the state transition diagram. The timing diagram (see Figure 6) defines the behavior of different objects within a timescale. It provides a visual representation of objects changing state and interacting over time. Timing diagrams can be used for defining the behavior of hardware-driven and/or software-driven components. While a simple timeline analysis is very useful in understanding relationships such as concurrency, overlap, and sequencing, state diagrams (see section below) allow for even greater flexibility in that they can depict events such as loops and decision processes that may have largely varying



Self-loops are permitted. In the example in Figure 8 the slew states can loop until they arrive at the correct location and then they can loop while they settle.

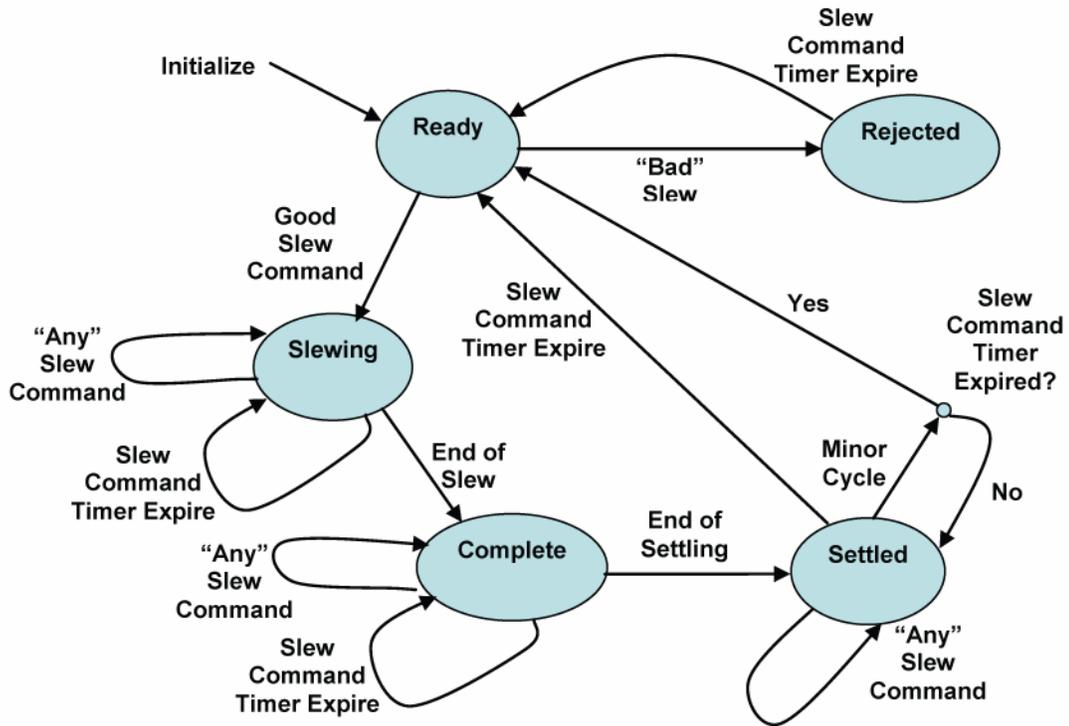


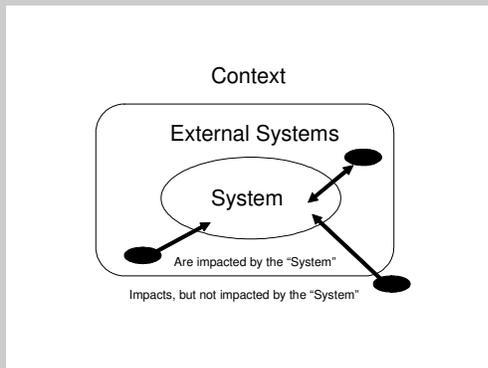
Figure 8: Slew Command Status State Diagram

When it is used to represent the behavior of a sequential finite state machine the state diagram is called a state transition diagram (STD). A sequential finite state machine is one that has no memory which means that the current output only depends on the current input. The STD models the event-based, time-dependent behavior of such a system.

[Some references on state transition diagrams are Dennis Buede's book, The Engineering Design of Systems, and the Handbook of Systems Engineering and Management by Andrew Sage and William Rouse.]

## Context Diagrams

When presented with a system design problem, the systems engineer's first task is to truly understand the problem. That means understanding the context in which the problem is set. A *context diagram* is a useful tool for grasping the system to be built and the *external domains* that are relevant to that system and which have interfaces to the system. The following diagram from Dennis Buede's book, *The Engineering Design of Systems*, page 38, shows the general structure of a context diagram. The system is shown surrounded by the external systems which have interfaces to the system. These systems are not part of the system but they interact with the system via the system's external interfaces. The external systems can impact the system and the system does impact the external systems. They play a major role in establishing the requirements for the system. Entities further removed are those in the system's context that can impact the system but cannot be impacted by the system. These entities in the system's context are responsible for some of the system's requirements.



Defining the boundaries of a system is a critical but often neglected task. Using an example from a satellite project, one of the external systems which are impacted by the satellite would be the Tracking and Data Relay Satellite System (TDRSS). TDRSS is not part of the satellite system but it defines requirements on the satellite and it is impacted by the satellite since it must schedule contacts, receive and transmit data and commands, and downlink the satellite data to the ground. An example of an entity in the context of the satellite system that is not impacted by the satellite system is the Global Positioning Satellite (GPS) system. GPS is not impacted in any way by the satellite, but GPS will levy some requirements on the satellite if the satellite is to use the GPS signals for navigation.

*Reference: Dennis Buede, The Engineering Design of Systems, Wiley Series in Systems Engineering, 2000.*

## **Appendix 4-4: Technology Assessment/Insertion**

### **Introduction, Purpose and Scope**

The Agency's programs and projects, by their very nature, frequently require the development and infusion of new technological advances in order to meet mission goals, objectives and resulting requirements. Sometimes the "new" technological advancement being infused is actually a "heritage" system that is being incorporated into a different architecture and operated in different environment from that for which it was originally designed. In this latter case, it is often not recognized that adaptation of heritage systems frequently requires technological advancement and as a result, key steps in the development process are given short shrift—often to the detriment of the program/project. In both contexts of technological advancement (new and adapted heritage), infusion is a very complex process that has been dealt with over the years in an ad hoc manner differing greatly from project to project with varying degrees of success. Frequently, technology infusion has resulted in schedule slips, cost overruns and occasionally even to cancellations or failures. In post mortem, the root cause of such events has often been attributed to "inadequate definition of requirements." If such were indeed the "root cause," then correcting the situation would simply be a matter of requiring better requirements definition, but since history seems frequently to repeat itself, this must not be the case—at least not in total. In fact there are many contributors to schedule slip, cost overrun, project cancellation and failure—among them lack of adequate requirements definition. The case can be made that most of these contributors are related to the degree of uncertainty at the outset of the project and that a dominant factor in the degree of uncertainty is the lack of understanding of the maturity of the technology required to bring the project to fruition and a concomitant lack of understanding of the cost and schedule reserves required to advance the technology from its present state to a point where it can be qualified and successfully infused with a high degree of confidence. Although this uncertainty cannot be eliminated, it can be substantially reduced through the early application of good systems engineering practices focused on understanding the technological requirements, the maturity of the required technology and the technological advancement required to meet program/project goals, objectives and requirements.

There are a number of processes that can be used to develop the appropriate level of understanding required for successful technology insertion. The intent of this section is to describe a systematic process that can be used as an example of how to apply standard systems engineering practices to perform a comprehensive Technology Assessment (TA). The TA is comprised of two parts, a Technology Maturity Assessment (TMA) and an Advancement Degree of Difficulty Assessment (AD<sup>2</sup>). The process begins with the TMA which is used to determine technological maturity via NASA's Technology Readiness Level (TRL) scale. It then proceeds to develop an understanding of what is required to advance the level of maturity through a process called the Advancement Degree of Difficulty (AD<sup>2</sup>). It is necessary to conduct a TA at various stages throughout a program/project in order to provide the KDP products required for transition between phases (Figure 1).

**KDP A—Transition from Pre-Phase A to Phase A:** Requires an assessment of potential technology needs versus current and planned technology readiness levels, as well as potential opportunities to use commercial, academic, and other government agency sources of technology. Included as part of the draft integrated baseline

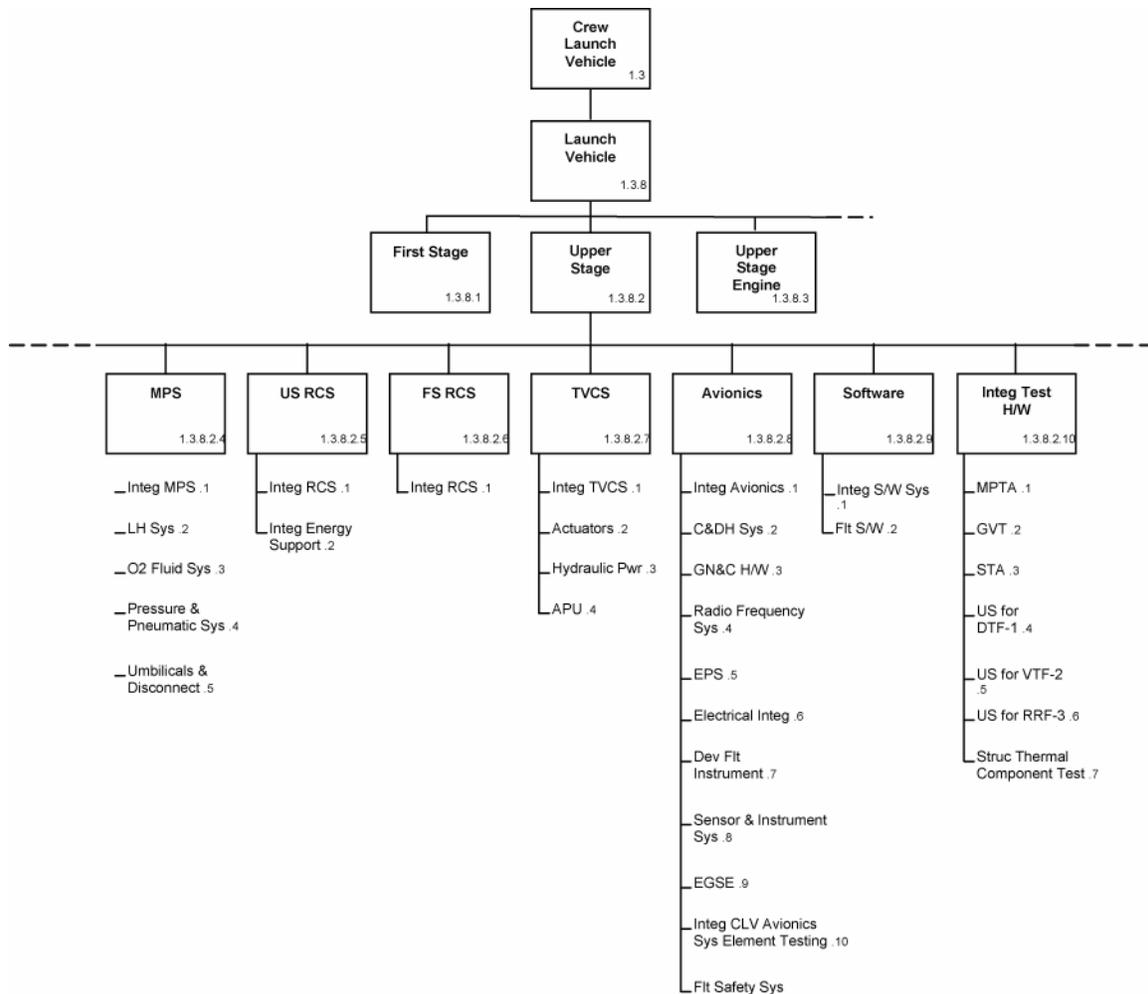
**KDP B—Transition from Phase A to Phase B:** Requires a Technology Development plan identifying technologies to be developed, heritage systems to be modified, alternate paths to be pursued, fall back positions and corresponding performance de-scopes, milestones, metrics and key decision points. Incorporated in the preliminary Project Plan.

**KDP C—Transition from Phase B to Phase C/D:** Technology Readiness Assessment Report (TRAR) demonstrating that all systems, subsystems and components have achieved a level of technological maturity with demonstrated evidence of qualification in a relevant environment.

