INCOSE
International Council on Systems Engineering

SYSTEMS ENGINEERING HANDBOOK

A GUIDE FOR SYSTEM LIFE CYCLE PROCESSES AND ACTIVITIES

INCOSE-TP-2003-002-03.1
INCOSE SYSTEMS ENGINEERING HANDBOOK, version 3.1
August 2007

Edited by:

Cecilia Haskins, CSEP

Revisions and Appendices D to N edited by:

Kevin Forsberg, CSEP

Michael Krueger, CSEP

This INCOSE Technical Product was prepared by the Systems Engineering Handbook Development Team of the International Council on Systems Engineering (INCOSE). It is approved by INCOSE for release as an INCOSE Technical Product.

First printing, August 2007.
Second printing with corrections, September 2008.

Preface

Objective

The INCOSE Systems Engineering Handbook, version 3 (SEHv3), represents a shift in paradigm toward global industry application consistent with the Systems Engineering Vision. The objective for this document is to provide an updated description of the key process activities performed by systems engineers. The intended audience is the new systems engineer, an engineer in another discipline who needs to perform systems engineering or an experienced systems engineer who needs a convenient reference. The primary purpose of this version 3.1 update is to make this document a stand-alone reference fully supporting the Certified Systems Engineering Professional (CSEP) examination. Eleven appendices have been added for that purpose. Appendix D explains the context for this update in more detail.

The descriptions in this handbook show what each systems engineering process activity entails, in the context of designing for affordability and performance. On some projects, a given activity may be performed very informally (e.g., on the back of an envelope, or in an engineer's notebook); on other projects, very formally, with interim products under formal configuration control. This document is not intended to advocate any level of formality as necessary or appropriate in all situations. The appropriate degree of formality in the execution of any systems engineering process activity is determined by:

a. the need for communication of what is being done (across members of a project team, across organizations, or over time to support future activities),
b. the level of uncertainty,
c. the degree of complexity, and
d. the consequences to human welfare.

On smaller projects, where the span of required communications is small (few people and short project life cycle) and the cost of rework is low, systems engineering activities can be conducted very informally (and thus at low cost). On larger programs, where the cost of failure or rework is high, increased formality can significantly help in achieving program opportunities and in mitigating program risk.

In a project environment, work necessary to accomplish project objectives is considered "in scope;" all other work is considered "out of scope." On every project, "thinking" is always "in scope." Thoughtful tailoring and intelligent application of the systems engineering process described in this handbook is essential to achieve the proper balance between the risk of missing project technical and business objectives on the one hand, and process paralysis on the other. Chapter 10 provides tailoring guidelines to help achieve that balance.

It is the intention of the SEHv3 steering committee that appendices will be developed to elaborate on significant topics, and that these appendices will be available on-line to members in the INCOSE Product Asset Library (IPAL). The addition of these on-line descriptions, work sheets, checklists, and how-to guides will evolve over time, and it is anticipated that all INCOSE members, working groups, and Corporate Advisory Board member companies will contribute to the creation of this resource. Actual content will be under the control of the IPAL working group.

Approved for SEHv3:

Terje Fossnes, CSEP, Chair, INCOSE SEHv3 Development Team
Kevin Forsberg, CSEP, Co-Chair, INCOSE SEHv3 Development Team
Eric Aslaksen, CSEP, INCOSE Associate Director, Technical Review
Samantha Brown, INCOSE Technical Director

Approved for SEHv3.1:

Kevin Forsberg, CSEP, Chair, INCOSE SEHv3.1 Development Team
Erik Aslaksen, CSEP, INCOSE Associate Director, Technical Review
Samantha Brown, INCOSE Technical Director

Acknowledgements for SEHv3.1 Contributions:

The development team for the INCOSE Systems Engineering Handbook version 3.1 owes a debt of gratitude to the people who contributed to versions 2a and 3.0. This update merges key elements of both of those activities. Everything in this update is based on those two solid platforms.

One of the constraints imposed on the development of Systems Engineering Handbook version 3.0 (SEHv3) was a page count limit of 150 pages. There were detailed sections in SEHv2a that we considered relevant, but which could not be included because of page count limitations. The intent in 2006 was that these sections would be extracted and put into an electronic file (IPAL) as a supplement to SEHv3. This revision is the initial step toward building a set of "living" appendices providing elaboration of the SE practices. We expect these appendices to evolve over time.

An important function of this handbook is to provide a body of knowledge for the INCOSE certification examination for Certified Systems Engineering Professionals (CSEP). Certification exams based on SEHv3.1 will include questions based on the information found in the appendices. As the body of knowledge expands with additional appendices and deletions of old material, exam questions will be deleted, modified, and/or added. Ample notification of such changes will be provided.

Several INCOSE working groups provided valuable contributions to this document, and we would like to specifically thank the Human Systems Integration WG, the Requirements WG, and the System Safety WG for their interest and support.

A review team, under the direction of Erik Aslaksen (CSEP), provided an invaluable assessment, resulting in 249 suggestions or comments, all of which were addressed. We would like to thank the entire team for their excellent review efforts: Dick Allen-Shalless, Eileen Arnold (CSEP), Gary Bakken, Carlos Caldeira, Mark Halverson, Jerry Huller (CSEP), Ken Kepchar (CSEP), Leonid Lev, Bernard Morais, John Muehlbauer (CSEP), Ramakrishnan Raman (CSEP), Alex Schmarr, and Mark Walker (CSEP). In addition, we would like to give especial thanks to Jerry Huller (CSEP) for his added effort in providing insightful editing on ten of the appendices. The end product is greatly improved because of his contribution.

Gratefully,

Kevin Forsberg, CSEP

Mike Krueger, CSEP

Acknowledgments for SEHv3 Contributions

The INCOSE Systems Engineering Handbook version 3 development team owes a debt of gratitude to all the contributors to prior editions (versions 1, 2, and 2a). The framework they provided gave a solid basis for moving ahead with this version. However this present document represents a significant departure from its predecessors since the goal was to create a shorter document consistent with the international standard ISO/IEC 15288:2002(E) – *Systems engineering – system life cycle processes*. As a result, we will not list all the contributors to earlier versions; interested readers are referred to the acknowledgment pages in those documents.

We want to thank the two co-chairs who worked with us in the formative stages of this handbook: John Leonard and Jim Chism. They provided valuable guidance and leadership in the difficult transition from a handbook based on the earlier versions to one based on ISO/IEC 15288.

It would be difficult to accurately characterize the specific contributions of each of the volunteers – section leads, steering committee, authors, and reviewers. Many served in multiple roles. A great deal of effort and enthusiasm was provided by the section leads and key authors, most of who also served on the steering committee. We acknowledge them in alphabetical order: Karen Bausman, Joe Carl, Sandy Friedenthal, Karl Geist, Ken Kepchar, Mike Krueger, Harold Kurstedt, Sean O'Neill, Mike Persson, Mary Redshaw, Andy Schuster, L. Mark Walker, and Jim Whalen. The steering committee also included the following people: Howard Eisner, Gerard Fisher, Richard Kitterman, David Long, William Mackey, and Paul Robitaille.

The review team lead by Erik Aslaksen included in alphabetical order: Jonas Andersson, Lily Birmingham, Samantha Brown, John Clark, Michael Eagan, Ayman El-Fatatry, Patrick Hale, Jorg Lalk, Harold Lawson, Virginia Lentz, William Miller, Juan Miro, John Muehlbauer, Robert Peffer, Robert Porto, John Quitter, Tom Strandberg, Dan Surber, and David Walden. In addition, representatives from the INCOSE Hampton Roads Area Chapter, the Swedish chapter, the Requirements Working Group, and the AIAA Technical Committee on Systems Engineering provided comments.

One of the requirements for this handbook is that it looks and reads as if it were written by a single person, and the reviewers all felt this objective has been met successfully. The co-chairs wish to thank our editor, Cecilia Haskins, for her dedication, and contributions to achieve this result.

We apologize in advance if we omitted anyone from this list in the final minutes before going to production.

Gratefully,

Terje Fossnes and Kevin Forsberg

Table of Contents

List of Figures and Tables

APPENDICES

THIS PAGE
INTENTIONALLY
LEFT BLANK

1 Systems Engineering Handbook Scope

1.1 Purpose

This handbook defines the discipline and practice of systems engineering (SE) for student and practicing professional alike. This handbook provides an authoritative reference to understand the discipline in terms of content and practice.

1.2 Application

This handbook is consistent with ISO/IEC 15288: 2002(E) – *Systems engineering – System life cycle processes* (hereafter referred to as ISO/IEC 15288) to ensure its usefulness across a wide range of application domains – man-made systems and products, as well as business and services.

ISO/IEC 15288 is an international standard that is a generic process description, whereas this handbook further elaborates the processes and activities to execute the processes. Before applying this handbook in a given organization or project, it is recommended that the tailoring guidelines in Chapter 10 be used to remove conflicts with existing policies, procedures and standards already in use. Processes and activities in this handbook do not supercede any international, national, or local laws or regulations.

For organizations including much of commercial industry that does not follow the principles of ISO/IEC 15288 to specify their life cycle processes, this handbook can serve as a reference to practices and methods which have proven beneficial to the systems engineering community at large and which can add significant value in new domains if appropriately selected and applied.

1.3 Contents

This chapter defines the purpose and scope of this handbook. Chapter 2 provides an overview of the goals and value of using systems engineering throughout the systems life cycle. Chapter 3 describes an informative life cycle model with six stages: Concept, Development, Production, Utilization, Support, and Retirement.

ISO/IEC 15288 identifies four process groups to support systems engineering. Each of these process groups is the subject of a chapter. A graphical overview of these processes is given in Figure 1-1.

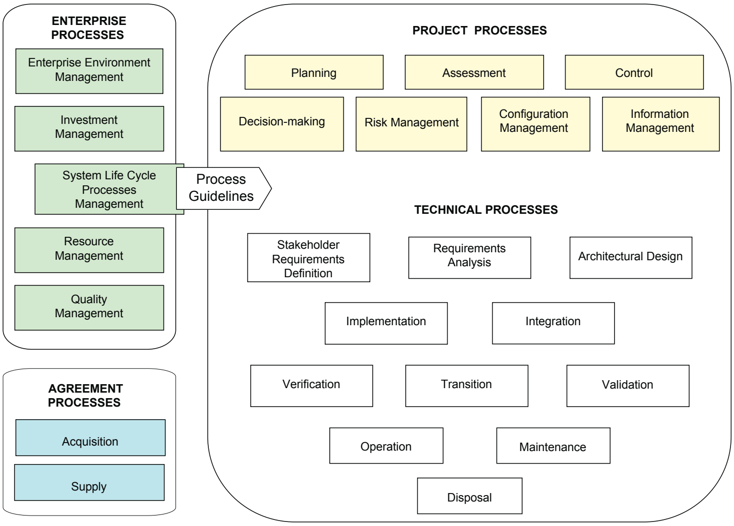


Figure 1-1 System Life Cycle Processes Overview per ISO/IEC 15288

Technical Processes (Chapter 4) include stakeholder requirements definition, requirements analysis, architectural design, implementation, integration, verification, transition, validation, operation, maintenance, and disposal.