**Figure1. Products provided by the TA as a function of program/project phase (NPR 7120.5d)**

The initial TMA serves to provide the baseline maturity of the system at program/project outset, and to monitor progress throughout development. The final TMA is performed just prior to the Preliminary Design Review (PDR) and it forms the basis for the Technology Readiness Assessment Report (TRAR) which documents the maturity of the systems, subsystems and components demonstrated through test and analysis. The initial AD<sup>2</sup> assessment provides the material necessary to develop preliminary cost, and schedule plans and preliminary risk assessments. In subsequent assessment the information is then used to build the technology development plan in the process identifying alternative paths, fall-back positions and performance de-scope options. The information is also vital to the preparation of milestones and metrics for subsequent Earned Value Management.

The assessment is performed against the hierarchical breakdown of the hardware and software products of the program/project Product Breakdown Structure (PBS) in order to achieve a systematic, overall understanding at the system, subsystem and component level (Figure 2).



**Figure 2. PBS Example**

### Inputs/Entry Criteria

It is extremely important that a Technology Assessment process be defined at the beginning of the program/project and that it be performed at the earliest possible stage (concept development) throughout the program/project through PDR. Inputs to the process will vary in level of detail according to the phase of the program/project, and even though there is a lack of detail in pre-phase A, the TA will drive out the major critical technological advancements required. Therefore, at the beginning of pre-phase A, the following should be provided:

- Refinement of Technology Readiness Level Definitions
- Definition of Advancement Degree of Difficulty Process
- Definition of terms to be used in the assessment process
- Establishment of meaningful evaluation criteria and metrics that will allow for clear identification of gaps and shortfalls in performance.

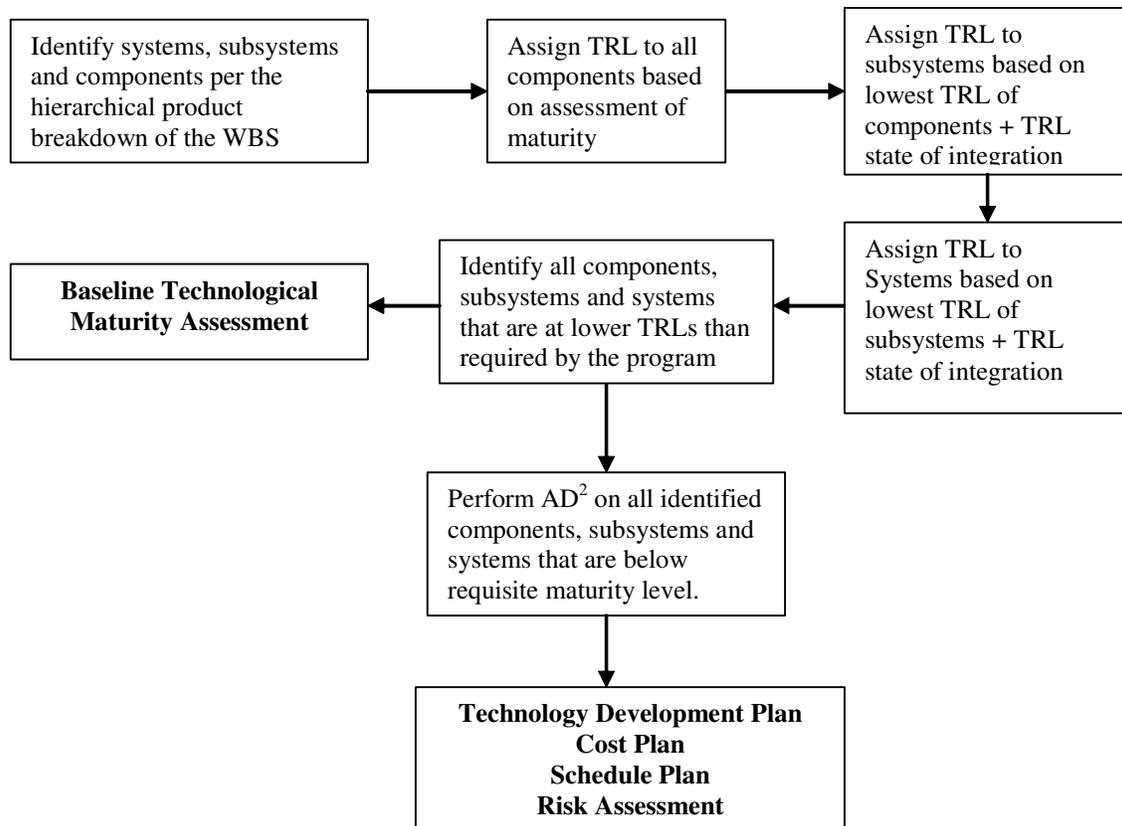
- Establishment of the TA team
- Establishment of an independent TA review team.

### **How to Do Technology Assessment**

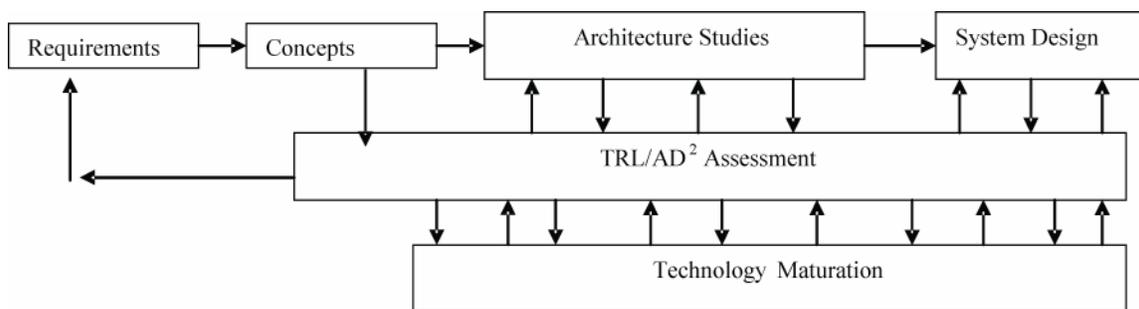
The technology assessment process makes use of basic systems engineering principles and processes. As mentioned previously, it is structured to occur within the framework of the Product Breakdown Structure (PBS) in order to facilitate incorporation of the results. Using the PBS as a framework has a two-fold benefit—it breaks the “problem” down into systems, subsystems and components that can be more accurately assessed, and it provides the results of the assessment in a format that can readily be used in the generation of program costs and schedules. It can also be highly beneficial in providing milestones and metrics for progress tracking using Earned Value Management. As discussed above, it is a two step process comprised of step one, the determination of the current technological maturity in terms of Technology Readiness Levels (TRLs) and step 2, the determination of the difficulty associated with a moving a technology from one TRL to the next through the use of the Advancement Degree of Difficulty (AD<sup>2</sup>). The overall process is iterative, starting at the conceptual level during program formulation, establishing the initial identification of critical technologies and the preliminary cost, schedule and risk mitigation plans. Continuing on into Phase A, it is used to establish the baseline maturity, the Technology Development plan and associated costs and schedule. The final TA consists only of the TMA and is used to develop the TRAR which validates that all elements are at the requisite maturity level (Figure 3.)

Even at the conceptual level, it is important to use the formalism of a PBS to avoid having important technologies slip through the crack. Because of the preliminary nature of the concept, the systems, subsystems and components will be defined at a level that will not permit detailed assessments to be made. The process of performing the assessment, however, is the same as that used for subsequent, more detailed steps that occur later in the program/project where systems are defined in greater detail.

Once the concept has been formulated and the initial identification of critical technologies made, it is necessary to perform detailed architecture studies with the Technology Assessment Process intimately interwoven (Figure 4). The purpose of the architecture studies is to refine end-item system design to meet the overall scientific requirements of the mission. It is imperative that there be a continuous relationship between architectural studies and maturing technology advances. The architectural studies must incorporate the results of the technology maturation, planning for alternate paths and identifying new areas required for development as the architecture is refined. Similarly, it is incumbent upon the technology maturation process to identify requirements that are not feasible and development routes that are not fruitful and to transmit that information to the architecture studies in a timely manner. Similarly, it is incumbent upon the architecture studies to provide feedback to the technology development process relative to changes in requirements. Particular attention must be given to “heritage” systems in that they are often used in architectures and environments different from those in which they were designed to operate.



**Figure 3. Technology Assessment Process**



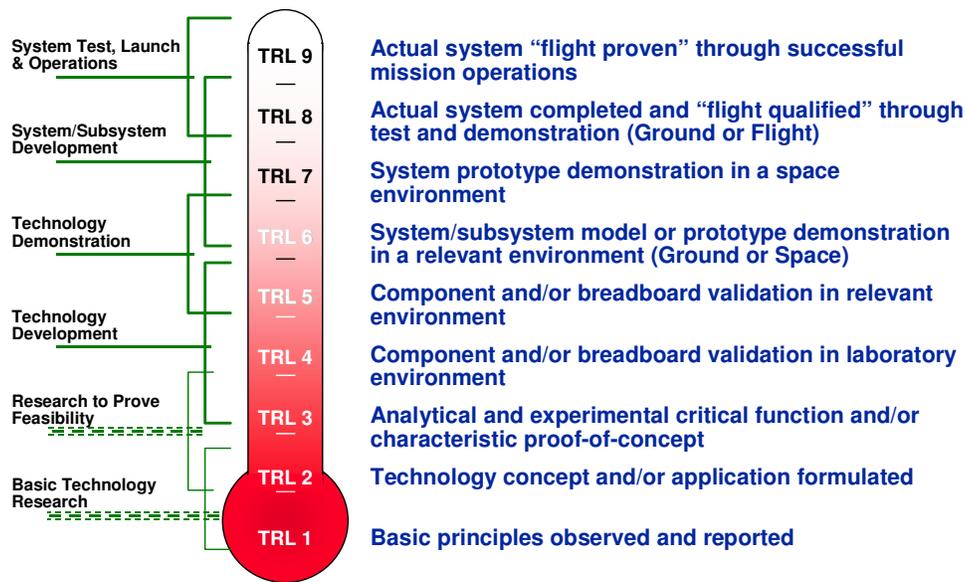
**Figure 4. Architectural studies and technology development**

### Establishing Technology Readiness Levels

TRL is, at its most basic, a description of the “performance history” of a given system, subsystem or component relative to a set of levels first described at NASA Headquarters in the 1980’s. The TRL essentially describes the state of the art of a given technology and provides a “baseline” from which maturity is gauged and advancement defined. Even though the concept of TRL has been around for almost 20 years, it is not well understood and frequently misinterpreted. As a result we often undertake programs without fully understanding either the

maturity of key technologies or what is needed to develop them to the required level. *It is impossible to understand the magnitude and scope of a development program without having a clear understanding of the baseline technological maturity of all elements of the system.* Establishing the TRL is a vital first step on the way to a successful program. A frequent misconception is that in practice it is too difficult to determine TRLs and that when you do it is not meaningful. On the contrary, identifying TRLs can be a straightforward systems engineering process of determining what was demonstrated and under what conditions was it demonstrated.

At first blush, the TRL descriptions in Figure 5 appear to be straightforward. It is in the process of trying to assign levels that problems arise. A primary cause of difficulty is in terminology—everyone knows what a breadboard is, but not everyone has the same definition. Also, what is a “relevant environment”? What is relevant to one application may or may not be relevant to another. Many of these terms originated in various branches of engineering and had, at the time, very specific meanings to that particular field. They have since become commonly used throughout the engineering field and often take differences in meaning from discipline to discipline, some subtle, some not so subtle. Breadboard for example comes from electrical engineering where the original use referred to checking out the functional design of an electrical circuit by populating a “breadboard” with components to verify that the design operated as anticipated. Other terms come from mechanical engineering, referring primarily to units which are subjected to different levels of stress under testing, i.e., qualification, proto-flight and flight units. The first step in developing a uniform TRL assessment is then to define the terms used. It is extremely important to develop and use a consistent set of definitions over the course of the program/project.

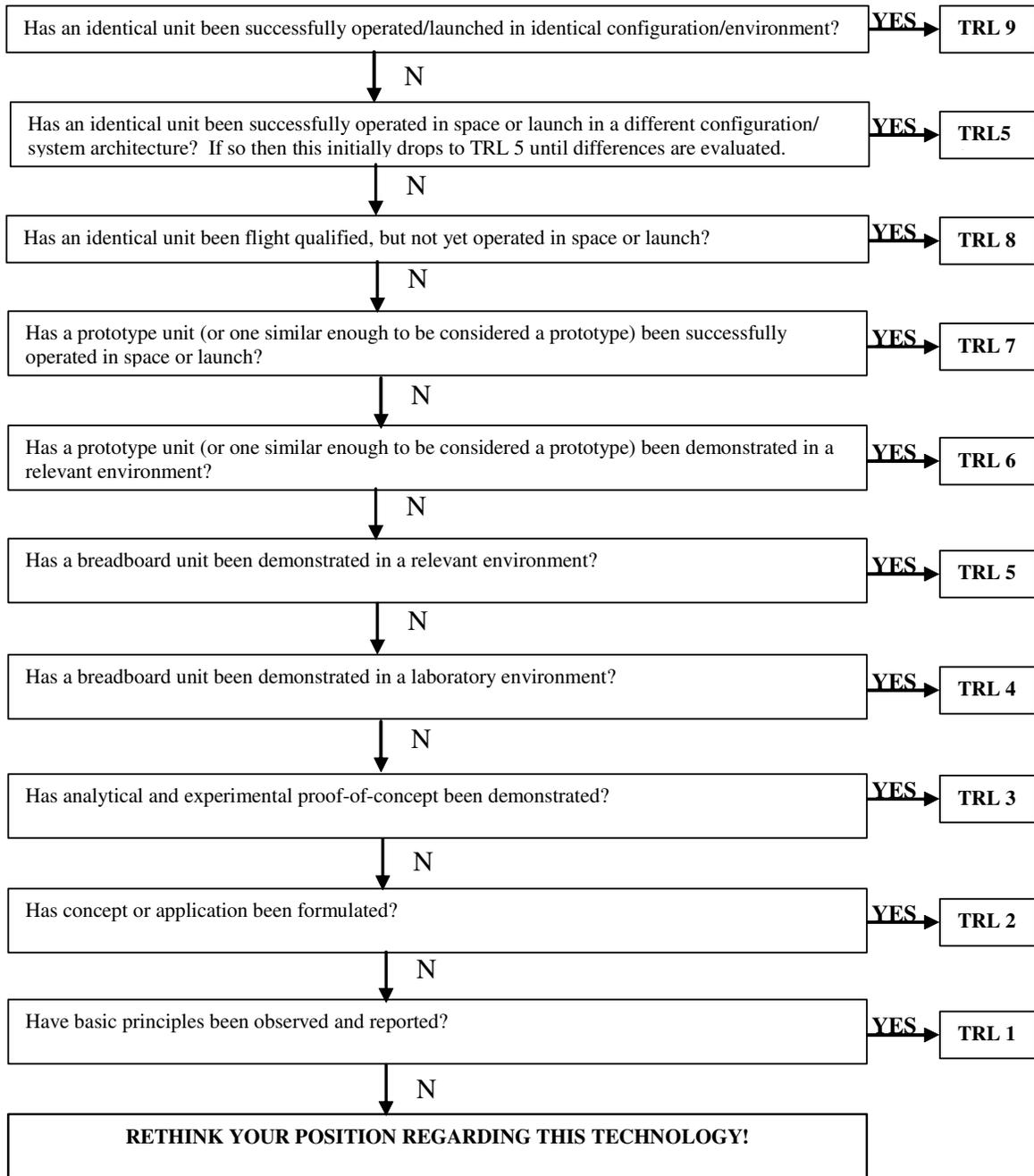


**Figure 5. Technology Readiness Levels**

Having established a common set of terminology, it is necessary to proceed to the next step—quantifying “judgment calls” on the basis of past experience. Even with clear definitions there will be the need for “judgment calls” when it comes time to assess just how similar a given element is relative to what is needed (i.e., is it “close enough” to a prototype to be considered a prototype, or is it more like an engineering breadboard?) Describing what has been done in terms of form, fit and function provides a means of quantifying an element based on its design intent and subsequent performance.

One of the problems associated with the use of TRL’s as currently defined is that they are very difficult to apply to software development. Consequently, the Agency is in the process of redefining the TRL’s to take that into account. The proposed revised description is contained in Appendix B.

A third critical element of any assessment relates to the question of who is in the best position to make “judgment calls” relative to the status of the technology in question. For this step, it is extremely important to have a well-balanced, experienced assessment team. Team members do not necessarily have to be “discipline experts.” The primary expertise required for a TRL assessment is that the systems engineer/user understands the current state of the art in applications. Having established a set of definitions, defined a process for quantifying “judgment calls,” and assembled an expert assessment team, the process primarily consists of asking the right questions. The flow chart depicted in Figure 6 demonstrates the questions to ask in order to determine TRL at any level in the assessment.



**Figure 6. The TMA Assessment thought process**

Note the 2<sup>nd</sup> box particularly refers to heritage systems. If the architecture and the environment has changed then the TRL drops to TRL 5—at least initially. If in subsequent analysis the new environment is sufficiently close to the old environment, or the new architecture sufficiently close to the old architecture then the resulting evaluation could be then TRL 6 or 7, but the most important thing to realize is that it is no longer at a TRL 9. Applying this process at the system level and then proceeding to lower levels of subsystem and component identifies those elements that require development and sets the stage for the subsequent phase, determining the AD<sup>2</sup>.

A method for formalizing this process is shown in Figure 7. In this figure, the process has been set up as a table, on the left hand side of the table, the rows identify the systems, subsystems, and components that are under assessment. The columns identify the categories that will be used to determine the TRL, i.e., what units have been built, to what scale, and in what environment have they been tested? Answers to these questions determine the TRL of an item under consideration. The TRL of the system is determined by the lowest TRL present in the system, i.e., a system is at TRL 2 if any single element in the system is at TRL 2. The problem of multiple elements being at low TRLs is dealt with in the AD<sup>2</sup> process. It should be noted that the issue of integration affects the TRL of every system, subsystem and component. All of the elements can be at a higher TRL, but if they have never been integrated as a unit, the TRL will be lower for the unit. How much lower depends on the complexity of the integration.

TRL Assessment														
	Concept	Demonstration Units				Environment			Unit Description				Overall TRL	
		Breadboard	Brassboard	Developmental Model	Prototype	Flight Qualified	Laboratory Environment	Relevant Environment	Space Environment	Space/Launch Operation	Form	Fit		Function
1.0 System														
1.1 Subsystem X														Red
1.1.1 Mechanical Components														
1.1.2 Mechanical Systems														
1.1.3 Electrical Components				X				X		X	X	X		Green
1.1.4 Electrical Systems														
1.1.5 Control Systems														
1.1.6 Thermal Systems						X				X	X			Yellow
1.1.7 Fluid Systems		X												Red
1.1.8 Optical Systems														
1.1.9 Electro-optical Systems														
1.1.10 Software Systems														
1.1.11 Mechanisms	X													Red
1.1.12 Integration														
1.2 Subsystem Y														Yellow
1.2.1 Mechanical Components														

Figure 7. TRL Assessment Matrix

## Appendix 5-2: Integration Plan Outline

[DRAFT]

Integration Plan Template

[adapted from [http://www.fhwa.dot.gov/cadiv/segb/views/document/Sections/Section8/8\\_4\\_8.htm](http://www.fhwa.dot.gov/cadiv/segb/views/document/Sections/Section8/8_4_8.htm)  
U.S. Department of Transportation Federal Highway Administration website]

**Purpose:** The defines the integration and verification strategies for a project interface with the system design and decomposition into the lower level elements. The Integration Plan is structured to bring the elements together to assemble each subsystem and to bring all of the subsystems together to assemble the system/product. The primary purposes of the Integration Plan are (1) to describe this coordinated integration effort that supports the implementation strategy, (2) to describe what needs to be done in each integration step for the participants, and (3) to identify the required resources that will be needed by when they are needed and where they will be needed.

### Questions/Checklist:

- Does the Integration Plan include and cover integration of all of the components and sub-systems, either developed or purchased, of the project?
- Does the Integration Plan account for all external systems to be integrated with the system [for example, communications networks, field equipment, other complete systems owned by the government or owned by other government agencies]?
- Does the Integration Plan fully support the implementation strategy. For Example, *when* and *where* the sub-systems and system are to be used?
- Does the Integration Plan mesh with the Verification Plan?
- For each integration step, does the Integration Plan define what components and sub-systems are to be integrated?
- For each integration step, does the Integration Plan identify all the needed participants and define what their roles and responsibilities are?
- Does the Integration Plan establish the sequence and schedule for every integration step?
- Does the Integration Plan spell out how integration problems are to be documented and resolved?

### INTEGRATION PLAN TEMPLATE

Section	Contents
Title Page	<p>The title page should follow the NASA procedures or style guide. At a minimum, it should contain the following information:</p> <p>INTEGRATION PLAN FOR THE [insert name of project] AND [insert name of organization]</p> <p>Contract number</p> <p>Date that the document was formally approved</p> <p>The organization responsible for preparing the document</p> <p>Internal document control number, if available</p> <p>Revision version and date issued</p>
1.0 Purpose of Document	<p>A brief statement of the purpose of this document. It is the plan for integrating the components and sub-systems of the project prior to verification.</p>
2.0 Scope of Project	<p>This section gives a brief description of the planned project and the purpose of the system to be built. Special emphasis is placed on the project's deployment complexities and challenges.</p>
3.0 Integration Strategy	<p>This section informs the reader what the high level plan is for integration and, most importantly, why the integration plan is structured the way it is. As mentioned before, the Integration Plan is subject to several constraints, sometimes conflicting constraints. Also, it is one part of the larger process of build, integrate, verify, and deploy. All of which must be synchronized to support the same project strategy. So, for even a moderately complex project, the integration strategy, based on a clear and concise statement of the project's goals and objectives, is described here at a high, but all-inclusive, level. It may also be necessary to describe the analysis of alternative strategies to make it clear why this particular strategy was selected.</p> <p>The same strategy is the basis for the Build Plan, the Verification Plan, and the Deployment Plan.</p> <p>This section covers and describes each step in the integration process. It describes what components are integrated at each step and gives a general idea of what threads of the operational capabilities, requirements are covered. It ties the plan to the previously identified goals and objectives so the stakeholders can understand the rationale for each integration step. This summary level description also defines the schedule for all the integration efforts.</p>
4.0 Phase 1 Integration	<p>This, and the following sections, define and explain each step in the integration process. The intent here is to identify all the needed participants and to describe to them what they have to do.</p> <p>In general, the description of each integration step should identify:</p> <p>The location of the activities</p> <p>The project-developed equipment and software products to be integrated Initially this is just a high level list but eventually the list must be exact and complete, showing part numbers and quantity</p> <p>Any support equipment [special software, test hardware, software stubs, and drivers to simulate yet-to-be-integrated software components, external systems] needed for this integration step. The same support equipment is most likely needed for the subsequent verification step</p> <p>All integration activities that need to be performed after installation, including integration with on-site systems and external systems at other sites</p> <p>A description of the verification activities, as defined in the applicable Verification Plan, that occur after this integration step</p> <p>The responsible parties for each activity in the integration step</p>

Section	Contents
	The schedule for each activity
5.0 Multiple Phase Integration steps [1 or N steps]	This, and any needed additional sections, follow the format for section 3. Each covers each step in a multiple step integration effort.