Project Processes (Chapter 5) include planning, assessment, control, decision-making, risk management, configuration management, and information management.

Enterprise Processes (Chapter 6) include enterprise management, investment management, system life cycle processes management, resource management, and quality management**.** As Figure 1-1 illustrates, the outputs of the system life cycle processes management process directs the tailoring of the Technical and Project processes.

Agreement Processes (included in Chapter 6) address acquisition and supply.

Activities that support these processes are further described in Chapters 7-9 – see Table 1-1 for an overview of topics.

Enabling Systems Engineering Activities (Chapter 7) elaborates on requirements management, risk and opportunity management, and decision-making.

System Life Cycle Processes Activities (Chapter 8) discusses selected topics within acquisition and supply, architectural design, configuration management, information management, investment management, project planning, quality management, resource management, validation, and verification.

Specialty Engineering Activities (Chapter 9) contains practical information about topics such as acquisition logistics and human factors engineering**.**

Appendix A contains an n-squared analysis of the processes showing where dependencies exist in the form of shared inputs or outputs. Agreement processes are not included because their interaction with the other processes is most subject to enterprise tailoring. Appendices B and C provide a glossary of terms and acronyms. Appendix D gives the context for and list of the remaining 10 appendices that have been added to this document as the major change introduced by revision 3.1. Errors, omissions and other suggestions for this handbook can be submitted to INCOSE using the User Feedback Form in Appendix O at the end of this document.

> Not every process will apply universally. Careful selection from the material that follows is recommended. Reliance on process over progress will not deliver a system. If you are not familiar with tailoring concepts, please read Chapter 10 before using this handbook

Table 1-1 Systems Engineering Process Activities Overview

	Systems Engineering Process Activity	Focus	When it is most Useful
7.0	***Enabling System Engineering***	-	-
7.1	Decision Management	Trade studies and project reviews	Through Life
7.2	Requirements Management	System requirements	Through Life
7.3	Risk and Opportunity Management	Recognizing opportunities and risks	Through Life
8.0	***Systems Engineering Support***	-	-
8.1	Acquisition and Supply	Procurement business relationships	Through Life
8.2	Architectural Design	Technical analysis	Development Stage
8.3	Configuration Management	Control of changes through life	Through Life
8.4	Information Management	Project archives and info exchange	Through Life
8.5	Investment Management	Estimation and analysis of costs	Through Life
8.6	Project Planning	Managing technical activities	Through Life
8.7	Quality Management	Product and process assessment	Through Life
8.8	Resource Management	Skills and resource availability	Development Stage
8.9	Validation	User concurrence – correct system	Through Life
8.10	Verification	Requirements met – system correct	Through Life
9.0	***Specialty Engineering Activities***	-	-
9.1	Design for Acquisition Logistics	Integrated logistics support solutions	Development Stage
9.2	Electromagnetic Compatibility	Electro-magnetic protections	Development Stage
9.3	Environmental Impacts	Care for the biosphere and humans	Development Stage
9.4	Human Factors	Human capabilities and well-being	Development Stage
9.5	Mass Properties	Physical characteristics of the system	Development Stage
9.6	Modeling, Simulation, & Prototype	Early validation and testing	Development Stage
9.7	Safety/Health Hazards	Minimum risk to users	Through Life
9.8	Sustainment Engineering	Continued use of system	Through Life
9.9	Training Need Analysis	Basis for training requirements	Development Stage

1.4 Format

A common format has been applied in Chapters 4 through 6 to the elaboration of the system life cycle processes found in ISO/IEC 15288. Each process is illustrated by a context diagram. A sample is shown in Figure 1-2. To understand a given process, the reader is encouraged to find the complete information in the combination of diagrams and text. The following heading structure provides consistency in the discussion of these processes.

- Purpose
- Description
- Inputs – this section discusses all inputs, including Controls and Enablers
- Outputs
- Process Activities
- Common approaches and tips – endnotes contain additional readings

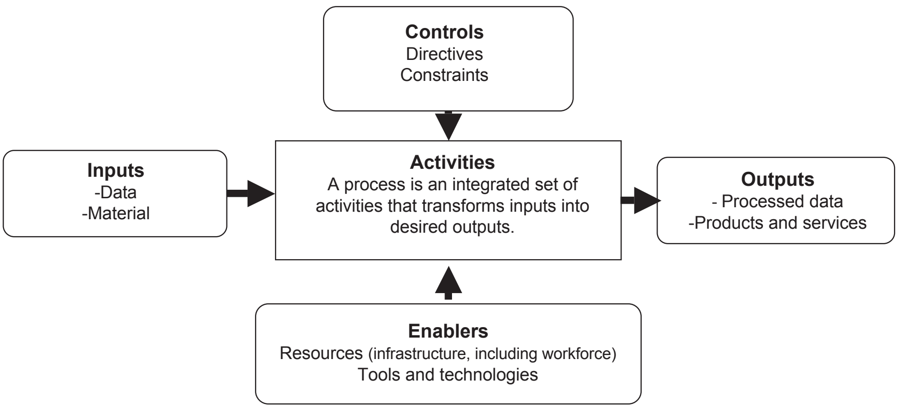


Figure 1-2 Sample of Context Diagram for Process

1.5 Definitions of frequently used terms

This section contains the definition of some terms used frequently throughout this handbook. Definitions in italics have been taken from ISO/IEC 15288. A full glossary of definitions is found in Appendix B. An elaboration of the system hierarchy is given in Appendix E.

Term	Definition
Activity	*set of actions that consume time and resources and whose performance is necessary to achieve or contribute to the realization of one or more outcomes*
Enabling system	*a system that complements a system-of-interest during its life cycle stages but does not necessarily contribute directly to its function during operation*

Term	Definition
Enterprise	*that part of an organization with responsibility to acquire and to supply products and/or services according to agreements*
Organization	*a group of people and facilities with an arrangement of responsibilities, authorities and relationships*
Process	*set of interrelated or interacting activities that transform inputs into outputs*
Project	*an endeavor with start and finish dates undertaken to create a product or service in accordance with specified resources and requirements*
Stage	*a period within the life cycle of a system that relates to the state of the system description or the system itself*
System	*a combination of interacting elements organized to achieve one more stated purposes*
System element	*a member of a set of elements that constitutes a system*
System-of-interest	*the system whose life cycle is under consideration*
Systems Engineering	Systems Engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem. Systems Engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs. (INCOSE)

1.6 References

The following documents have been used to establish the framework and practical foundations for this handbook.

- *SE Guidebook for ITS,* California Department of Transportation, Division of Research and Innovation, v 2.0, Jan 2007
- INCOSE, *Systems Engineering Handbook*, version 2a, June, 2004.
- ISO/IEC 15288: 2002(E), *Systems engineering – System life cycle processes*, Geneva: International Organization for Standardization, issued 1 November 2002.
- ISO/IEC TR 19760: 2003(E), *Systems Engineering – A guide for the application of ISO/IEC 15288,* Geneva: International Organization for Standardization, issued 15 November 2003.

THIS PAGE
INTENTIONALLY
LEFT BLANK

2 Systems Engineering Overview

2.1 Introduction

This chapter offers a brief overview of the discipline of Systems Engineering, beginning with some definitions, an abbreviated survey of the origins of the discipline and discussions of the value of applying systems engineering. Systems are pervasive in our daily life. They are tangible in that they exist in the products we use, the technologies we employ, the services we procure, and in the fabric of society.

2.2 Definition of systems engineering

Systems engineering is a profession, a process, and a perspective as illustrated by these three representative definitions.

> Systems engineering is a discipline that concentrates on the design and application of the whole (system) as distinct from the parts. It involves looking at a problem in its entirety, taking into account all the facets and all the variables and relating the social to the technical aspect. (Ramo [1])

> Systems engineering is an iterative process of top-down synthesis, development, and operation of a real-world system that satisfies, in a near optimal manner, the full range of requirements for the system. (Eisner [2])

> Systems engineering is an interdisciplinary approach and means to enable the realization of successful systems. (INCOSE [3])

Certain keywords emerge from this sampling – interdisciplinary, iterative, socio-technical, and wholeness.

The systems engineering perspective is based on systems thinking. Systems' thinking occurs through discovery, learning, diagnosis, and dialog that lead to sensing, modeling, and talking about the real-world to better understand, define, and work with systems. Systems thinking is a unique perspective on reality — a perspective that sharpens our awareness of wholes and how the parts within those wholes interrelate. A systems thinker knows how systems fit into the larger context of day-to-day life, how they behave, and how to manage them. Systems thinking recognizes circular causation, where a variable is both the cause and the effect of another and recognizes the primacy of interrelationships and non-linear and organic thinking — a way of thinking where the primacy of the whole is acknowledged.

The systems engineering process has an iterative nature that supports learning and continuous improvement. As the processes unfold, systems engineers uncover the real requirements and the emergent properties of the system. Complexity can lead to unexpected and unpredictable behavior of systems, hence, one of the objectives is to minimize undesirable consequences. This can be accomplished through the inclusion of and contributions from experts across relevant disciplines coordinated by the systems engineer.

Since systems engineering has a horizontal orientation, the discipline (profession) includes both technical and management processes. Both processes depend upon good decision making. Decisions made early in the life cycle of a system, whose consequences are not clearly understood, can have enormous implications later in the life of a system. It is the task of the systems engineer to explore these issues and make the critical decisions in a timely manner. The role of the systems engineer is varied and Sheard[4] is one source for a description of these variations.

2.3 Origins of systems engineering

The modern origins of systems engineering can be traced to the 1930's followed quickly by other programs and supporters.[5] Table 2-1 offers a thumbnail of some important highlights in the origins and history of the application of systems engineering.

Table 2-1 Important Dates in the Origins of Systems Engineering as a Discipline

1829	Rocket locomotive; progenitor of main-line railway motive power
1937	British multi-disciplinary team to analyze the air defense system
1939-1945	Bell Labs supported NIKE development
1951-1980	SAGE Air Defense System defined and managed by MIT
1956	Invention of systems analysis by RAND Corporation
1962	Publication of *A Methodology for Systems Engineering*
1969	Jay Forrester (Modeling Urban Systems at MIT)
1994	Perry Memorandum urges military contractors to adopt commercial practices such as IEEE P1220
2002	Release of ISO/IEC 15288

With the introduction of the international standard ISO/IEC 15288 in 2002, the discipline of systems engineering was formally recognized as a preferred mechanism to establish agreement for the creation of products and services to be traded between two enterprises – the acquirer and supplier. But even this simple designation is often confused in a web of contractors and subcontractors as the context of most systems today is as a part of a "system of systems."

2.4 Systems of systems

"Systems-of-Systems" (SoS) are defined as an interoperating collection of component systems that produce results unachievable by the individual systems alone.[6] The systems considered in ISO/IEC 15288

> *are man-made, created and utilized to provide services in defined environments for the benefit of users and other stakeholders. These systems may*

be configured with one or more of the following: hardware, software, humans, processes (e.g., review process), procedures (e.g., operator instructions), facilities, and naturally occurring entities (e.g., water, organisms, minerals). In practice, they are thought of as products or services. The perception and definition of a particular system, its architecture and its system elements depend on an observer's interests and responsibilities. One person's system-of-interest can be viewed as a system element in another person's system-of-interest. Conversely, it can be viewed as being part of the environment of operation for another person's system-of-interest.[7]

Figure 2-1 illustrates these concepts. The Global Positioning System (GPS), which is an integral part of the navigation system on board an aircraft, is a system in its own right rivaling the complexity of the air transportation system. Another characteristic of SoS is that the component systems may be part of other unrelated systems. For instance, the GPS may be an integral part of automobile navigation systems.