## **Appendix 5-4: V&V Plan Sample Outline**

1. INTRODUCTION
  - 1.1 Purpose and Scope
  - 1.2 Responsibility and Change Authority
  - 1.3 Definitions
2. APPLICABLE AND REFERENCE DOCUMENTS
  - 2.1 Applicable Documents
  - 2.2 Reference Documents
  - 2.3 Order of Precedence
3. SYSTEM X DESCRIPTION
  - 3.1 System X Requirements Flowdown
  - 3.2 System X Architecture
  - 3.3 End Item Architectures
    - 3.3.1 System X End Item A
    - 3.3.2 System X End Item i
  - 3.4 System X Ground Support Equipment
  - 3.5 Other Architecture Descriptions
4. VERIFICATION AND VALIDATION PROCESS
  - 4.1 Verification and Validation Management Responsibilities
  - 4.2 Verification Methods
    - 4.2.1 Analysis
    - 4.2.2 Inspection
    - 4.2.3 Demonstration
    - 4.2.4 Test
      - 4.2.4.1 Qualification Testing
  - 4.3 Validation Methods
  - 4.4 Certification Process
  - 4.5 Acceptance Testing
5. VERIFICATION AND VALIDATION IMPLEMENTATION
  - 5.1 SYSTEM X Design and Verification & Validation Flow
  - 5.2 Test Articles
  - 5.3 Support Equipment
  - 5.4 Facilities
6. SYSTEM X END ITEM VERIFICATION AND VALIDATION
  - 6.1 End Item A
    - 6.1.1 Developmental/Engineering Unit Evaluations
    - 6.1.2 Verification Activities
      - 6.1.2.1 Verification Testing
        - 6.1.2.1.1 Qualification Testing
        - 6.1.2.1.2 Other Testing
      - 6.1.2.2 Verification Analysis
        - 6.1.2.2.1 Thermal Analysis
        - 6.1.2.2.2 Stress Analysis
        - 6.1.2.2.3 Analysis of Fracture Control
        - 6.1.2.2.4 Materials Analysis
        - 6.1.2.2.5 EEE Parts Analysis

- 6.1.2.3 Verification Inspection
- 6.1.2.4 Verification Demonstration
- 6.1.3 Validation Activities
- 6.1.4 Acceptance Testing
- 6.2 End Item i
- 7. SYSTEM X VERIFICATION AND VALIDATION
  - 7.1 End Item to End Item Integration
    - 7.1.1 Developmental/Engineering Unit Evaluations
    - 7.1.2 Verification Activities
      - 7.1.2.1 Verification Testing
      - 7.1.2.2 Verification Analysis
      - 7.1.2.3 Verification Inspection
      - 7.1.2.4 Verification Demonstration
    - 7.1.3 Validation Activities
  - 7.2 Complete System Integration
    - 7.2.1 Developmental/Engineering Unit Evaluations
    - 7.2.2 Verification Activities
      - 7.2.2.1 Verification Testing
      - 7.2.2.2 Verification Analysis
      - 7.2.2.3 Verification Inspection
      - 7.2.2.4 Verification Demonstration
    - 7.2.3 Validation Activities
- 8. SYSTEM X PROGRAM VERIFICATION AND VALIDATION
  - 8.1 Vehicle Integration
  - 8.2 End-to-End Integration
  - 8.3 On-Orbit V&V Activities
- 9. SYSTEM X CERTIFICATION PRODUCTS

APPENDIX A—ACRONYMS AND ABBREVIATIONS

APPENDIX B—DEFINITION OF TERMS

APPENDIX C REQUIREMENT VERIFICATION MATRIX

APPENDIX D SYSTEM X VALIDATION MATRIX

## **Appendix 6-1: SEMP Content Outline**

### **SEMP Content**

The SEMP is the foundation document for the technical and engineering activities conducted during the project. The SEMP conveys information on the technical integration methodologies and activities for the project within the scope of the project plan to all of the personnel. Because the SEMP provides the specific technical and management information to understand the technical integration and interfaces, its documentation and approval serves as an agreement within the project of how the technical work will be conducted. Although the development of the SEMP is the responsibility of the technical manager (systems engineering manager??), it incorporates inputs of the project manager and all other project participants. The SEMP the following three general sections:

1. Technical Program Planning and Control which describes the processes for planning and control of the engineering efforts for the design, development, test, and evaluation of the system.
2. Systems Engineering Processes which includes specific tailoring of the systems engineering process as described in the NPR, implementation procedures, trade study methodologies, tools and models to be used.
3. Engineering Specialty Integration describes the integration of the technical disciplines efforts into the Systems engineering process and summarizes each technical discipline effort and cross references each of the specific and relevant plans.

### **Purpose and Scope**

This section provides a brief description of the purpose, scope, and content of the SEMP. The scope encompasses the SE technical effort required to generate the work products necessary to meet the exit criteria for the product-line life-cycle phases. The SEMP is a plan for doing the project technical effort by a technical team for a given WBS Model in the system structure and to help meet life-cycle phase exit criteria.

### **Applicable Documents**

This section of the SEMP lists the documents applicable to this specific project and its SEMP implementation and describes major standards and procedures that this technical effort for this specific project needs to follow. Specific implementation of standardization tasking is incorporated into pertinent sections of the SEMP.

Provide the engineering standards and procedures to be used in the project. Examples of specific procedures could include any hazardous material handling, crew training for control room operations, special instrumentation techniques, special interface documentation for vehicles, maintenance procedures specific to the project.

### **Technical Summary**

This section contains an executive summary describing the problem to be solved by this technical effort and the purpose, context and products of the WBS model to be developed and integrated with other interfacing systems identified.

### **System Description**

This subsection contains a definition of the purpose/mission/objective of the system being developed and a brief description of the purpose of the products of the WBS models of the system structure for which this SEMP applies and the expected scenarios for the system. Each WBS model includes the system end products and their subsystems and the supporting or enabling products and any other work products (plans, baselines) required for the development of the system. The description should include any interfacing systems and system products, including humans, with which the WBS model system products will interact physically, functionally, or electronically.

Identify and document systems constraints, including cost, schedule, technical (for example, environmental, design)

### **System Structure**

This subsection contains an explanation of how the WBS models will be developed; how the resulting WBS model will be integrated into the project WBS; and how the overall system structure will be developed. This subsection contains a description of the relationship of the specification tree and the drawing tree with the products of the system structure and how the relationship and interfaces of the system end products and their life-cycle-enabling products will be managed throughout the planned technical effort.

#### *Product Integration*

This subsection contains an explanation of how the product will be integrated and will describe clear organizational responsibilities and interdependencies whether the organizations are geographically dispersed or managed across Centers This includes identifying organizations, intra-, inter- NASA, other government agencies, contractors or other partners and delineating their roles and responsibilities.

Documentation of when components or elements will be available for integration needs to be clearly understood and identified on the schedule to establish critical schedule issues.

#### *Planning Context*

This section contains the product-line life-cycle model constraints (e.g., NPR 7120.5) that affect the planning and implementation of the common technical processes to be applied in performing the technical effort. The constraints provide a linkage of the technical effort with the applicable product-line life-cycle phases covered by the SEMP including, as applicable, milestone decision gates, major technical reviews, key intermediate events leading to project completion, life-cycle phase, event entry and exit criteria, and major baseline and other work products to be delivered to the sponsor or customer of the technical effort.

### *Boundary of Technical Effort*

This subsection contains a description of the boundary of the general problem to be solved by the technical effort. Specifically, it identifies what can be controlled by the technical team (inside the boundary) and what influences the technical effort and is influenced by the technical effort but not controlled by the technical team (outside the boundary). Specific attention should be given to physical, functional, and electronic interfaces across the boundary.

Define the system to be addressed. A description of the boundary of the system can include the following: definition of internal and external elements/items involved in realizing the system purpose as well as the system boundaries in terms of space, time, physical, operational. Also, identification of what initiates the transitions of the system to operational status and what initiates its disposal is important. The following is a general listing of other items to include, as appropriate:

- General and functional descriptions of the subsystems
- Document current and established subsystem performance characteristics
- Identify and document current interfaces and characteristics
- Develop functional interface descriptions and functional flow diagrams
- Identify key performance interface characteristics
- Identify current integration strategies and architecture

### *Cross References*

This subsection contains cross references to appropriate non-technical plans and critical reference material that interface with the technical effort. This section contains a summary description of how the technical activities covered in other plans are accomplished as fully integrated parts of the technical effort.

### **Technical Effort Integration**

This section contains a description of how the various inputs to the technical effort will be integrated into a coordinated effort that meets cost, schedule, and performance objectives.

This section should describe the integration and coordination of the efforts of the specialty engineering disciplines into the systems engineering process during each iteration of the processes. Where there is potential for overlap of specialty efforts, the SEMP should define the relative responsibilities and authorities of each. This section should contain, as needed, the project's approach to the following:

- Concurrent engineering
- The activity phasing of specialty engineering
- The participation of specialty disciplines
- The involvement of specialty disciplines
- The role and responsibility of specialty disciplines
- The participation of specialty disciplines in system decomposition and definition
- The role of specialty disciplines in verification and validation
- Reliability

- Maintainability
- Quality assurance
- Integrated logistics
- Human engineering
- Safety
- Producibility
- Survivability/vulnerability
- Environmental assessment
- Launch approval/flight readiness

Provide the approach for coordination of diverse technical disciplines and integration of the development tasks. For example, this can include the use of integrated teaming approaches. Ensure that the specialty engineering disciplines are properly represented on all technical teams and during all life-cycle phases of the project. Define the scope and timing of the specialty engineering tasks.

#### *Responsibility and Authority*

This subsection contains a description of the organizing structure for the technical teams assigned to this technical effort and includes how the teams will be staffed and managed, including (a) what organization/panel will serve as the DGA for this project and, therefore, will have final signature authority for this SEMP; (b) how multidisciplinary teamwork will be achieved; (c) identification and definition of roles, responsibilities, and authorities required to perform the activities of each planned common technical process; (d) planned technical staffing by discipline and expertise level, with human resource loading; (e) required technical staff training; and (f) assignment of roles, responsibilities, and authorities to appropriate project stakeholders or technical teams to ensure planned activities are accomplished.

Provide an organization chart and denote who is responsible for which activities on the team. Indicate the lines of authority and responsibility. Define the resolution authority to make decisions/decision process. Show how the engineers/engineering disciplines relate.

The systems engineering roles and responsibilities need to be addressed for the following: project office, user, Contracting Office Technical Representative (COTR), systems engineering, design engineering, specialty engineering, contractor

#### *Contractor Integration*

This subsection contains a description of how the technical effort of in-house and external contractors is to be integrated with the NASA technical team efforts. This includes establishing technical agreements, monitoring contractor progress against the agreement, handling technical work or product requirements change requests, and acceptance of deliverables. The section will specifically address how interfaces between the NASA technical team and the contractor will be implemented for each of the 17 common technical processes. For example, it addresses how the NASA technical team will be involved with reviewing or controlling contractor-generated design solution definition documentation or how the technical team will be involved with product verification and product validation activities.

Key deliverables for the contractor to complete their systems and those required of the contractor for other project participants need to be identified and established on the schedule.

### *Support Integration*

This subsection contains a description of the methods (such as integrated computer-aided tool sets, integrated work product databases, and technical management information systems) that will be used to support technical effort integration.

**Common Technical Processes Implementation** Each of the 17 common technical processes will have a separate subsection that contains the plan for performing the required process activities as appropriately tailored. (See NPR 7123.1 chapter 3 for the process activities required and Appendix F for tailoring.) Implementation of the 17 common technical processes includes (1) the generation of the outcomes needed to satisfy the entry and exit criteria of the applicable product-line life-cycle phase or phases identified in D.4.4.4 and (2) the necessary inputs for other technical processes. These sections contain a description of the approach, methods, and tools for:

- a. Identifying and obtaining adequate human and non-human resources for performing the planned process, developing the work products, and providing the services of the process.
- b. Assigning responsibility and authority for performing the planned process, developing the work products, and providing the services of the process.
- c. Training the technical staff performing or supporting the process, where training is identified as needed.
- d. Designating and placing designated work products of the process under appropriate levels of configuration management.
- e. Identifying and involving stakeholders of the process.
- f. Monitoring and controlling the process.
- g. Identifying, defining and tracking metrics and success.
- h. Objectively evaluating adherence of the process and the work products and services of the process to the applicable requirements, objectives, and standards and addressing noncompliance.
- i. Reviewing activities, status, and results of the process with appropriate levels of management and resolving issues.

This section should also include the project specific description of each of the 17 processes to be used, including the specific tailoring of the requirements to the system and the project, the procedures to be used in implementing the processes; in-house documentation; trade study methodology; types of mathematical and/or simulation models to be used; and generation of specifications.

### **Technology Insertion**

This section contains a description of the approach and methods for identifying key technologies and their associated risks and criteria for assessing and inserting technologies, including those for inserting critical technologies from technology development projects. An approach should be developed for appropriate level and timing of technology insertion. This could include alternative approaches to take advantage of new technologies to meet systems needs as well as alternative options if the technologies do not prove appropriate in result or timing. The strategy for an initial technology assessment within the scope of the project requirements should be provided to identify technology constraints for the system

### **Additional SE Functions and Activities**

This section contains a description of other areas not specifically included in previous sections but that are essential for proper planning and conduct of the overall technical effort.

#### *System Safety*

This subsection contains a description of the approach and methods for conducting safety analysis and assessing the risk to operators, the system, the environment, or the public.

#### *Engineering Methods and Tools*

This subsection contains a description of the methods and tools not included in the Technology Insertion section that are needed to support the overall technical effort and identifies those tools to be acquired and tool training requirements.

Define the development environment for the project, including automation and software tools. If required, develop and/or acquire the tools and facilities for all disciplines on the project. Standardize when possible across the project, or enable a common output format of the tools that can be used as input by a broad range of tools used on the project. Define the requirements for information management systems and for using existing elements. Define and plan for the training required to use the tools and technology across the project.

#### *Specialty Engineering*

This subsection contains a description of engineering discipline and specialty requirements that apply across projects and the WBS models of the system structure. Examples of these requirement areas would include planning for safety, reliability, human factors, logistics, maintainability, quality, operability, and supportability. Estimate staffing levels for these disciplines and incorporate with the project requirements.

### **Integration with the Project Plan and Technical Resource Allocation**

This section contains how the technical effort will integrate with project management and defines roles and responsibilities. This section addresses how technical requirements will be integrated with the project plan to determinate the allocation of resources, including cost, schedule, and personnel, and how changes to the allocations will be coordinated.

This section describes the interface between all of the technical aspects of the project and the overall project management process during the systems engineering planning activities and

updates. All activities to coordinate technical efforts with the overall project are included, such as, technical interactions with the external customers, users, contractors.

### **Waivers**

This section contains all approved waivers to the Center Directors SE NPR Implementation Plan requirement for the SEMP. This section also contains a separate subsection that includes any tailored SE NPR requirements that are not related and able to be documented in a specific SEMP section or subsection.

### **Appendices**

Appendices are included, as necessary, to provide a glossary, acronyms and abbreviations, and information published separately for convenience in document maintenance. Included would be: (a) information that may be pertinent to multiple topic areas (e.g., description of methods or procedures); (b) charts and proprietary data applicable to the technical efforts required in the SEMP; and (c) a summary of technical plans associated with the project. Each appendix should be referenced in one of the sections of the engineering plan where data would normally have been provided.

### **Reference/Template**

MIL-HBK-881 Work Breakdown Structures for Defense Material Items, 30 July 2005  
The Project Management Institute's Practice Standards for Work Breakdown Structures  
DAU Systems Engineering Fundamentals  
DOE CIO document, chapter 3, page 47 Planning Stage, CIO.doe.gov  
NASA Cost Estimating Handbook

### **SEMP Preparation Checklist**

The SEMP, as the key reference document capturing the technical planning, some basic topics need to be addressed. The following are five suggested topic areas to resolve prior to developing and documenting the systems engineering process in the SEMP. (derived from "Systems Engineering Guidebook" by James Martin)

1. Characterize the problem
  - a. Define the problem to be solved by the project
  - b. Delineate all of the influencing factors
  - c. List the critical questions
  - d. List the overall project constraints including cost, schedule and technical performance
  - e. Identify how the team will know when the problem is adequately defined
  - f. Identify how the team will know when the problem has been adequately solved.
  - g. Define the customers, stakeholders and the users
  - h. Identify the priorities of the customers, stakeholders and users
  - i. Define the relationship between this project and other projects
2. Identify the integration relationships on the Project

- a. Define how the project will be structured to enable the solution of the problem within the constraints, including cost and schedule
  - b. Define the value added of systems engineering to this project
  - c. Define how the various disciplines required on the project will be integrated
  - d. Define the special knowledge, skills, abilities and training that will be required
3. Identify the methodologies, techniques and parameters for assessment of project progress
- a. Define metrics to be used to measure technical performance
  - b. Define metrics that will be used to identify opportunities for process improvements
  - c. Define how progress will be measured against plans and schedules
  - d. Define the progress reporting increments, who will report and to whom will it be reported.
  - e. Define how risk will be assessed. Define the thresholds for triggering mitigation activities. Define a plan for integrating risk management into the technical decision process.
  - f. Define the communication processes inside and outside of the project
  - g. Define the method for recording decisions and maintaining the decision in a repository
  - h. Define methodology for incorporating lessons learned from other projects and for communicating lessons learned from this project to others.
4. Process Definition
- a. Define the systems engineering process for this project
  - b. Define the methods that will be used/applied for each systems engineering task
  - c. Identify the tools required to support these methods and provide a plan for their implementation/integration into the project
  - d. Determine how to control configuration development
  - e. Determine how and when technical reviews will be conducted
  - f. Determine how required trade-off studies will be identified and managed
  - g. Identify who has responsibility for and can authorize technical changes and how the process will be controlled
  - h. Identify the process for management of requirements, interfaces and configuration and documentation management

## 5. Technology Insertion

- a. Identify the process, including how and when new or special technology can be inserted into a project and what will be the minimum insertion requirements.
- b. Determine the role of research and development labs to the project, including how they will support the project and how the project will incorporate their results
- c. Establish the schedule for technology insertion and integrate it with the technical schedule
- d. Establish a plan for incorporating system elements provided by the customer and/or the user. Include the approach to determine the appropriateness or adequacy of these items.
- e. Determine what technical facilities are required to be used by the project
- f. Determine how project transitions will be made (development to implementation; implementation to operation; operation to disposal)

## **Appendix 6-3: Interface Requirements Document Outline**

### **1.0 INTRODUCTION**

#### **1.1 PURPOSE AND SCOPE**

State the purpose of this document and briefly identify the interface to be defined herein. For example, (“This IRD defines and controls the interface(s) requirements between \_\_\_\_\_ and \_\_\_\_\_”).

#### **1.2 PRECEDENCE**

Define the relationship of this document to other Program documents and specify which is controlling in the event of a conflict.

#### **1.3 RESPONSIBILITY AND CHANGE AUTHORITY**

State the responsibilities of the interfacing organizations for development of this document and its contents. Define document approval authority (including change approval authority).

### **2.0 DOCUMENT**

#### **2.1 APPLICABLE DOCUMENTS**

List binding documents which are invoked to the extent specified within this IRD. The latest revision or most recent version should be listed. Documents and requirements imposed by higher-level documents (higher order of precedence) shall not be repeated.

#### **2.2 REFERENCE DOCUMENTS**

List any document that is referenced in the text of this section.

### **3.0 INTERFACES**

#### **3.1 GENERAL**

In the subsections that follow, provide the detailed description, the responsibilities, coordinate systems, and the numerical requirements as it relates to the interface plane.

##### **3.1.1 INTERFACE DESCRIPTION**

In this section describe the interface as defined in the System Specification. Use and refer to figures, drawings, or tables for numerical information.

##### **3.1.2 INTERFACE RESPONSIBILITIES**

Define interface hardware and interface boundary responsibilities to depict the interface plane. Use tables, figures, or drawings as appropriate.

### 3.1.3 COORDINATE SYSTEMS

Define the coordinate system used for interface requirements on each side of the interface. When a component coordinate system is already controlled by TBD,

Reference Coordinate Systems, simply refer to it here.

### 3.1.4 ENGINEERING UNITS, TOLERANCES, AND CONVERSIONS

Numerical data entered in all IRDs shall be in English (inch–pound–second) units with applicable tolerances. The equivalent International System of Units (SI) (metric) value shall be added in parenthesis for IRDs. For example, THA–9lbs (4.1KG), Conversions between English and S1 units shall be in accordance with ASTM E–380, standard procedure for use of the SI.

## 3.2 INTERFACE REQUIREMENTS

Define structural limiting values at the interface, such as interface loads, forcing functions, and dynamic conditions. Refer to TBD, Structural Design Loads Data Book for additional data.

### 3.2.1 INTERFACE PLAN

Document the interface requirements on each side of the interface plane in the sections below.

#### 3.2.1.1 ENVELOPE

#### 3.2.1.2 MASS PROPERTIES

In this section of the IRD, document the derived interface requirements based on the allocated requirements contained in the applicable segment specification pertaining to that side of the interface. For example, this section of the IRD should cover the weight of the element in pounds or the metric equivalent on this side of the interface.

#### 3.2.1.3 STRUCTURAL/MECHANICAL

In this section of the IRD, document the derived interface requirements based on the allocated requirements contained in the applicable segment specification pertaining to that side of the interface. For example, this section of the IRD should cover attachment, stiffness, latching and mechanisms, etc.

#### 3.2.1.4 FLUID

In this section of the IRD, document the derived interface requirements based on the allocated requirements contained in the applicable segment specification pertaining to that side of the interface. For example, this section of the IRD should cover fluid areas such as thermal control, O2 and N2, potable and waste water, fuel cell water, atmospheric sampling, etc.

### 3.2.1.5 ELECTRICAL (POWER)

In this section of the IRD, document the derived interface requirements based on the allocated requirements contained in the applicable segment specification pertaining to that side of the interface. For example, this section of the IRD should cover various electric current, voltage, wattage and resistance levels, etc.

### 3.2.1.6 ELECTRONIC (SIGNAL)

In this section of the IRD, document the derived interface requirements based on the allocated requirements contained in the applicable segment specification pertaining to that side of the interface. For example, this section of the IRD should cover various signal types such as audio, video, command data handling, navigation, etc.

### 3.2.1.7 SOFTWARE AND DATA

In this section of the IRD, document the derived interface requirements based on the allocated requirements contained in the applicable segment specification pertaining to that side of the interface. For example, this section of the IRD should cover various data standards, message timing, protocols, error detection/correction, functions, initialization, status, etc.

### 3.2.1.8 ENVIRONMENTS

In this section of the IRD, document the derived interface requirements based on the allocated requirements contained in the applicable segment specification pertaining to that side of the interface. For example, this section of the IRD should cover the dynamic envelope measures of the element in English or the metric equivalent on this side of the interface.

#### 3.2.1.8.1 ELECTROMAGNETIC EFFECTS

##### 3.2.1.8.1.1 ELECTROMAGNETIC COMPATIBILITY

The end item 1 to end item 2 interface shall meet the requirements of TBD, Systems Requirements for Electromagnetic Compatibility.

##### 3.2.1.8.1.2 ELECTROMAGNETIC INTERFERENCE

The end item 1 to end item 2 interface shall meet the requirements of TBD, Electromagnetic Emission, and Susceptibility Requirements for Electromagnetic Compatibility.

##### 3.2.1.8.1.3 GROUNDING

The end item 1 to end item two 2 interface shall meet the requirements of TBD, Grounding Requirements.

##### 3.2.1.8.1.4 BONDING

The end item 1 to end item 2 structural/mechanical interface shall meet the requirements of TBD, Electrical Bonding Requirements.

#### 3.2.1.8.1.5 CABLE AND WIRE DESIGN

The end item 1 to end item 2 cable and wire interface shall meet the requirements of TBD, Cable/Wire Design, and Control Requirements for Electromagnetic Compatibility.

#### 3.2.1.8.2 ACOUSTIC

The acoustic noise levels on each side of the interface shall be required to meet the Program or Project requirements.

#### 3.2.1.8.3 STRUCTURAL LOADS

End item 1 to end item 2 mated loads that each end item must accommodate.

#### 3.2.1.8.4 VIBROACOUSTICS

Account vibro-acoustic loads.

#### 3.2.1.9 OTHER TYPES OF INTERFACE REQUIREMENTS

In this section of the IRD, document other types of unique interface requirements that may be applicable.

## Appendix 6-5: CM Plan Outline

A typical CM Plan should include the following:

Section	Description
Section 1.0 Introduction	This section includes; The purpose and scope of the CM Plan and the program phases to which it applies. Brief description of the system or top level CIs
Section 2.0 Applicable and Reference Documents	This section includes a list of the specifications, standards, manual, and other documents, referenced in the Plan by title, document number, issuing authority, revision, and as applicable, change notice, amendment and issue date.
Section 3.0 CM Concept of Operations and Organization	This section includes: CM objectives Information needed to support the achievement of objectives in the current and future phases Description and graphic portraying the Projects planned organization with emphasis on the CM activities
Section 4.0 CM Process CM Management and Planning Configuration Identification Configuration Control Configuration Status Accounting Configuration Audits	This section includes a description of the Project's CM process for accomplishing the five CM Activities which includes but not limited to: CM activities for the current and future phases Baselines Configuration Items Establishment and membership of Configuration Control Boards Nomenclature and numbering Hardware/Software Identification FCAs and PCAs
Section 5.0 Data Management	This section includes: Data Identification Data Preparation Data Control Data Access and Records
Section 6.0 Interface Management	This section includes a description on how CM will maintain and control Interface Control Documents (ICDs).