The following challenges all influence the development of systems of systems:

1. **System elements operate independently.** Each system in a system of systems is likely to be operational in its own right.

2. **System elements have different life cycles.** SoS involves more than one system element. Some of the system elements are possibly in their development life cycle while others are already deployed as operational. In extreme cases, older systems elements in SoS might be scheduled for disposal before newer system elements are deployed.

3. **The initial requirements are likely to be ambiguous.** The requirements for a system of systems can be very explicit for deployed system elements. But for system elements that are still in the design stage, the requirements are usually no more explicit than the system element requirements. Requirements for SoS mature as the system elements mature.

4. **Complexity is a major issue.** As system elements are added, the complexity of system interaction grows in a non-linear fashion. Furthermore, conflicting or missing interface standards can make it hard to define data exchanges across system element interfaces.

5. **Management can overshadow engineering.** Since each system element has its own product/project office, the coordination of requirements, budget constraints, schedules, interfaces, and technology upgrades further complicate the development of SoS.

6. **Fuzzy boundaries cause confusion.** Unless someone defines and controls the scope of a SoS and manages the boundaries of system elements, no one controls the definition of the external interfaces.

7. **SoS engineering is never finished.** Even after all system elements of a SoS are deployed, product/project management must continue to account for changes in the various system element life cycles, such as new technologies that impact one or more system elements, and normal system replacement due to pre-planned product improvement.

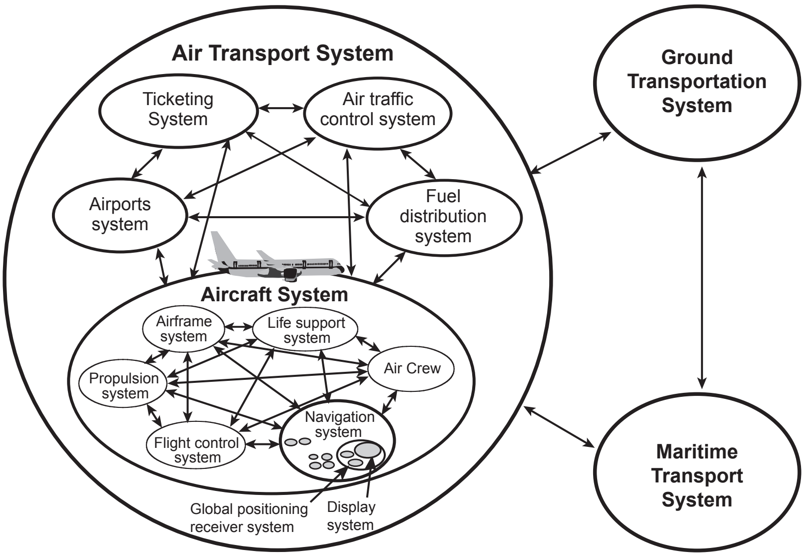


Figure 2-1 Example of the multitude of perceivable systems-of-interest in an aircraft and its environment of operation within a Transport System-of-Systems[8]

A Camera as an Example of a System-of-Systems – Not all SoS involve an environment as complex as the air transportation system. A digital camera may seem simple, but it is a system of systems with rigidly controlled interfaces. Multiple camera bodies, from simple fixed focus digital cameras to sophisticated single lens reflex cameras have a common interface to digital memory cards. The full single-lens reflex camera system has many different models of camera bodies which interface with 50 or more lens systems and multiple flash units. To be a commercial success these simple to sophisticated camera systems are designed to conform to external interfaces for standard commercial batteries, compact flash memory cards, interface cables, computers, and printer software as illustrated in Figure 2-2. In the context of SoS, systems are enclosed in the white boxes, system elements are displayed in the gray area.

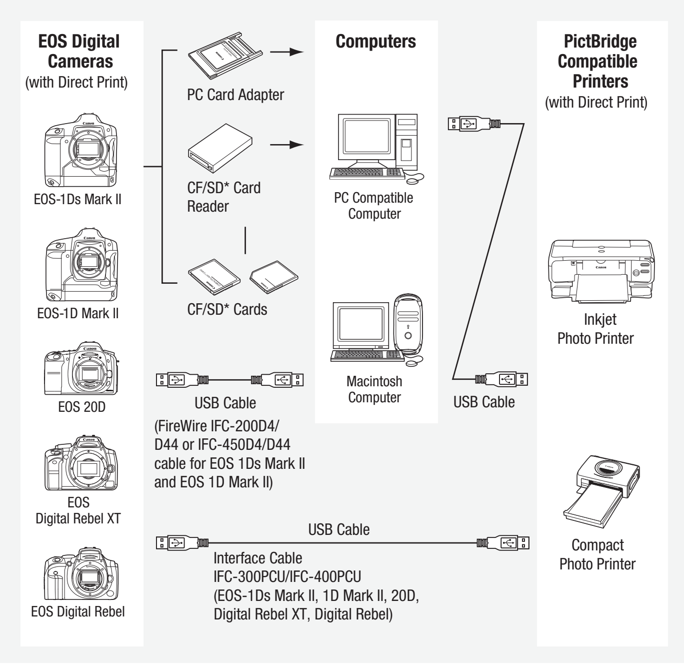


Figure 2-2 Digital Camera and Printer Systems of Systems[9]

Part of the systems engineer's job in the SoS environment is to be aware of and mitigate the risk of each of these seven challenges. Focus is placed on controlling the interfaces between system elements and external systems. It is especially important to ensure that the interfaces are still operational when an older component system is replaced with a newer version. Verification and validation processes play a critical role in such transitions.

2.5 Use of systems engineering

As can be readily inferred from the nature of the earliest projects, the systems engineering discipline emerged as an effective way to manage complexity and change. Both complexity and change have escalated in our products, services, and society. Reducing the risk associated with new systems or modifications to complex systems continues to be a primary goal of the systems engineer. This is illustrated in Figure 2-3. The percentages along the time line represent the actual life cycle cost (LCC) accrued over time – which means that the concept phase of a new system averages 8% of the total LCC. The curve for committed costs indicates the amount of LCC committed by the decisions taken. The curve indicates that when 20% of the actual cost has been accrued, 80% of the total LCC has already been determined – based on a statistical analysis performed on projects in the US DoD as reported by the Defense Acquisition University. The light arrow under the curve reminds us that errors are less expensive to remove early in the lifecycle.

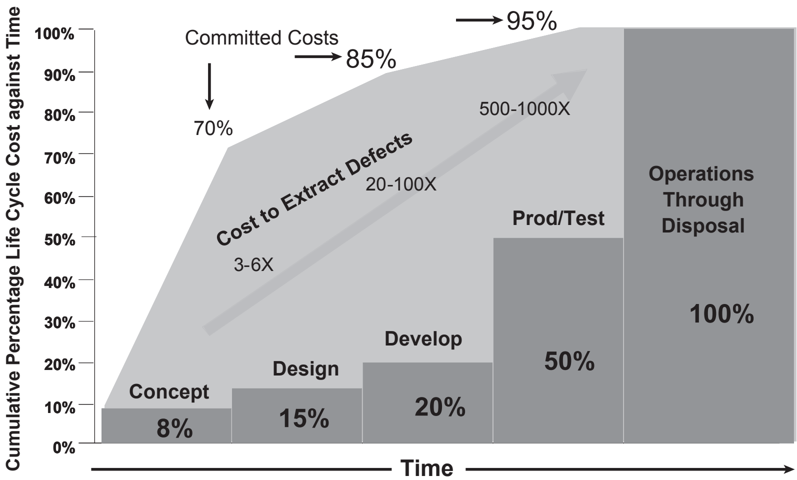


Figure 2-3 Committed Life Cycle Cost against Time [10]

This figure also demonstrates the consequences of taking early decisions without the benefit of good information and analysis. Systems engineering extends the effort performed in concept exploration and design to exceed the percentages shown in the cumulative effort step-curve and reduce the risk of hasty commitments without adequate study. The recursive nature of modern development means that the execution of the various life cycle phases is not linear as illustrated – but the consequence of ill-formed decisions is the same.

Another factor driving the need for systems engineering is that the time from prototype to significant market penetration of a new product has dropped by more than a factor of four in the past 50 years (Figure 2-4). Complexity has an impact on innovation. Few new products represent the big-bang introduction of new invention – most products and services are the result of incremental improvement. This means that the life cycle of today's products and services is longer and subject to increasing uncertainty. A well-defined systems engineering process becomes critical to establishing and maintaining a competitive edge in the 21st century.

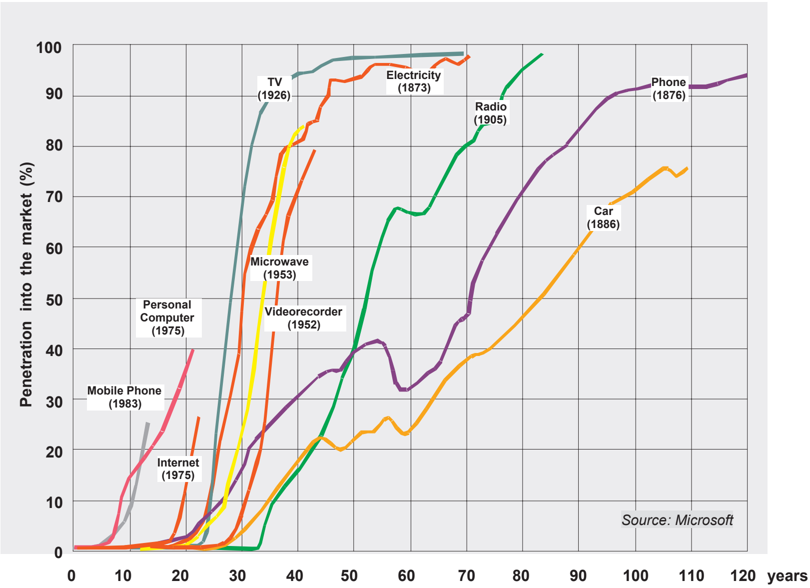


Figure 2-4 In the last century, the time from prototype to significant market penetration is dramatically reduced [11]

2.6 Value of systems engineering

A study researching the return on investment from using systems engineering was conducted by the INCOSE Systems Engineering Center of Excellence beginning in 2001. The results uncovered an inverse correlation between cost and schedule overruns and the amount of systems engineering effort (SEE). As illustrated in the graph to the left in Figure 2-5, cost overrun lessens with increasing SEE and appears to minimize at something greater than 10% SEE. Variance in the cost overrun also lessens with increasing SEE. At low SEE, a project has difficulty predicting its overrun, which may be between 0% (actual = planned) and 200% (actual = 3 x planned). At 12% SEE, the project cost is more predictable, falling between minus 20% (actual = 0.80 x planned) and 41% (actual = 1.41 x planned). The dashed lines are the 90th percentile when assuming a normal distribution.