## **Appendix 6-7: Guidance on Technical Peer Reviews/Inspections**

### **Introduction**

The objective of technical peer reviews/inspections is to remove defects as early as possible in the development process. Peer reviews/inspections are a well defined review process for finding and fixing defects, conducted by a team of peers with assigned roles, each having a vested interest in the work product under review. Peer reviews/inspections are held within development phases, between milestone reviews, on completed products or completed portions of products. The results of peer reviews/inspections can be reported at milestone reviews. Checklists are heavily utilized in peer reviews/inspections to improve the quality of the review.

Technical peer reviews/inspections have proven over time to be one of the most effective practices available for ensuring quality products and on-time deliveries. Many studies have demonstrated their benefits, both within NASA and across industry. Peer reviews/inspections improve quality and reduce cost by reducing rework. The studies have shown that the rework effort saved not only pays for the effort spent on inspections, but also provides additional cost savings on the project. By removing defects at their origin (e.g., requirements and design documents, test plans and procedures, software code, etc.), inspections prevent defects from propagating through multiple phases and work products, and reduce the overall amount of rework necessary on projects. In addition, improved team efficiency is a side-effect of peer reviews/inspections (e.g., by improving team communication, more quickly bringing new members up to speed, and educating project members about effective development practices).

### **How to Perform Technical Peer Reviews/Inspections**

Figure 1 shows a diagram of the peer review/inspection stages and the text below the figure explains how to perform each of the stages.

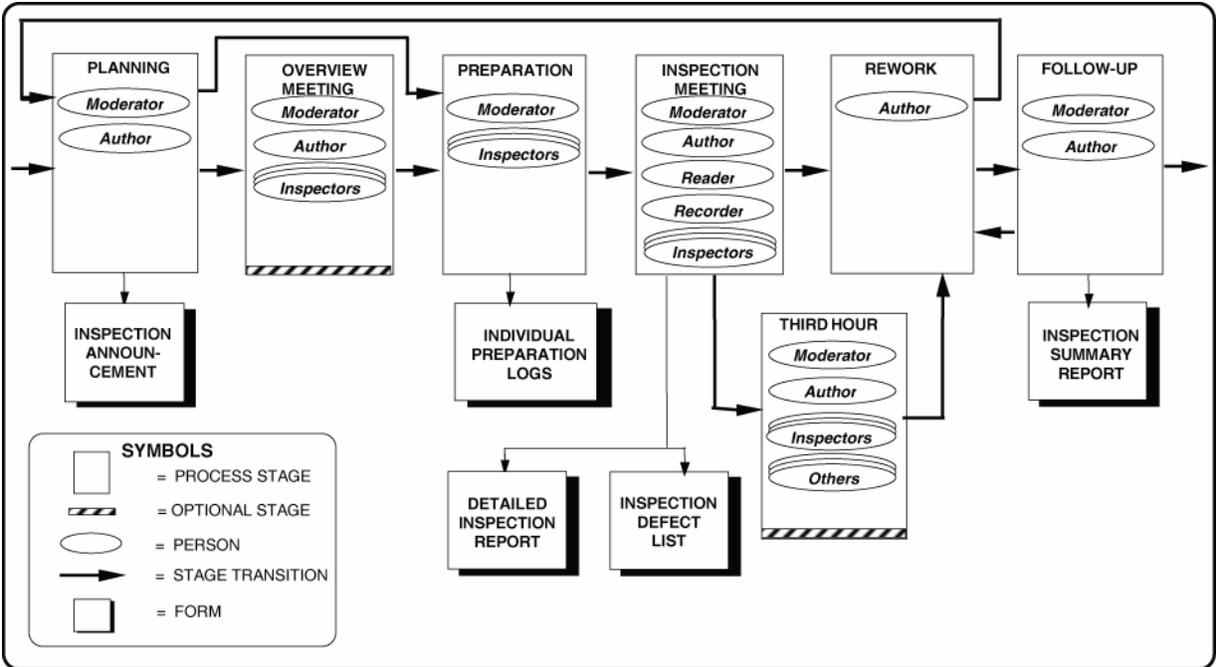


Figure 1. The Formal Inspection Process

It is recommended that the Moderator review the “Planning Inspection Schedule and Estimating Staff Hours,” “Guidelines for Successful Inspections,” and “10 Basic Rules of Inspections” in the Quick Reference Guide below before beginning the planning stage. (Note: *NPR 7150: NASA Software Engineering Requirements* defines Agency requirements on the use of Peer Reviews/Inspections. NASA peer review/inspection training is offered by the NASA Office of Chief Engineer/HQ. InSpec, a tool for partly automating the peer review/inspection process, is available from the NASA Independent Verification and Validation Facility.)

### A.—Planning

The Moderator of the peer review/inspection performs the following activities:

- a. Determine whether peer review/inspection entrance criteria have been met.
- b. Determine whether an overview of the product is needed.
- c. Select the peer review/inspection team and assign roles. For guidance on roles, see “Roles of Participants” in the Quick Reference Guide below. Reviewers have a vested interest in the work product (e.g., they are peers representing areas of the life cycle affected by the material being reviewed).
- d. Determine if the size of the product is within the prescribed guidelines for the type of inspection (see the “Meeting Rate Guidelines” in the Quick Reference Guide below for guidelines on the optimal number of pages or lines of code to inspect for each type of inspection). If the product exceeds the prescribed guidelines, break the product into parts and inspect each part separately. (It is highly recommended that the peer review/inspection meeting not exceed 2 hours).

- e. Schedules the overview (if one is needed).
- f. Schedule peer review/inspection meeting time and place.
- g. Prepare and distribute the Inspection Announcement<sup>1</sup> and package. Include in the package the product to be reviewed and the appropriate checklist for the peer review/inspection (see URL: ??? for appropriate checklist).
- h. Record total time spent in Planning\*.

### **B.—Overview Meeting**

- a. Moderator runs the meeting and the Author presents background information to the reviewers.
- b. Record total time spent in the Overview\*.

### **C.—Peer Review/Inspection Preparation**

- a. Peers review the checklist definitions of defects.
- b. Examine materials for understanding and possible defects.
- c. Prepare for assigned role in peer review/inspection.
- d. Complete and turn in Individual Preparation Log<sup>2</sup> to the Moderator.
- e. The Moderator reviews the Individual Preparation Logs and makes “Go/No Go” decision and organizes inspection meeting.
- f. Record total time spent in the Preparation\*.

### **D.—Peer Review/Inspection Meeting**

- a. The Moderator introduces people and identifies their peer review/inspection roles.
- b. The Reader presents work products to the peer review/inspection team in a logical and orderly manner.
- c. Peer reviewers/Inspectors find and classify defects by severity, category, and type (see “Classification of Defects” in the Quick Reference Guide).
- d. The Recorder writes the major and minor defects on the Inspection Defect List<sup>3</sup> (for definitions of major and minor, see the “Severity” section of the Quick Reference Guide).

---

<sup>1</sup> Langley Research Center, *Instructional Handbook for Formal Inspections*. This document provides more detailed instructions on how to perform technical peer reviews/inspections. It also provides templates for the forms used in the peer review/inspection process described above: Inspection Announcement, Individual Preparation Log, Inspection Defect List, Detailed Inspection Report, and the Inspection Summary Report. The document is available at URL: <<http://sw-eng.larc.nasa.gov/process/docslistnew.cfm>>.

<sup>2</sup> Ibid.

- e. Steps b through d are repeated until the review of the product is completed.
- f. Open issues are assigned to Peer reviewers/Inspectors if irresolvable discrepancies occur.
- g. Summarize the number of defects and their classification on the Detailed Inspection Report.<sup>4</sup>
- h. Determine the need for a re-inspection or third-hour. Optional: Trivial defects (e.g., redlined documents) can be given directly to the Author at the end of the inspection.
- i. The Moderator obtains an estimate for rework time and completion date from the Author, and does the same for action items if appropriate.
- j. The Moderator assigns writing of change requests and/or problem reports (if needed).
- k. Record total time spent in the peer review/inspection meeting\*.

**E.—Third-Hour**

- a. Complete assigned action items and provide information to the Author.
- b. Attend Third-hour meeting at Author's request.
- c. Provide time spent in Third-hour to the Moderator\*.

**F.—Rework**

- a. All major defects noted in the Inspection Defect List are resolved by the Author.
- b. Minor and trivial defects (which would not result in faulty execution) are resolved at the discretion of the Author as time and cost permit.
- c. Record total time spent in the Rework on the Inspection Defect List.<sup>5</sup>

**G.—Follow-up**

- a. The Moderator verifies all major defects have been corrected and no secondary defects have been introduced.
- b. The Moderator ensures all open issues are resolved and verifies all exit criteria for the peer review/inspection are met.
- c. Record total time spent in Rework and Follow-up\*
- d. File the inspection package.
- e. The Inspection Summary Report<sup>6</sup> is distributed.

---

<sup>3</sup> Ibid.

<sup>4</sup> Ibid.

<sup>5</sup> Ibid.

f. Communicate the peer review/inspection has been passed.

\*The Moderator records the time on the Inspection Summary Report.<sup>7</sup>

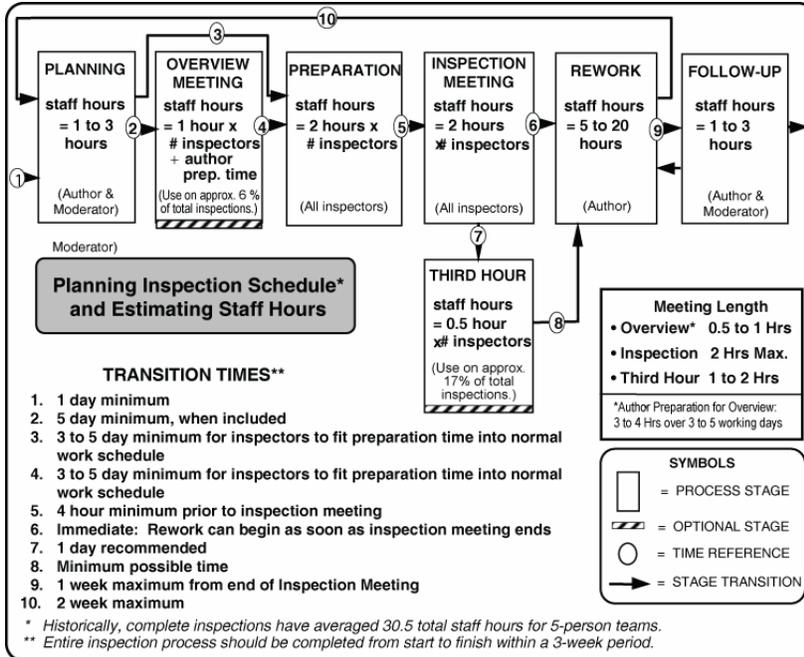
---

<sup>6</sup> Ibid.

<sup>7</sup> Ibid.

# Peer Reviews/Inspections

## QUICK REFERENCE GUIDE



### Guidelines for Successful Inspections

- Train moderators, inspectors & managers
- No more than 25% of developers' time should be devoted to inspections
- Inspect 100% of work product
- Be prepared
- Share responsibility for work product quality
- Be willing to associate and communicate
- Avoid judgmental language
- Do not evaluate author
- Have at least one positive & negative input
- Raise issues; don't resolve them
- Avoid discussions of style
- Stick to standard or change it
- Be technically competent
- Record all issues in public
- Stick to technical issues
- Distribute inspection documents as soon as possible
- Let author determine when work product is ready for inspection
- Keep accurate statistics

### 10 Basic Rules of Inspections:

- Inspections are carried out at a number of points inside phases of the life cycle. Inspections are not substitutes for milestone reviews.
- Inspections are carried out by peers representing areas of life cycle affected by material being inspected (usually limited to 6 or fewer people). All inspectors should have a vested interest in the work product.
- Management is not present during inspections. Inspections are not to be used as a tool to evaluate workers.
- Inspections are led by a trained moderator.
- Trained inspectors are assigned roles.
- Inspections are carried out in a prescribed series of steps.
- Inspection Meeting is limited to 2 hours.
- Checklists of questions are used to define task and to stimulate defect finding.
- Material is covered during Inspection Meeting within an optional page rate which has been found to give maximum error finding ability.
- Statistics on number of defects, types of defects, and time expended by engineers on inspections are kept.