Schedule overrun on the reported projects is illustrated in the right-hand graph in Figure 2-5. Two effects are apparent:

- Schedule overrun lessens with increasing SEE. Overrun appears to minimize at something greater than 10% SEE, although few data points exist to support a reliable calculation. The solid line is the least-squares trend line for a second order curve.

- Variance in the schedule overrun also lessens with increasing SEE. At low SEE, a project has difficulty predicting its overrun, which may be between minus 35% (actual = 0.65 x planned) and 300% (actual = 4 x planned). At 12% SEE, the project cost is more predictable, falling between minus 22% (actual = 0.78 x planned) and 22% (actual = 1.22 x planned). The dashed lines are the 90th percentile when assuming a normal distribution.

Additional work is underway to collect more data about the value of applying systems engineering to a project. These initial results indicate that systems engineering effort can be a positive factor in controlling cost overruns and reducing the uncertainty of project execution.

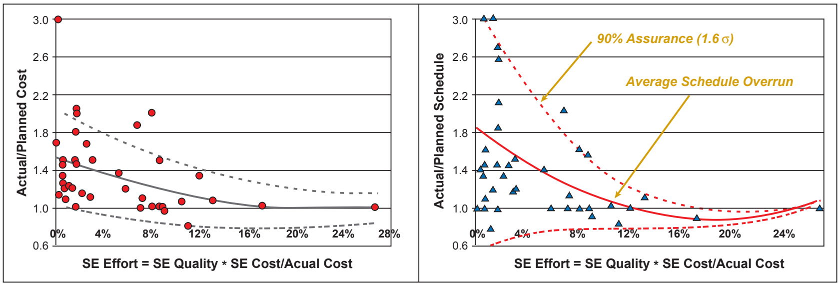


Figure 2-5 Cost and schedule overruns correlated with systems engineering effort [12]

An Allegorical Tale

Upon casual reading, systems engineers appear to be responsible for everything that happens on a project and systems engineering appears to introduce excessive process overhead and non-value added activities. A senior systems engineer at a major US company visited all of the divisions with the goal of increasing the use of good system engineering practices. His message included all the things that SE can/should do in commercializing products. His message also included a strong bias towards planning and documentation. Over a period of months he visited with Division Managers, Chief Engineers, Program Managers and Senior Engineers. He returned completely depleted of his enthusiasm. The problem was that the message was totally rejected because it either looked like useless work or way beyond anything they could afford to do from a time and dollars perspective. Some time later another senior systems engineer visited many of the same people with the same purpose but a different message. The message this engineer delivered was that big gains could be made by focusing on the most important customer needs and using a select group of synergistic system engineering tools/practices. This time the message was well received.

The lesson: "Systems engineering is a multi-disciplinary effort that involves both the technical effort and technical project management aspects of a project. Enterprises seeking to incorporate the benefits of processes outlined in ISO 15288 will remember that application of those processes, and the enablers discussed in this handbook, requires vision and practical application of the principles."[13]

1 Federal Aviation Agency (USA FAA) *Systems Engineering Manual,* definition contributed by Simon Ramo

2 Eisner, Howard, *Essentials of Project and Systems Engineering Management*

3 INCOSE, *Systems Engineering Handbook*, version 2a, June, 2004, page 11

4 Sheard, Sarah, "Twelve Roles of SE" *Proceedings of the 6th Annual INCOSE International Symposium*, 1996.

5 Hughes, Thomas P., *Rescuing Prometheus*, Chapter 4, pp. 141-195, Pantheon Books, New York, 1998

6 Krygiel, Annette J. Behind the Wizard's Curtain, CCRP Publication Series, July, 1999, p 33

7 ISO/IEC 15288, page 52

8 ISO/IEC 15288, page 53

9 Canon EOS Digital Camera Brochure

10 Defense Acquisition University, 1993

11 Microsoft

12 Honour, Eric, (2004), Understanding the Value of Systems Engineering, *Proceedings of the 14th Annual INCOSE International Symposium*, 1996, available online from the INCOSE Systems Engineering Center of Excellence (SECOE), http://www.incose.org/secoe.

13 Submitted by handbook review team

THIS PAGE
INTENTIONALLY
LEFT BLANK

3 Generic Life Cycle Stages

3.1 Introduction

Every man-made system has a life cycle, even if it is not formally defined. In keeping with increased awareness of environmental issues, the life cycle for any system-of-interest must encompass not only the development, production, and usage stages but also provide early focus on the retirement stage when decommissioning and disposal of the system will occur.

The role of the systems engineer encompasses the entire life cycle for the system-of-interest. The systems engineer works closely with the project manager in tailoring the generic life cycle, including the decision gates, to meet the needs of their specific project.

Per ISO/IEC 15288: *"6.3 – The purpose and outcomes shall be defined for each stage of the life cycle. The life cycle processes and activities are selected, tailored as appropriate, and employed in a stage to fulfill the purpose and outcomes of that stage."*

The purpose in defining the system life cycle is to establish a framework for meeting the stakeholders' needs in an orderly and efficient manner. This is usually done by defining life cycle stages, and using decision gates to determine readiness to move from one stage to the next. Skipping phases and eliminating "time consuming" decision gates can greatly increase the risks (cost and schedule), and may adversely affect the technical development as well by reducing the level of systems engineering effort as discussed in Section 2.6.

Systems engineers orchestrate the development of a solution from requirements determination through operations and system retirement by assuring that domain experts are properly involved, that all advantageous opportunities are pursued, and that all significant risks are identified and mitigated.

Systems engineering tasks are usually concentrated at the beginning of the life cycle, but both commercial and government organizations recognize the need for systems engineering throughout the systems life span, often to modify or change a system product or service after it enters production or is placed in operation.

3.2 Life Cycle Characteristics

3.2.1 Three Aspects of the Life Cycle

Every system or product life cycle consists of the business aspect (business case), the budget aspect (funding), and the technical aspect (product). The systems engineer creates technical solutions that are consistent with the business case and the funding constraints. System integrity requires that these three aspects are in balance and given equal emphasis at all decision gate reviews. For example, when the Iridium project started in the late 1980s the concept of satellite-based mobile phones was a breakthrough, and would clearly capture a significant market share. Over the next dozen years, the technical reviews ensured a highly successful technical solution. In fact, in the first decade of the 21st century, the Iridium project is proving to be a good business venture for all except for the original team who had to sell all the assets—at about two percent of their investment—through the bankruptcy court. The original team lost sight of the competition and changing consumer patterns that substantially altered the original business case. Figure 3-1 highlights two critical parameters that engineers sometimes lose sight of: time to break even (indicated by red circle) and Return on Investment (ROI; indicated by green line (lower curve)).

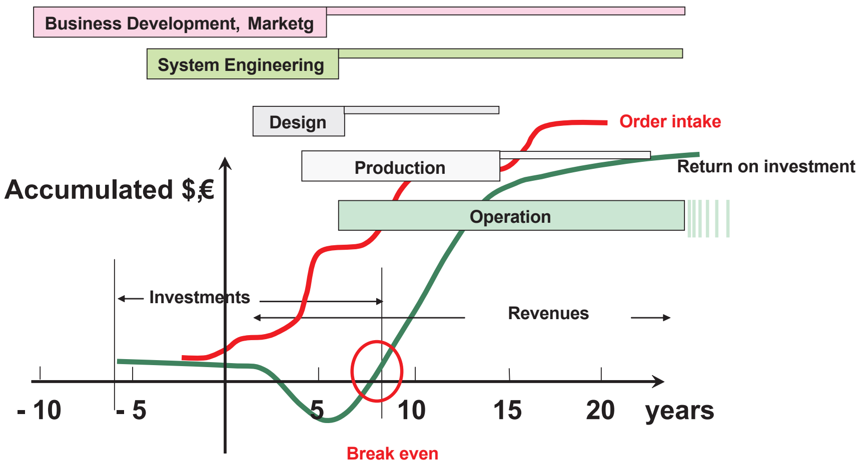


Figure 3-1 Generic Business Life Cycle [1]

3.2.2 Decision Gates

Decision gates, also known as control gates, are often called "Milestones" or "Reviews." All decision gates are both reviews and milestones; however, not all reviews and milestones are decision gates. Decision gates address the following questions:

- Does the project deliverable still satisfy the business case?
- Is it affordable?
- Can it be delivered when needed?

Decision gates represent major decision points in the system life cycle. They ensure that new activities are not pursued until the previously scheduled activities, on which new ones depend, are satisfactorily completed and placed under configuration control.

The primary objectives of decision gates are to:

- Ensure that the elaboration of the business and technical baselines are acceptable and will lead to satisfactory verification and validation.
- Ensure that the risk of proceeding to the next step is acceptable.
- Continue to foster buyer and seller teamwork.

Decision gate approval follows review by qualified experts and involved stakeholders and is based on hard evidence of compliance to the criteria of the review. Detailed information about the decision gate activity is provided in chapter 7.1.

There are at least two decision gates in any project: authority to proceed and final acceptance of the project deliverable. The project team needs to decide which life cycle stages are appropriate for their project and which decision gates beyond the basic two are needed. Each decision must have a beneficial purpose; "pro-forma" reviews waste everyone's time. Even in agile development frequent interaction with the customer may minimize, but not eliminate, the need for decision gates. The consequences of conducting a superficial review, omitting a critical discipline, or skipping a decision gate altogether are usually long-term and costly.

3.3 Life Cycle Stages

ISO/IEC 15288 states: *"6.2 - A life cycle model that is composed of stages shall be established. The life cycle model comprises one or more stage models, as needed. It is assembled as a sequence of stages that may overlap and/or iterate, as appropriate for the scope, magnitude, and complexity, changing needs and opportunities."*

Table 3-1 lists the six life cycle stages that are identified in ISO/IEC 15288. The purpose of each is briefly identified in the table, and the options from decision gates events are indicated. Note that stages can overlap, and the utilization and support stages run in parallel. Note also that the outcome possibilities for decision gates are the same for all decision gates, from the first in the concept review to the last one in the retirement stage. Subsequent chapters of this handbook will define processes and activities to meet the objectives of these lifecycle stages.

Table 3-1 Life cycle stages, their purposes, and decision gate options[2]

LIFE CYCLE STAGES	PURPOSE	DECISION GATES
CONCEPT	*Identify stakeholders' needs* *Explore concepts* *Propose viable solutions*	*Decision Options* *– Execute next stage* *– Continue this stage* *– Go to a preceding stage* *– Hold project activity* *– Terminate project*
DEVELOPMENT	*Refine system requirements* *Create solution description* *Build system* *Verify and validate system*	
PRODUCTION	Produce systems Inspect and test [verify]	
UTILIZATION	*Operate system to satisfy users' needs*	
SUPPORT	*Provide sustained system capability*	
RETIREMENT	*Store, archive, or dispose of the system*	

Figure 3-2 compares the life cycle stages of the ISO/IEC 15288 to other life cycle viewpoints. For example, the Concept Stage is aligned with the commercial project's Study Period; and with the Pre-systems Acquisition and the Project Planning Period in the US Departments of Defense and Energy, respectively. Typical decision gates are presented in the bottom line. Various life cycle models such as the waterfall, spiral, Vee, and agile development models are useful in defining the start, stop, and activities appropriate to life cycle stages.

The Vee model is used to visualize the system engineering focus, particularly during the concept and development stages. The Vee highlights the need to define verification plans during requirements development, the need for continuous validation with the stakeholders, and the importance of continuous risk and opportunity assessment.

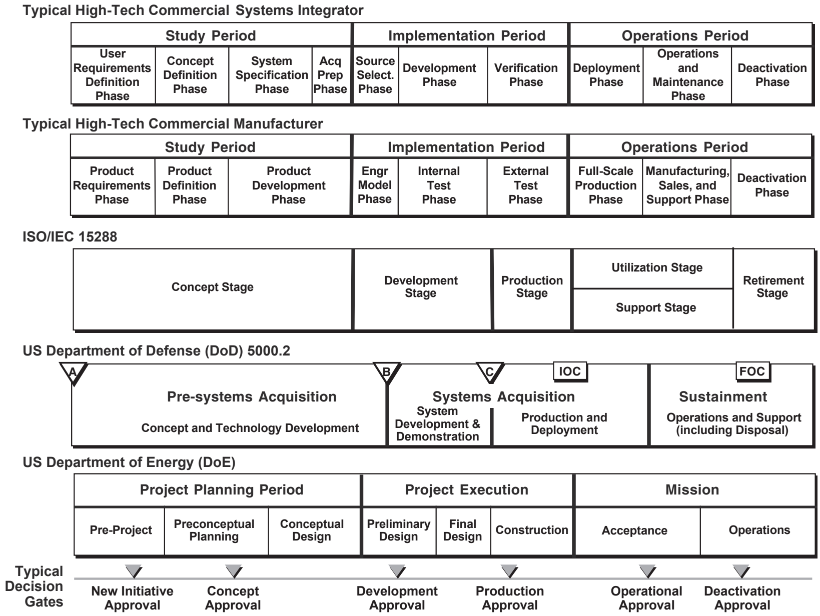


Figure 3-2 Comparison of life cycles[3]

The Vee model provides a useful illustration of the systems engineering activities during the life cycle stages. In the Vee model, time and system maturity proceed from left to right. The core of the Vee depicts the evolving baseline from user requirements agreement to identification of a system concept to definition of systems components that will comprise the final product. With time moving to the right and with the system maturity shown vertically, the evolving baseline defines the left side of the core of the Vee, as shown in figure 3-3. As entities are constructed, verified and integrated, the right side of the core of the Vee is executed.

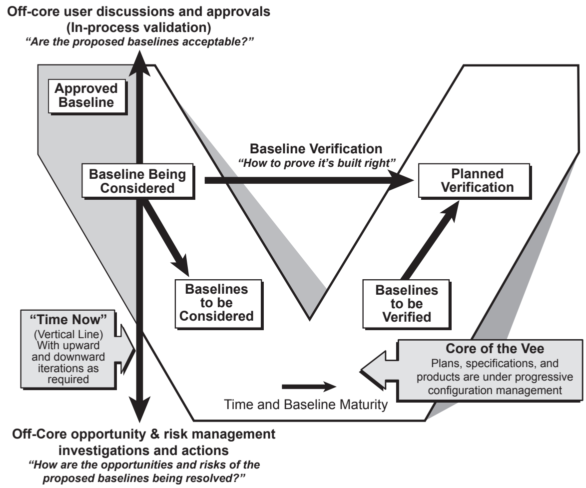


Figure 3-3 Left side of the Vee model [4]

Since one can never go backward in time, all iterations in the Vee are performed on the vertical "time now" line. Upward iterations involve the stakeholders and are the in-process validation activities that assure that the proposed baselines are acceptable. The downward vertical iterations are the essential off-core opportunity and risk management investigations and actions.

In each stage of the system life cycle the systems engineering process iterates to ensure that a concept or design is feasible, and that the stakeholders remain supportive of the solution as it evolves. In the following paragraphs, the text in italics has been taken from ISO/IEC 15288, Appendix B, where detailed activities and outcomes for each life cycle stage are itemized.

3.3.1 Pre-Concept Exploratory Research Stage

The Pre-Concept Exploratory Research Stage is sometimes referred to as the User Requirements Definition Phase. In many industries, it is common for research studies to lead to new ideas or enabling capabilities which then mature into the initiation of a new project (system-of-interest). A great deal of creative systems engineering is done in this exploratory stage, and the systems engineer leading these studies is likely to follow a new idea into the Concept Stage, perhaps as project champion. Often the Pre-Concept activities identify the enabling technologies. As discussed in chapter 2, if the work is done properly in early stages of the life cycle, it is possible to avoid recalls, and rework in later stages.

3.3.2 Concept Stage

Purpose: *The Concept Stage is executed to assess new business opportunities and to develop preliminary system requirements and a feasible design solution.*

This stage is a refinement and broadening of the studies, experiments, and engineering models pursued during the Pre-Concept Stage. The processes described in this handbook are requirements-driven, as opposed to product driven. Thus, the first step is to identify, clarify, and document stakeholders' requirements. If there was no Pre-Concept stage, that effort is done here.

During the Concept Stage, the team begins in-depth studies that evaluate multiple candidate concepts and eventually provide a substantiated justification for the system concept that is selected. As part of this evaluation mockups may be built (for hardware) or coded (for software), engineering models and simulations may be executed, and prototypes of critical components may be built and tested. Prototypes are helpful to verify the feasibility of concepts and to explore risks and opportunities. These studies expand the risk and opportunity evaluation to include affordability assessment, environmental impact, failure modes, hazard analysis, technical obsolescence, and system disposal. Key objectives are to provide confidence that the business case is sound and the proposed solutions are achievable. The systems engineer coordinates the activities of engineers from many disciplines.

The Concept Stage includes system, element, and key subsystem-level concept and architecture definition, and integration, verification and validation (IV&V) planning. The following appendices elaborate on key activities that are initiated in this stage: Appendix G on planning, Appendix I on requirements definition, Appendix J on functional analysis, Appendix K on system architecture synthesis, Appendix L on SE analysis activities, Appendix M on human systems integration, and Appendix N on system integration.

Early validation efforts align requirements with stakeholder expectations. The systems capabilities specified by the stakeholders will be met by the combination of system elements. Problems identified for individual hardware parts or software modules should be addressed early to minimize the risk that, when these entities are finally designed and verified, they fall short of the required functionality or performance.

Many projects are driven by eager project champions who want "to get on with it." They succumb to the temptation to cut short the concept stage, and they use exaggerated projections to support starting detailed design without adequate understanding of the challenges involved. Many commissions reviewing failed systems after the fact have identified insufficient or superficial study in the concept stage as a root cause of failure.

3.3.3 Development Stage

Purpose: The Development Stage is executed to develop a system-of-interest that meets acquirer requirements and can be produced, tested [verified], evaluated, operated, supported, and retired.

The development stage includes detailed planning, development, and integration, verification and validation (IV&V) activities. Figure 3-4 illustrates the evolving baseline as system components are integrated and verified. A source of additional information about IV&V and the significance for project cost and risk when these activities are optimized was the subject of the European Union SysTest[5] program.

The following appendices elaborate on key development activities: Appendix G on planning, Appendix I on requirements definition, Appendix J on functional analysis, Appendix K on system architecture synthesis, Appendix L on SE analysis activities, Appendix M on human systems integration, and Appendix N on system integration.

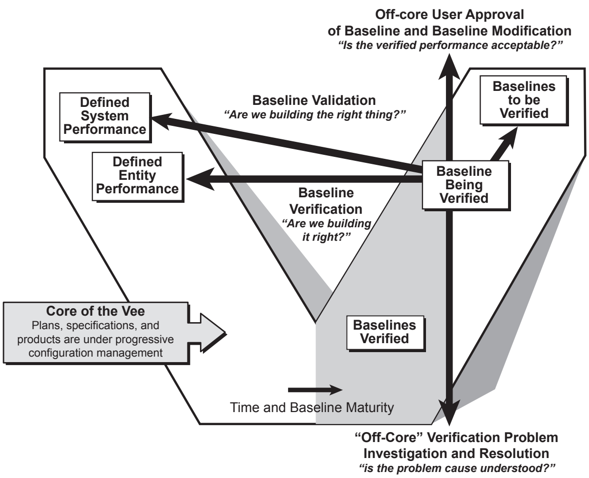


Figure 3-4 Right side of the Vee[6]

3.3.4 Production Stage

Purpose: The Production Stage is executed to produce or manufacture the product, to test [verify] the product, and to produce related supporting and enabling systems as needed.

Product modifications may be required to resolve production problems, to reduce production costs, or to enhance product or system-of-interest capabilities. Any of these may influence system requirements, and may require system re-verification or re-validation. All such changes require systems engineering assessment before changes are approved.

3.3.5 Utilization Stage

<u>Purpose:</u> *The Utilization Stage is executed to operate the product, to deliver services within intended environments and to ensure continued operational effectiveness.*

Product modifications are often planned for introduction throughout the life cycle. Such upgrades enhance the capabilities of the system. These changes should be assessed by systems engineers to ensure smooth integration with the operational system-of-interest. The corresponding technical process is the Operations Process.

3.3.6 Support Stage

<u>Purpose:</u> *The Support Stage is executed to provide logistics, maintenance, and support services that enable continued system-of-interest operation and a sustainable service.*

Modifications may be proposed to resolve supportability problems, to reduce operational costs, or to extend the life of a system. These changes require systems engineering assessment to avoid loss of system capabilities while under operation. The corresponding technical process is the Maintenance Process.

3.3.7 Retirement Stage

<u>Purpose:</u> *The Retirement Stage is executed to provide for the removal of a system-of-interest and related operational and support services, and to operate and support the retirement system itself.*

Systems engineering activities in this stage are primarily focused on ensuring that disposal requirements are satisfied. In fact, planning for disposal is part of the system definition during the concept stage. Experience in the 20th century repeatedly demonstrated the consequences when system retirement and disposal are not considered from the outset. Early in the 21st century, many countries have changed their laws to hold the creator of a system-of-interest accountable for proper end-of-life disposal of the system.

3.4 Development Stage Approaches

Per ISO/IEC 15288: *"6.3 - Different organizations may undertake different stages in the life cycle. However, each stage is conducted by the organization responsible for that stage with due consideration of the available information on life cycle plans and decisions made in preceding stages. Similarly, the organization responsible for that stage records the decisions made and records the assumptions regarding subsequent stages in the life cycle."*

A discussion of System Life Cycle Stages does not imply that the project should follow a predetermined set of activities or processes unless they add value toward achieving the final goal. Representations of stages tend to be linear in graphical depictions, but this hides the true incremental and iterative nature of the underlying processes. Illustrations which follow imply full freedom to choose a development model and are not restricted to a waterfall or other plan-driven methods. For the development stage, as for all stages, the organization will select the processes and activities that suit the project needs.

3.4.1 Plan-driven Development

The requirements/design/build/test/deploy paradigm is considered the traditional way to build systems. On projects where it is necessary to coordinate large teams of people working in multiple companies, plan-driven approaches provide an underlying framework to provide discipline to the development processes. Plan-driven methods are characterized by a systematic approach that adheres to specified processes as the system moves through a series of representations from requirements through design to finished product (Appendix H elaborates on the Integrated Product and Process Development). There is attention to the completeness of documentation, traceability from requirements, and verification of each representation after the fact.