### Meeting\* Rate Guidelines for Various Inspection Types

Type	Inspection Meeting	
	Target per 2 Hrs	Range per 2 Hrs
R0	20 Pages	10 to 30 Pages
R1	20 Pages	10 to 30 Pages
I0	30 Pages	20 to 40 Pages
I1	35 Pages	25 to 45 Pages
I2	500 Lines of Source Code**	400 to 600 Lines of Source Code**
IT1	30 Pages	20 to 40 Pages
IT2	35 Pages	25 to 45 Pages

\* Assumes a 2 hour meeting. Scale down planned meeting duration for shorter work products.  
\*\* Flight Software and other highly complex code segments should proceed at about half this rate.

### Roles of Participants

**Moderator**  
Responsible for conducting inspection process and collecting inspection data. Plays key role in all stages of process except rework. Required to perform special duties during an inspection in addition to inspector's tasks.

**Inspectors**  
Responsible for finding defects in work product from a general point of view, as well as defects which affect their area of expertise.

**Author**  
Provides information about work product during all stages of process. Responsible for correcting all major defects and any minor and trivial defects which cost and schedule permit. Performs duties of an inspector.

**Reader**  
Guides team through work product during inspection meeting. Reads or paraphrases work product in detail. Should be an inspector from same (or next) life cycle phase as author. Performs duties of an inspector in addition to reader's role.

**Recorder**  
Accurately records each defect found during inspection meeting on INSPECTION DEFECT LIST. Performs duties of an inspector in addition to recorder's role.

### Types of Inspections

- SY1 System Requirements
- SY2 System Design
- SU1 Subsystem Requirements
- SU2 Subsystem Design
- R1 Software Requirements
- I0 Architectural Design
- I1 Detailed Design
- I2 Source Code
- IT1 Test Plan
- IT2 Test Procedures & Functions

### Classification of Defects:

#### Severity

##### Major

- An error which would cause a malfunction or prevents attainment of an expected or specified result.
- Any error which would in the future result in an approved change request or failure report.

##### Minor

- A violation of standards, guidelines, or rules, which would not result in a deviation from requirements if not corrected, but could result in difficulties in terms of operations, maintenance, or future development.

##### Trivial

- Editorial errors such as spelling, punctuation, and grammar which do not cause errors or change requests. Recorded only as red-lines. Presented directly to author.

Author is required to **correct all major defects** and should correct minor and trivial defects as time and cost permit.

#### Category

- Missing
- Wrong
- Extra

#### Type

Types of defects are derived from headings on checklist used for the inspection. Defect types can be standardized across inspections from all phases of the life cycle. A suggested standard set of defect types are:

- Clarity
- Completeness
- Compliance
- Consistency
- Correctness / Logic
- Data Usage
- Fault Tolerance
- Functionality
- Interface
- Level of Detail
- Maintainability
- Performance
- Reliability
- Testability
- Traceability
- Other

#### EXAMPLE

The following is an example of a defect classification that would be recorded on the *Inspection Defect List*:

Description	Classification																				
Line 169 -- While counting the number of leading spaces in variable NAME, the wrong "I" is used to calculate "J".	<table border="0"> <tr> <td>Major Defect</td> <td><input checked="" type="checkbox"/></td> <td>Missing</td> <td><input type="checkbox"/></td> </tr> <tr> <td>Minor Defect</td> <td><input type="checkbox"/></td> <td>Wrong</td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>Open Issue</td> <td><input type="checkbox"/></td> <td>Extra</td> <td><input type="checkbox"/></td> </tr> <tr> <td>Type</td> <td colspan="3">Data Usage</td> </tr> <tr> <td>Origin</td> <td colspan="3"></td> </tr> </table>	Major Defect	<input checked="" type="checkbox"/>	Missing	<input type="checkbox"/>	Minor Defect	<input type="checkbox"/>	Wrong	<input checked="" type="checkbox"/>	Open Issue	<input type="checkbox"/>	Extra	<input type="checkbox"/>	Type	Data Usage			Origin			
Major Defect	<input checked="" type="checkbox"/>	Missing	<input type="checkbox"/>																		
Minor Defect	<input type="checkbox"/>	Wrong	<input checked="" type="checkbox"/>																		
Open Issue	<input type="checkbox"/>	Extra	<input type="checkbox"/>																		
Type	Data Usage																				
Origin																					

Based on: JCK/LLW/SSPHS: 10/92  
Based on: JCK/LLW/SSPHS: 10/92

## **Appendix 7-4: Project Protection Plan Outline**

The following outline will assist systems engineers in preparing a project protection plan (PPP). The PPP is a living document that will be written and updated as the project progresses through major milestones and ultimately through end-of-life.

### **1. INTRODUCTION**

- 1.1 PROTECTION PLAN OVERVIEW
- 1.2 PROJECT OVERVIEW
- 1.3 ACQUISITION STATUS

### **2. REFERENCES**

- 2.1 DIRECTIVES AND INSTRUCTIONS
- 2.2 REQUIREMENTS
- 2.3 STUDIES AND ANALYSES

### **3. REFERENCES**

- 3.1 THREATS: HOSTILE ACTION
  - 3.1.1 *Overview*
  - 3.1.2 *Threat Characterization*
    - 3.1.2.1 *Cyber Attack*
    - 3.1.2.2 *Electronic Attack*
    - 3.1.2.3 *Lasers*
    - 3.1.2.4 *Ground Attack*
    - 3.1.2.5 *Asymmetric Attack on Critical Commercial Infrastructure*
    - 3.1.2.6 *Anti-Satellite Weapons*
    - 3.1.2.7 *High-Energy Radio Frequency Weapons*
    - 3.1.2.8 *Artificially Enhanced Radiation Environment*
- 3.2 THREATS: ENVIRONMENTAL
  - 3.2.1 *Overview*
  - 3.2.2 *Threat Characterization*
    - 3.2.2.1 *Natural Environment Storms*
    - 3.2.2.2 *Earthquakes*
    - 3.2.2.3 *Floods*
    - 3.2.2.4 *Fires*
    - 3.2.2.5 *Radiation Effects in the Natural Environment*
    - 3.2.2.6 *Radiation Effects to Spacecraft Electronics*

### **4. PROTECTION VULNERABILITIES**

- 4.1 GROUND SEGMENT VULNERABILITIES
  - 4.1.1 *Command and Control Facilities*
  - 4.1.2 *Remote Tracking Stations*
  - 4.1.3 *Spacecraft Simulator(s)*
  - 4.1.4 *Mission Data Processing Facilities*
  - 4.1.5 *Flight Dynamic Facilities*
  - 4.1.6 *Flight Software Production/Verification/Validation Facilities*

## 4.2 COMMUNICATIONS/INFORMATION SEGMENT VULNERABILITIES

4.2.1 *Command Link*

4.2.2 *Telemetry Link (Mission Data)*

4.2.3 *Telemetry Link (Engineering Data)*

4.2.4 *Ground Network*

## 4.3 SPACE SEGMENT VULNERABILITIES

4.3.1 *Spacecraft Physical Characteristics*

4.3.2 *Spacecraft Operational Characteristics*

4.3.3 *Orbital Parameters*

4.3.4 *Optical Devices (Sensors/Transmitters/Receivers)*

4.3.5 *Communications Subsystem*

4.3.6 *Command and Data Handling Subsystem*

4.3.7 *Instruments*

## 4.4 LAUNCH SEGMENT VULNERABILITIES

4.4.1 *Launch Parameters*

4.4.2 *Launch Site Integration and Test Activities*

## 4.5 COMMERCIAL INFRASTRUCTURE VULNERABILITIES

4.5.1 *Electrical Power*

4.5.2 *Natural Gas*

4.5.3 *Telecommunications*

4.5.4 *Transportation*

## 5. PROTECTION COUNTERMEASURES

5.1 PROTECTION STRATEGY

5.2 MISSION THREAT MITIGATION

5.3 MISSION RESTORATION OPTIONS

5.4 MISSION SURVIVABILITY CHARACTERISTICS

## 6. DEBRIS MITIGATION

6.1 DESIGN GUIDELINES

6.2 END-OF-LIFE MITIGATION PROCEDURES

6.3 COLLISION AVOIDANCE

## 7. CRITICAL PROGRAM INFORMATION AND TECHNOLOGIES

7.1 CRITICAL PROGRAM INFORMATION ELEMENTS

7.2 CRITICAL INFORMATION PROGRAM

## 8. PROGRAM PROTECTION COSTS

8.1 SYSTEMS TRADES ANALYSES

8.2 COST/BENEFIT ANALYSES

## Bibliography/Suggested Reading

### Selected Systems Engineering Reading

#### Fundamentals of Systems Engineering

Blanchard, Benjamine S. and Wolter J. Fabrycky. *Systems Engineering and Analysis*. 5th ed. Englewood Cliffs, NJ:Prentice Hall, 2005.

Sage, Andrew P. *Systems Engineering*. New York:Wiley, 1992.

Sage, Andrew P. and James E. Armstrong, Jr. *Introduction to Systems Engineering*. New York:Wiley, 2000.

#### Management Issues in Systems Engineering

Government Electronics and Information Technology Association. *Processes for Engineering a System*, ANSI/GEIA EIA-632. 2003/

IEEE. *IEEE Standard for Application and Management of the Systems Engineering Process*. IEEE Std 1220-2005. Defense Systems Management College. *Systems Engineering Management Guide* (DTIC #ADA 223-168). Fort Belvoir, VA, 1990.

Blanchard, Benjamin S. *System Engineering Management*. 3<sup>rd</sup> ed. New York:Wiley, 2003.

Chestnut, Harold. *Systems Engineering Methods*. New York:Wiley, 1967.

Miles, Ralph, Jr. (editor). *Systems Concepts: Lectures on Contemporary Approaches to Systems*. New York:Wiley, 1973.

Successful Systems Engineering for Engineers and Managers, Norman B. Reilly

#### Systems Analysis and Modeling

Chestnut, Harold. *Systems Engineering Tools*. New York:Wiley, 1965

De Neufville, Richard and Joseph H. Stafford. *Systems Analysis for Engineers and Managers*. New York: McGraw-Hill, 1971

Fisher, Gene H. *Cost Considerations in Systems Analysis*. New York:American Elsevier,1971

#### Space Systems Design and Operations

Space Vehicle Design, Michael D. Griffin and James R. French

Space Mission Analysis and Design (2nd ed.), Wiley J. Larson and James R. Wertz (editors)

Design of Geosynchronous Spacecraft, Brij N. Agrawal

Spacecraft Systems Engineering, Peter W. Fortescue and John P.W. Stark (editors)

Cost-Effective Space Mission Operations, Daryl Boden and Wiley J. Larson (editors)

Reducing Space Mission Cost, Wiley J. Larson and James R. Wertz (editors)

#### 4.5 Where can I go for more information?

NPR 7123.1 Systems Engineering Procedural Requirements:2006

JSC Work Instruction EA-WI-016 Off-the-Shelf Hardware Utilization in Flight Hardware Development

MSFC Work Instruction MWI 8060.1 Rev A Off-the-Shelf Hardware Utilization in Flight Hardware Development

INCOSE Systems Engineering Handbook, version 3.:2006

AIAA G-118-200X Managing the Use of Commercial Off The Shelf (COTS) Software Components for Mission-Critical Systems

#### **4.7 Verification Additional Reading Material**

The following documents offer additional reading on the subject of Verification. Most items listed were utilized in developing the content of this section:

NPR7123.1 NASA Systems Engineering Processes and Requirements

NPR7120.5 NASA Program and Project Management Processes and Requirements

EIA-632 Processes for Engineering a System, ANSI/EIA -632- 1998, 1999.

ISO/IEC 15288

IEEE Standard 1220

SMC Systems Engineering

JPR7120.3: Project Management: Systems Engineering & Project Control Processes and Requirements. NASA/JSC; March 2004.

SP6105: Systems Engineering Handbook. NASA/HQ; June 1995.

#### **4.8 Validation Additional Reading**

The following documents are additional sources of information on the topic of Product Validation, many of which were referenced in preparation of this subject matter material::

NPR7123.1 *NASA Systems Engineering Processes and Requirements*

NPR7120.5 *NASA Program and Project Management Processes and Requirements*

EIA-632 *Processes for Engineering a System*, ANSI/EIA -632- 1998, 1999.

ISO/IEC 15288

IEEE Standard 1220

SMC Systems Engineering

JPR7120.3: *Project Management: Systems Engineering & Project Control Processes and Requirements*. NASA/JSC; March 2004.

SP6105: *Systems Engineering Handbook*. NASA/HQ; June 1995.

#### **4.9 Transition**

NPR 7123.1 Systems Engineering Procedural Requirements:2006

ANSI/EIA 632:1998 *Processes for Engineering a System*; section 4.4.2

DoD Defense Acquisition Guide:2004

INCOSE Systems Engineering Handbook, version 3.:2006

ISO/IEC 15288:2002 *System Life Cycle Processes*

ISO/IEC 19760:2003 *A Guide for the application of ISO/IEC 15288*

NAVAIR Systems Command SE Guide:2003 (based on requirements of ANSI/EIA 632:1998)

### 5.1 References

Reference: AIAA Guide for Managing the Use of Commercial Off the Shelf (COTS) Software Components for Mission-Critical Systems (G-118-2006e)

AIAA Standards Series  
Published by AIAA, © 2006, 65 pages, Digital

### 5.3 References

[Kaplan] Kaplan, S., B. John Garrick, *On the Quantitative Definition of Risk*, Risk Analysis, Vol. 1, No. 1, 1981.

[NPR 7123.1] NASA Headquarters, NPR 7123.1, *NASA Systems Engineering Processes and Requirements*, NASA/HQ, 2006

[NPR 8000.4] NASA Headquarters, NPR 800.4, *Risk Management Procedures and Guidelines, w/Change 1*, Office of Safety and Mission Assurance, NASA/HQ, 2004.

[NPR 8705.5] NASA Headquarters, NPR 8705.3, *Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners*, NASA/HQ, 2002.

[NPR 8715.3] NASA Headquarters, NPR 8715.3a, *NASA General Safety Program Requirements*, NASA/HQ, 2006.

[NRC] National Research Council, *Understanding Risk: Informing Decisions in a Democratic Society*, Stern, C. and H. Fineberg, Eds., Committee on Risk Characterization, 1996.

### 5.3 Bibliography

Clemen, R., and T. Reilly, *Making Hard Decisions: with DecisionTools*, Pacific Grove, CA, 2001.

Dezfuli, H., *Role of System Safety in Risk-informed Decision Making*, The NASA Risk Management Conference 2005, Orlando, Florida, December 7, 2005.

Morgan, M. Granger, and M. Henrion, *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*, Cambridge University Press, Cambridge, U.K., 1990.

Nuclear Regulatory Commission, *White Paper on Risk-Informed and Performance-Based Regulation*, SECY-98-144, 1998.

Stamelatos, M., H. Dezfuli and G. Apostolakis, *A Proposed Risk-Informed Decision-Making Framework for NASA*, Proceedings of the 8<sup>th</sup> International Conference on Probabilistic Safety Assessment and Management, New Orleans, LA, May 14-18, 2006.

### 5.4 Where can I go for more information?

The following documents are cited to give the user additional information or reference material.

Department of Defense (DOD)

Defense Logistics Agency (DLA) Cataloging Handbook H4/H8 Series

MIL-STD-881—Work Breakdown Structure (WBS)

MIL-HDBK-965—Parts Control Program

Other Government Documents

FAR Part 46.407, “Quality Assurance, Nonconforming Supplies or Services”

14CFR1214— Part 1214—Space Flight 48CFR1814—Acquisition of Commercial Items

48CFR1814—Acquisition of Commercial Items

National Aeronautics and Space Administration

NPR 7120.5, NASA Program and Project Management Processes and Requirements

NPR 7123.1 NASA Systems Engineering Processes and Requirements

NPR 7150.2, NASA Software Engineering Requirements

NPR 1441.1, NASA Records Retention Schedules

IEEE/EIA

IEEE/EIA (ISO/IEC) 12207.1, Software Life Cycle Processes—Life Cycle Data

ASME

ASME Y14.24, Types & Applications of Engineering Drawings

ASME Y14.100, Engineering Drawing Practices

IEEE

IEEE 828, Software Configuration Management Plans

IEEE 1042, Guide to Software Configuration Management

ISO

ISO/IEC TR15846, Technical Report Information Technology—Software Life Cycle Processes Configuration Management

ISO 10007:1995(E), Quality management—Guidelines for configuration management

NDIA

ANSI-GEIA-EIA-649, National Consensus Standard for Configuration Management

GEIA-859, Data Management Standard

## **5.5 Where can I get more information?**

NPR 7120.5, NASA Program and Project Management Processes and Requirements

NPR 7123.1 NASA Systems Engineering Processes and Requirements

NPR 7150.2, NASA Software Engineering Requirements

NPR 1441.1, NASA Records Retention Schedules

NDIA, ANSI-GEIA-EIA-649, National Consensus Standard for Configuration Management

NDIA, GEIA-859, Data Management Standard for Industry & Government

## 5.7 References

Blanchard, Benjamin S., and Fabrycky, Wolter, *Systems Engineering and Analysis*, 4th Edition (Prentice Hall, 2006).

Clemen, R., and T. Reilly, *Making Hard Decisions: with DecisionTools*, Pacific Grove, CA, 2001.

Keeney, Ralph L., and McDaniels, Timothy L., "A Framework to Guide Thinking and Analysis Regarding Climate Change Policies," *Risk Analysis* **21**, No. 6, pp 989-1000 (Society for Risk Analysis, 2001).

Keeney, Ralph L., *Value-Focused Thinking: A Path to Creative Decisionmaking* (Harvard University Press, 1992).

Keeney, Ralph L., and Raiffa, Howard, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, (Cambridge University Press, 1993).

Morgan, M. Granger, and M. Henrion, *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*, Cambridge University Press, Cambridge, U.K., 1990.

Saaty, Thomas L., *The Analytic Hierarchy Process* (McGraw-Hill, 1980).

## 6.1 References/Templates

[CMMI-Acq] Hofmann, Hubert F., Kathryn M. Dodson, Gowri S. Ramani, and Deborah K. Yedlin, *Adapting CMMI for Acquisition Organizations: A Preliminary Report*, CMU/SEI-2006-SR-005, Software Engineering Institute Carnegie Mellon University, Pittsburgh, PA, June 2006, p. 338-340.

[Eslinger] Eslinger, Suellen, *Software Acquisition Best Practices for the Early Acquisition Phases*, The Aerospace Corporation, August 2004

[FAR] Federal Acquisition Regulations

[Metrics] *System and Software Metrics for Performance-Based Contracting*, Guidance\_on\_Metrics\_for\_PBC\_R1V01HB1.doc. (Note: Document was developed by NASA and therefore can be freely reused.)

[Mitre] *Common Risks and Risk Mitigation Actions for a COTS-based System*, <http://www.mitre.org/work/sepo/toolkits/risk/taxonomies/files/CommonRisksCOTS.doc>, The Mitre Corporation

[NFS] NASA FAR Supplement

[NAVAIR] Naval Air Systems Command, *Systems Engineering Guide*, May 2003

[Procurement] NASA Procurement Library, <http://ec.msfc.nasa.gov/hq/library/library.html>, Solicitation to Contract Award, <http://ec.msfc.nasa.gov/hq/library/v-Sol-Award.htm>

[S06012] NASA Office of Inspector General, *Final Memorandum on NASA's Acquisition Approach Regarding Requirements for Certain Software Engineering Tools to Support NASA Programs*, Assignment No. S06012, August 2006

[SMC] R. J. Adams, S. Eslinger, P. Hantos, K. L. Owens, L.T.Stephenson, J.M.Tagami, R. Weiskopf, (Computer Systems Division, Engineering Technology Division) and Lt. Col. G.A.Newbury and M.A.Zambrana, (Directorate of Systems Engineering, Space and Mission Systems Center, USAF), *Software Development Standard for Space Systems*, Aerospace Report No. TOR—2004(3909)-3537, Revision B, March 11, 2005

[SEB] *The SEB Source Evaluation Process*, <http://ec.msfc.nasa.gov/hq/library/SEB-cover.html>

[SOWC] *Statement of Work Checklist*, 5523-SOWReviewChecklist\_R5V0.doc (Note: Document was developed by NASA and therefore can be freely reused).

[INCOSE] *Systems Engineering Handbook: A "What To" Guide for all SE Practitioners*, INCOSE-TP-2003-016-02, Version 2a, 1 June 2004, Technical Board—International Council on Systems Engineering (INCOSE)

NASA-STD-(I)-7009, *Standard for Models and Simulations*

## 6.2 Reference

Miao, Y. and J.M. Haake (1998). Supporting Concurrent Design by Integrating Information Sharing and Activity Synchronization, in Proceedings of the 5<sup>th</sup> ISPE International Conference on Concurrent Engineering Research and Applications (CE98), p. 165-174.

## 6.3 References

Blanchard, B.S. and Fabrycky, W. J. (1990) *Systems Engineering and Analysis*. Englewood Cliffs, NJ; Prentice-Hall.

Chapanis, A. (1980) The error-provocative situation: A central measurement problem in human factors engineering. In W. E. Tarrants (Ed.), *The Measurement of Safety Performance*. New York: Garland STPM Press.

MIL-HDBK-46855A. (1996) Human Engineering Program Process and Procedures. Washington, DC: Department of Defense.

Eggemeier, F. T. and Wilson, G. F. (1991) Performance and subjective measures of workload in multitask environments: In D. Damos (Ed.), *Multiple-task Performance*. London: Taylor and Francis.

Endsley, M. R. and Rogers, M. D. (1994) Situation awareness information requirements analysis for en route air traffic control. In *Proceedings of the Human Factors and Ergonomics Society 38<sup>th</sup> Annual Meeting*. Santa Monica, CA: Human Factors and Ergonomics Society.

Fuld, R. B. (1993) The fiction of function allocation. *Ergonomics in Design*, January, 20-24.

Glass, J. T., Zaloom, V., and Gates, D. (1991) A micro-computer-aided link analysis tool. *Computers in Industry*, 16, 179-187.

Gopher, D. and Donchin, E. (1986) Workload: an examination of the concept. In K. R. Boff, L. Kaufman and J. P. Thomas (Eds.), *Handbook of Perception and Human Performance: Vol. II. Cognitive Processes and Performance*. New York: John Wiley & Sons.

Hart, S. G. and Wickens, C. D. (1990) Workload assessment and prediction. In H. R. Booyer (Ed.) *MANPRINT: An Approach to Systems Integration*. New York: Van Nostrand Reinhold.

Huey, B. M. and Wickens, C. D. (Eds.) (1993) *Workload Transition*. Washington, DC: National Academy Press.

Jones, E. R., Hennessy, R. T. and Deutsch, S. (Eds.) (1985) *Human Factors Aspects of Simulation*. Washington, DC: National Academy Press.

Kirwin, B. and Ainsworth, L. K. (1992) *A Guide to Task Analysis*. Londo: Taylor and Francis.

Kurke, M. I. (1961) Operational sequence diagrams in system design. *Human Factors*, 3, 66-73.

Meister, D. (1985) *Behavioral Analysis and Measurement Methods*. New York: John Wiley & Sons.

Price, H. E. (1985) The allocation of functions in systems. *Human Factors*, 27, 33-45.

Shafer, J. B. (1987) Practical workload assessment in the development process. In *Proceedings of the Human Factors Society 31<sup>st</sup> Annual Meeting*. Santa Monica, CA: Human Factors Society.

### **7.3 Reference**

David Meister's, *Human Factors: Theory and Practice*, 1971

(Ref.: <http://www.hf.faa.gov/webtraining/HFModel/HFInterModel/overview.htm>)

### **7.5 Reference**

National Institute of Standards and Technology (NIST) Special Publications (SP) 811, Guide for the Use of the International System of Units (SI) by Barry Taylor of the Physics Laboratory NIST Standard Reference Database 146

Inquiries or comments: Feedback

Online: May 1996—Last update: May 2007

DoD Guide for Identification and Development of Metric Standards, SD-10, Office of the Under Secretary of Defense, Acquisition, Technology, & Logistics, December 2003.

### **Appendix 4-3**

David Long's *Relationships between Common Graphical Representations in Systems Engineering.*, Vitech Corp. 2003]

**Index**  
**[to come]**