The strengths of plan-driven methods are predictability, stability, repeatability, and high assurance. Process improvement focuses on increasing process capability through standardization, measurement, and control. These methods rely on the "master plans" to anchor their processes and provide project-wide communication. Historical data is usually carefully collected and maintained as inputs to future planning to make projections more accurate.[7]

Safety-critical products, such as the Therac-25 medical equipment described in section 3.5.1, can only meet modern certification standards by following a thorough, documented set of plans and specification. Such standards mandate strict adherence to process and specified documentation to achieve safety or security. However, unprecedented projects or projects with a high rate of unforeseeable change, predictability and stability degrade, and a project may incur significant investment trying to keep documentation and plans up-to-date.

3.4.2 Incremental and Iterative Development

Incremental and iterative development (IID) methods have been in use since the 1960's.[8] They represent a practical and useful approach which allows a project to provide an initial capability followed by successive deliveries to reach the desired system-of-interest. The goal is to provide rapid value and responsiveness. This approach is generally presented in opposition to the perceived burden associated with using any process, including those defined in this handbook.

This development approach is used when the requirements are unclear from the beginning or the customer wishes to hold the system-of-interest open to the possibilities of inserting new technology. Based on an initial set of assumptions a candidate system-of-interest is developed, and then assessed to determine if it meets user needs or requirements. If not, another evolutionary round is initiated. This process is repeated until there is a satisfied user, or until the investor runs out of interest or money.

Most literature agrees that IID methods are best applied to smaller, less complex systems or to system elements. The focus is on flexibility, and allowing selected events to be taken out of sequence when the risk is acceptable. Tailoring in this way highlights the core activities of product development.

The features that distinguish IID from the plan-driven approaches are velocity and adaptability. While market strategies often emphasize that "time to market" or speed is critical, a more appropriate criterion is "velocity," which considers direction in addition to speed. By incorporating the customer into their working-level teams, the project receives continuous feedback that they are going in a direction that satisfies the customer's highest needs first. One downside is that reactive project management with a customer that often changes direction can result in an unstable, chaotic project. On one hand, this approach avoids the loss of large investments in faulty assumptions; on the other hand, emphasis on a tactical viewpoint may generate short-term or localized solution optimizations.

A specific IID methodology called evolutionary development[9] is common in research and development environments. Figure 3-5 illustrates how this approach was used in the evolution of the tiles for the NASA Space Shuttle.[10]

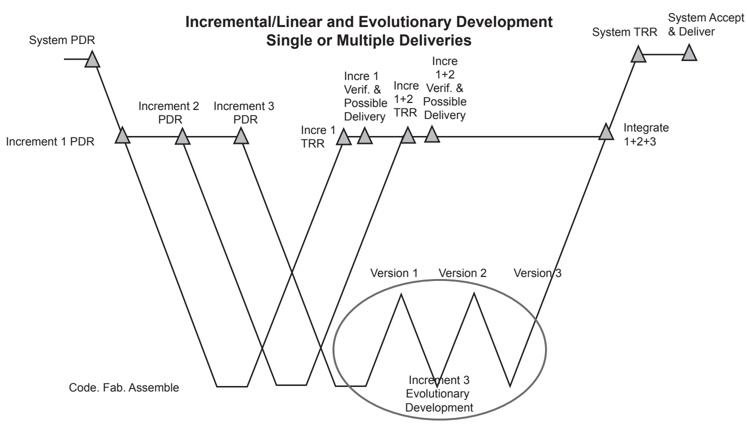


Figure 3-5 IID and Evolutionary Development[11]

3.4.3 What is best for your organization?

Conway's law suggests that effective systems design emerges from system-oriented organizations. One of the earliest books on Systems Engineering Management[12] identified three simple criteria for such an organization; facilitate communications, streamline controls, and simplify paperwork. The way to effective systems engineering management is not “in the direction of formal, formidable, massive documentation. It does, however, reside in the direction of creating a total environment which is conducive to the emergence and effective utilization of creative and inventive talents oriented toward achieving a system approach with a minimum of management encumbrances.”[13]

According to the Merriam-Webster’s Dictionary, a process can be defined as “a series of actions or operations conducing to an end.” Is process the way a company operates—from marketing to human resources, to actual development—or the way a developer produces design or code, or tests the software? Does the process refer to management, engineering, or both? Does process imply a lot of formalism and expanding effort for writing and reading documents, instead of product development? The best answer is that it depends on the situation. A “one size fits all” approach does not work when defining processes. One would hardly expect to find the same processes used in a startup e-commerce company as in NASA. The intended goal shapes a process in terms of scope (namely, the stages and activities covered) and organizational level. Depending on the perspective, processes are defined for entire organizations, teams, or an individual. In any case, the process should help by guiding people on what to do—on how to divide and coordinate the work—and by ensuring effective communication. Coordination and communication, for example, form the main problems in large projects involving many people—especially in distributed projects where people cannot communicate face to face.[14]

For a given organizational level, the process varies with the project’s goals and available resources. At a high level, the company’s business strategy determines the business approach. The main goals of time to market, minimum cost, or higher quality and customer satisfaction set the priorities. The company’s size; the number, knowledge, and experience of people[15] (both engineers and support personnel); and hardware resources determine how to achieve the goal. The application domain and the corresponding system requirements together with other constraints form another main factor. The space shuttle or nuclear plant control software embedded in a complex system have different safety and reliability constraints than a word processor running on a PC; software for a car has different time response constraints than a payroll system.

Whenever someone (be it an individual or a company) wants to reach a desired end, they must perform a series of actions or operations. They must consider the order of these actions, their dependencies, who will perform them, what they require and what they will generate, how long it will take to complete them, and what tools they will employ. Thus, they do follow a process, be it predefined or ad hoc. Because all these process components (activities, products, agents, tools) and their interactions (information flow, artifacts flow, control, communication, timing, dependencies, and concurrency) can vary, processes will differ — even if they have the same level, scope, and goal.

So why should an organization care about processes?[16] To understand, evaluate, control, learn, communicate, improve, predict, and certify the work performed. What can organizations do to processes? They can document, define, measure, analyze, assess, compare, and change them. But the most difficult question of all is, "What is best for my organization?" That answer may be contained in this handbook.

3.5 Introduction to three cases

Real-world examples are provided throughout this handbook. They draw from diverse industries and types of systems. Three case studies have been selected used to illustrate the diversity of systems to which systems engineering principles and practices can be applied; medical therapy equipment, a bridge, and a super-high-speed train. They represent examples of failed, successful, and prototype systems that all define(d) the state-of-the-art. They may be categorized as medicine, infrastructure and transportation applications; in the manufacturing and construction industry domains; with and without software elements; complex; and subject to scrutiny both in the development and utilization stages as all have a need to be safe for humans and are constrained by government regulations.

3.5.1 Case 1: Radiation Therapy; the Therac-25

Therac-25, a dual-mode linear medical accelerator was developed by the medical division of Atomic Energy Commission Limited (AECL) of Canada, starting in 1976. The completely computerized system became commercially available in 1982. This new machine could be built at lower production cost, resulting in lower prices for the customers. A series of tragic accidents led to the recommended recall, and discontinuation of production of the system.

Summary of product features and history

The Therac-25 was a medical linear accelerator. A linear accelerator (LINAC) is a particle accelerator, capable of increasing the energy of electrically charged atomic particles. The charged particles are accelerated by the introduction of an electric field, producing beams of particles, or radiation, which are then focused by magnets. LINACS are used to treat cancer patients by exposing malignant cells to radiation. Since malignant tissues are more sensitive than normal tissues to radiation exposure, a treatment plan can be developed that permits the absorption of an amount of radiation that is fatal to tumors but causes relatively minor damage to surrounding tissue.

Therac-25 was a revolutionary design comparing to its predecessors, Therac-6 and Therac-20, both with exceptional safety records. It was based on a double-pass concept that allowed a more powerful accelerator to be built into a compact and versatile machine. AECL designed Therac-25 to fully utilize the potential of software control. While Therac-6 and Therac-20 were built as stand-alone machines and could be operated without a computer, Therac-25 depended on a tight integration of software and hardware. In the new, tightly coupled system, AECL used software to monitor the state of the machine and to ensure its proper operations and safety. Previous versions had included independent circuits to monitor the status of the beam as well as hardware interlocks that prevented the machine from delivering doses of radiation that were too high, or from performing any unsafe operation that could potentially harm the patient. In Therac-25, AECL decided not to duplicate these hardware interlocks since software already performed status checks and handled all the malfunctions. This meant that the Therac-25 software had far more responsibility for safety than the software in the previous models. If in the course of treatment, the software detected a minor malfunction it would pause the treatment. In this case, the procedure could be restarted by pressing a single "proceed" key. Only if a serious malfunction was detected was it required to completely reset the treatment parameters in order to restart the machine.

Software for Therac-25 was developed from the Therac-20's software, which was developed from the Therac-6's software. One programmer, over several years, evolved the Therac-6 software into the Therac-25 software. A stand-alone real-time operating system was added along with application software written in assembly language and tested as a part of the Therac-25 system operation. In addition, significant adjustments had been made to simplify the operator interface and minimize data entry, since initial operators complained that it took too long to enter a treatment plan.

At the time of its introduction to market in 1982, Therac-25 was classified as a Class II medical device. Since the Therac-25 software was based on software used in the earlier Therac-20 and Therac-6 models, Therac-25 was approved by the Federal Drug Administration under Pre-Market Equivalency.

Six accidents involving enormous radiation overdoses to patients took place between 1985 and 1987. Tragically, three of these accidents were the direct cause of the death of the patient. This case is ranked in the top ten worst software-related incidents on many lists. Details of the accidents and analysis of the case is available from many sources.[17,18,19]

3.5.2 Case 2: Joining two countries; the Øresund Bridge

The Øresund Region is composed of eastern Denmark and southern Sweden and since 2000 is linked by the Øresund Bridge. The area includes the two major cities Copenhagen and Malmø, has a population of 3 million, and counts as Europe's eight largest economic center. One fifth of the total Danish and Swedish GNP (Gross National Product) is produced in the region. The official name of the bridge is translated "the Øresund Connection" to underscore the full integration of the region. For the first time ever, Sweden is joined permanently to the mainland of Europe by a 10-minute drive or train ride. The cost for the entire Øresund Connection construction was calculated at 30.1 billion DKK the investment expected to be paid back by 2035.

The Øresund Bridge is the world's largest composite structure, has the longest cable-stayed bridge span in the world carrying motorway and railway traffic, and boasts the highest freestanding pylons. The 7.9 km (5 mi) long bridge crosses the international navigation route between the Baltic Sea and the North Sea. A cable-stayed high bridge rises 57 m (160 feet) above the surface of the sea, with a main span of 490 m (0.3 miles). Both the main span and the approach bridges are constructed as a two-level composite steel-concrete structure. The upper deck carries a four-lane motorway, and the lower deck carries a two-track railway for both passenger trains and freight trains. The rest of the distance is spanned by the artificial island Peberholm ("Pepper" islet, named to complement the Saltholm islet to the north) and a tunnel on the Danish side that is the longest immersed concrete tunnel in the world. Since completion, Peberholm has become a natural habitat for colonies of rare birds, one of the largest of its kind in Denmark and Sweden.

Nations other than Denmark and Sweden also contributed to this project. Canada provided a floating crane, aptly named Svanen (the swan), to carry prefabricated bridge sections out to the site and place them into position. Forty-nine steel girders for the approach bridges were fabricated in Cádiz, Spain. A specially designed catamaran was built to handle transportation of the foundations for the pylons, which weighed 19,000 tons each.

The project began with well-defined time, budget, and quality constraints. The design evolved over more than seven years, from start to delivery of final documentation and maintenance manuals. More than 4000 drawings were produced. The consortium dealt with changes as necessary using a combination of technical competence and stakeholder cooperation. Notably, there were no disputes and no significant claims against the owners at the conclusion and this has been attributed to the spirit of partnership.

From the beginning, the owners defined comprehensive requirements and provided definition drawings as part of the contract documents, to ensure a project result that not only fulfilled the quality requirements on materials and workmanship, but also had the envisioned appearance. The contractor was responsible for the detailed design and for delivering a quality assured product in accordance with the owners' requirements.

Selected Requirements

The following are representative of the requirements levied at the start of the project:

Schedule: Design life 100 years; Construction time Apr. 1996 - Apr. 2000

Railway: Rail load UIC 71; Train speed 200 km/h

Motorway: Road axle load 260 kN; Vehicle speed 120 km/h

Ambient environment:

Wind speed (10 min) 61 m/s; Wave height 2.5 m; Ice thickness 0.6 m; Temperature +/- 27 ºC

Ship impact: to pylons 560 MN; to girder 35 MN

Constraints

In addition to established requirements, this project crossed national boundaries and was thereby subject to the legislations of each country. Technical requirements were based on the Eurocodes, with project specific amendments made to suit the national standards of both countries. Special safety regulations were set up for the working conditions, meeting the individual safety standards of Denmark and Sweden.

The railway link introduced yet another challenge. In Denmark, the rail traffic is right handed as on roadways, whereas the trains in Sweden pass on the left hand side. The connection needed to ensure a logical transition between the two systems, including safety aspects. In addition, the railway power supply differs between the two countries, thus it was necessary to develop a system that could accommodate power supply for both railway systems.

Bridge Design involves many disciplines

The design of a major cable-stayed bridge with approach spans for both road and railway traffic involves several disciplines as illustrated by this partial list: geotechnical engineering; aerodynamics; foundation engineering; wind tunnel tests; design of piers and pylons; design of composite girders; design of cables and anchorages; design of structural monitoring system; ship impact analysis; earthquake analysis; analysis of shrinkage and creep of concrete; ice loads analysis; fatigue analysis; pavement design; mechanical systems; electrical systems; comfort analyses for railway passengers; traffic forecast; operation and maintenance aspects; analysis of construction stages; risk analysis for construction and operation; quality management; and environmental studies and monitoring.

Risk Management

Comprehensive risk analyses were carried out in connection with the initial planning studies, including specification of requirements to secure all safety aspects. Important examples of the results of these studies for the Øresund Bridge were:

- Navigation span was increased from 330 m to 490 m
- Realignment and deepening of the navigation channel to reduce groundings
- Introduction of pier protection islands

Risks were considered in a systematic way, using contemporary risk analysis methods such as functional safety analyses using fault tree and “what if” techniques. Three main issues were considered under the design-build contract:

- General identification and assessment of construction risks
- Ship collision in connection with realignment of navigation channel
- Risks in connection with 5 years bridge operation by contractor

A fully quantified risk assessment of the human safety and traffic delay risks was carried out for a comprehensive list of hazards including: fire; explosion; train collisions and derailments; road accidents; ship collisions and groundings; aircraft collisions; environmental loads beyond design basis; and, toxic spillages. An example of a consequence of this analysis was the provision of passive fire protection on the tunnel walls and ceilings.

Environmental Considerations

Both Denmark and Sweden are proud of being among the cleanest industrial countries in the world. Their citizens, and therefore the politicians, would not allow for any adverse environmental impact from the construction or operation of a bridge. The Great Belt and Øresund Strait both constitute corridors between the salty Kattegat and the sweeter water of the Baltic Sea. Any reduction in water exchange would reduce the salt content, and, therefore, the oxygen content, of the Baltic Sea and would alter its ecological balance. The Danish and Swedish Authorities decided that the bridge should be designed in such a way that the flow-through of water, salt and oxygen into the Baltic was not affected. This requirement was designated the zero solution. In order to limit impacts on the local flora and fauna in Øresund during the construction, the Danish and Swedish authorities imposed a restriction that the spillage of seabed material from dredging operations should not exceed 5% of the dredged amounts. The zero solution was obtained by modeling with 2 different and independent hydrographical models.

In total, 18 million cubic meters of seabed materials were dredged. All dredged materials were reused for reclamation of the artificial peninsula at Kastrup and the artificial island, Peberholm. A comprehensive and intensive monitoring of the environment was performed in order to ensure and document the fulfillment of all environmental requirements. In their final status report from 2001 the Danish and Swedish Authorities concluded that the zero solution as well as all environmental requirements related to the construction of the link had been fulfilled. Continual monitoring of eel grass and common mussels showed that, after a general but minor decline, populations had recovered by the time the bridge was opened. Overall, the environment paid a low price at Øresund and the Great Belt, because it was given consideration throughout the planning and construction phases of the bridges.

This award-winning bridge is the subject of numerous articles and a PhD thesis, where details of the construction history and collaboration among all the stakeholders are provided.[20,21,22]

3.5.3 Case 3: Prototype system; The Super-high-speed train in China

Shanghai Transrapid is the first commercial high-speed commuting system using the state-of-the-art electromagnetic levitation (or maglev) technology. The train runs from Shanghai's financial district to Pudong International Airport, and the total track length is about 30 kilometers (20 miles). The train takes 7 minutes and 20 seconds to complete the journey, and can reach almost 200 mph (320 km/h) in 2 minutes, and reaches its maximum speed of 267 mph (430 km/h) within 4 minutes. The Shanghai Transrapid project took 1.2 billion USD (10 billion Yuan) and 2.5 years to complete the track. Construction began in March 2001, and public service commenced on January 1, 2003. Critics argue that the speed over such a short distance is unnecessary and that the line may never recoup this cost.

Prior to this installation, many countries had argued over the feasibility of maglev trains. They do not have wheels or use a traditional rail. Rather, powerful magnets lift the entire train about 10 millimeters above the special track, called a guideway, since it mainly directs the passage of the train. Electromagnetic force is used to make the train hover, and to provide vertical and horizontal stabilization. The frequency, intensity and direction of the electrical current in the track control the train's movement, while the power for the levitation system is supplied by the train's onboard batteries, which recharge whenever the train is moving. Maglev trains also do not have an onboard motor. The guideway contains a built-in electric motor that generates an electromagnetic field that pulls the train down the track. Putting the propulsion system in the guideway rather than onboard the trains, makes the cars lighter, which enables the train to accelerate quickly. The super-high speeds are attained largely due to the reduction of friction.

Despite the high speed, the maglev system runs more quietly than a typical commuter train, consumes less energy and is nearly impossible to derail because of the way the train's underside partially wraps around the guideway, like a giant set of arms hugging the train to the elevated platform. Passengers experience a comfortable and quiet ride due to the maglev technology and the specially designed window; noise level is less than 60 decibels at a speed of 300 km/h.

The Chinese authorities considered the economical operation, low energy consumption, less environmental impact, and high speed when choosing a solution suitable for ground transport between hubs that range from hundreds to over one thousand kilometers apart. But the same solution also needed to be suitable for modern mass rapid passenger transportation between a center city and adjacent cities. Despite the many advantages, in 1999 the technology was considered to be in an experimental stage, not yet proven by commercialized operation, its technological superiority, safety and economic performance remaining to be proved. The current line is the result of a compromise; it was built as a demonstration to verify the maturity, availability, economics and safety of a high speed maglev transportation system.

The basic technology to create a maglev system has been around since 1979, but until this project it had never been realized – mostly due to the expense of developing a new train system. Many experts believe that super-fast steel-wheel rail systems – such as those in France and Japan – have reached the limits of this technology and can not go any faster. Maglev proponents describe the system as "the first fundamental innovation in the field of railway technology since the invention of the railway" and are watching proposals for maglev installations in Germany and the USA.[23,24,25,26]

1 Used with permission of H. Stoewer
2 ISO/IEC 15288, Table D-1, page 57
3 Figure provided by Kevin Forsberg, CSM.
4 Forsberg, K., H. Mooz, H. Cotterman, *Visualizing Project Management,* 3rd Ed., J. Wiley & Sons, 2005. p. 111
5 SysTest, *Systems Verification, Validation and Testing Methodology Guidelines,* Contract: G1RD-CT-2002-00683, http://www.incose.org/secoe/0105.htm
6 Forsberg, K., H. Mooz, H. Cotterman, *Visualizing Project Management,* 3rd Ed., J. Wiley & Sons, 2005. p. 115
7 Boehm, B. and R. Turner. Balancing Agility and Discipline. Addison-Wesley, 2004.
8 Larman, C, V.Basili, "Iterative and Incremental Development: A Brief History," IEEE Software, My/Jn 2003.
9 Gilb, T., *Competitive Engineering,* Elsevier, 2005, p. 358.
10 "If I Could Do That, Then I Could…," by K. Forsberg, Proceedings of Fifth NCOSE Summer Symposium, 1995.
11 Forsberg, K., H. Mooz, H. Cotterman, *Visualizing Project Management,* 3rd Ed., J. Wiley & Sons, 2005. p. 358
12 Chase, W. P., *Management of System Engineering.* John Wiley & Sons, 1974.
13 Ibid. p 147.
14 Lindvall, Mikael and Ioana Rus, "Process Diversity in Software Development" *IEEE Software* Jl/Au 2000.
15 Cockburn, A. "Selecting a Project's Methodology". *IEEE Software,* Jl/Au 2000, p. 64-71
16 McConnell, Steve. "The Power of Process" *IEEE Computer,* May 1998
17 Jacky, Johathan. "Programmed for Disaster." *The Sciences* 29 (1989) : 22-27.
18 N. Leveson, C. S. Turner. "An Investigation of the Therac-25 Accidents." *IEEE Computer,* Vol. 26, No. 7, 1993, pp. 18-41.
19 Porrello, Anne Marie. Death and Denial: The Failure of the THERAC-25, A Medical Linear Accelerator. Downloaded from URL: http://www.csc.calpoly.edu/~jdalbey/SWE/Papers/THERAC25.html.
20 Nissen, J. The Øresund Link. *The Arup Journal,* 31(2), pp37-41, 2/1996.
21 Jensen, J. S. The Øresund Bridge – Linking Two Nations. URL: www.cowi.dk
22 Website of the Øresund Bridge Consortium, URL: osb.oeresundsbron.dk
23 "China's Supertrain Takes to Tracks." BBC News Online: Asia-Pacific. 9 Aug. 2002. 29 May 2003. http://news.bbc.co.uk/2/hi/asia-pacific/2182975.stm.
24 McGrath, Dermot. "China Awaits High Speed 'Maglev.'" *Wired News,* 23 Jan 2003. http://www.wired.com/news/technology/0,1282,57163,00.html.
25 "Transrapid." Transrapid International. 23 May 2003 www.transrapid.de.
26 Website of Shanghai Maglev Transportation Development Company, URL: www.smtdc.com/en/gycf4.asp.

THIS PAGE
INTENTIONALLY
LEFT BLANK

4 Technical Processes

4.1 Introduction

The ISO/IEC 15288 technical processes are invoked throughout the life cycle stages of a system. Technical processes are used to establish requirements for the system as the basis for the efforts to create an effective product or service; to sustain the system through its useful life; and to support retirement of the system. Figure 1-1 illustrates the relationship of the technical processes to the Agreement, Enterprise and Project Processes. Without the technical processes, the risk of project failure would be unacceptably high. In his opening keynote at the 15th Annual INCOSE International Symposium, Riley Duren of Jet Propulsion Laboratory, California, stressed that systems engineering is a way of thinking about and solving challenges and that systems engineers are the GLUE that hold the elements of complex space programs together. To achieve good results, systems engineers involve themselves in nearly every aspect of a project, pay close attention to interfaces where two or more systems or system elements work together, and establish an interaction network with stakeholders and other organizational units of the enterprise. Figure 4-1 shows the critical interactions for systems engineers.

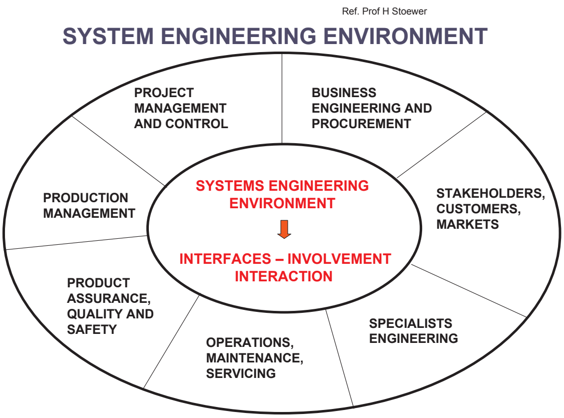


Figure 4-1 Context of Systems Engineering Technical Processes[1]

Technical processes enable systems engineers to coordinate the interactions between engineering specialists, systems stakeholders and operators, and manufacturing. They also address conformance with the expectations and legislated requirements of society. These processes lead to the creation of a full set of requirements that address the desired capabilities within the bounds of performance, environment, external interfaces, and design constraints.

4.2 Stakeholder Requirements Definition Process

4.2.1 Purpose

The purpose of the Stakeholder Requirements Definition Process is to elicit, negotiate, document, and maintain stakeholders' requirements for the system-of-interest within a defined environment. (Appendix I elaborates on the Requirements Definition Process and Appendix F elaborates on defining user needs.)

4.2.2 Description

A stakeholder is any entity (individual or organization) with a legitimate interest in the system. Typical stakeholders include users, operators, enterprise decision-makers, parties to the agreement, regulatory bodies, developing agencies, support organizations, and society-at-large. When direct contact is not possible, systems engineers find agents, such as marketing or non-governmental organizations to represent the concerns of a class of stakeholders, such as consumers, or future generations.

The Stakeholder Requirements govern the system's development; and they are an essential factor in further defining or clarifying the scope of the development project. If an enterprise is acquiring the system, this process provides the basis for the technical description of the deliverables in an agreement – typically in the form of a system-level specification and defined interfaces at the system boundaries. In the next process (Requirements Analysis Process), the verification criteria are added to this definition. Requirements management is the subject of a section in Chapter 7. Figure 4-2 is the context diagram for this process.

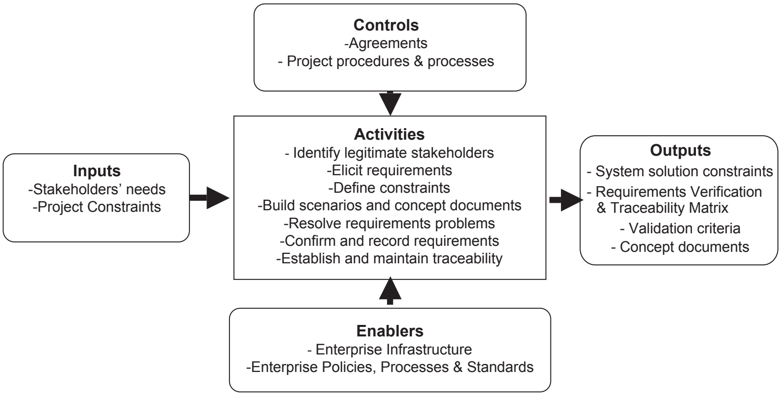


Figure 4-2 Context Diagram for Stakeholder Requirements Definition Process

4.2.3 Inputs

The inputs to this process include the description of users' needs or services that the system will provide, cost, schedule, and solution constraints, terms and conditions of the agreement, and industry specifications and standards. This process is governed by project plans and enterprise standards, guidelines, policies, procedures, and reporting mechanisms.

4.2.4 Outputs

The outputs of this process consist of formally documented and approved stakeholder requirements that will govern the project, including: required system capabilities, functions and/or services; quality standards; cost and schedule constraints; concept of operations; and concept of support. The outputs should include measures of effectiveness and suitability that will be used for assessing the realized system and enabling systems. The initial Requirements Verification and Traceability Matrix (RVTM) is created as part of this process. Validation criteria may specify who will perform validation activities, and the environments of the system-of-interest and enabling systems. Other outputs establish the initial baseline for project scope and associated agreements. The following are instances of concept documentation:

- *Concept of Production* describes the way the system will be manufactured.
- *Concept of Deployment* describes the way the system will be delivered and installed.
- *Concept of Operations (ConOps)* describes the way the system works from the operator's perspective. The ConOps includes the User Description which summarizes the needs, goals,and characteristics of the system's user community, including operators, maintainors, and support personnel.
- *Concept of Support* describes the desired support infrastructure and manpower considerations for maintaining the system after it is deployed. This includes specifying equipment, procedures, facilities and operator training requirements.
- *Concept of Disposal* describes the way the system will be removed from operation and retired.

4.2.5 Process Activities

This process includes the following activities:

- Identify stakeholders who will have an interest in the system throughout its entire life cycle.
- Elicit requirements – what the system must accomplish and how well.
- Define constraints imposed by agreements or interfaces with legacy enabling systems.
- Establish critical and desired system performance – thresholds and objectives for system performance parameters that are critical for system success and those that are desired but may be subject to compromise in order to meet the critical parameters
- Establish measures of effectiveness and suitability – measures that reflect overall customer/user satisfaction (e.g. performance, safety, reliability, availability, maintainability, and workload requirements).

- Analyze requirements for clarity, completeness and consistency.
- Negotiate modifications to resolve unrealizable or impractical requirements.
- Validate, record, and maintain stakeholder requirements throughout the system life cycle, and beyond for historical or archival purposes.
- Establish and maintain a traceability matrix to document how the formal requirements are intended to meet the stakeholder objectives and achieve stakeholder agreement.

■ *Common approaches and tips:*

- Use scenarios to define the concept documents – bound the range of anticipated uses of system products, the intended operational environment and interfacing systems, platforms or products. Scenarios help identify requirements that might otherwise be overlooked. Social and organizational influences also emerge from using scenarios.
- A description of the user community is necessary to provide common understanding across the effort and to validate the appropriateness of scenarios. A user description may cover the demographic group(s) to which a product will be marketed or the specific personnel categories that will be assigned to employ the system or otherwise benefit from its operation.
- Once established, the stakeholders' requirements are placed under configuration control.
- Establish good relationships and open communications between systems engineers and stakeholders. This is helpful when negotiations begin to refine and clarify the set of requirements.
- Identify all stakeholders; it is critical to identify and include key system stakeholders in this process including the development/design team.
- Avoid designing a final solution or establishing unjustified constraints on the solution.
- Avoid acceptance of unrealistic or competing objectives.
- Write clearly and create statements with quantifiable values.[2]
- Capture source and rationale for each requirement.[3]

4.3 Requirements Analysis Process

4.3.1 Purpose

The purpose of the Requirements Analysis Process is to review, assess, prioritize, and balance all stakeholder and derived requirements (including constraints); and to transform those requirements into a functional and technical view of a system description capable of meeting the stakeholders' needs. This view can be expressed in a specification, set of drawings or any other means that provides effective communication.

4.3.2 Description

Requirements analysis is an interim process: the output of the process must be compared for traceability and consistency with the Stakeholder Requirements, before being used to drive the Architectural Design Process, without introducing implementation biases. Figure 4-3 is the context diagram for Requirements Analysis.

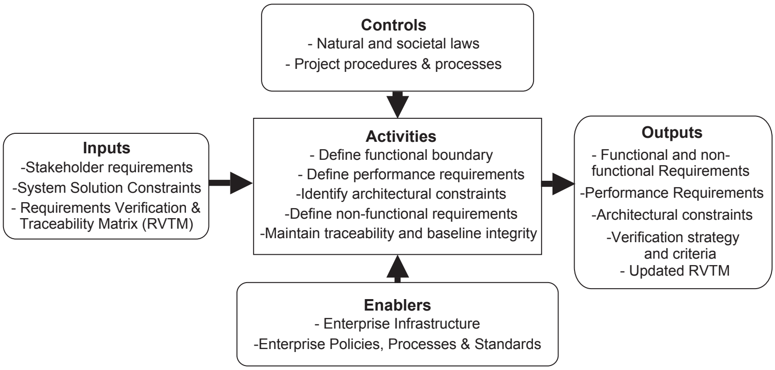


Figure 4-3 Context Diagram for the Requirements Analysis Process

4.3.3 Inputs

The primary input to the Requirements Analysis Process is the baseline documented during the Stakeholder Requirements Definition Process. Additional inputs to the Requirements Analysis Process include applicable statutes, regulations, and policies; the intended operational use and utilization environment for the system; any design or enterprise constraints; manufacturing; life cycle support considerations; design considerations (e.g. applicable standards, environmental and safety concerns, etc.); and any decisions or data resulting from previous phases of development.

4.3.4 Outputs

The output of Requirements Analysis is a technical description of characteristics the future system must have in order to meet Stakeholder Requirements – not a specific solution – which will be evolved in subsequent development processes. The project team derives additional requirements resulting from analysis of the input Stakeholder Requirements as required to meet project and design constraints; defines the functional boundaries for the system to be developed; and identifies and documents any interfaces and information exchange requirements with systems external to the functional boundaries. The total set of requirements encompasses the functional, performance, and non-functional requirements and the architectural constraints. Any decisions taken are documented in the information repository.

4.3.5 Process Activities

This process includes the following activities:

- Define and specify the functional boundary and performance. This will specify what the system should be able to do (functional requirements) when fielded and operated in its intended operating environment. The levels and measures of performance (MOP) for the top-level system functional requirements required to satisfy the system Measures of Effectiveness (MOE) are determined from the following:
- Selected Standards – identify standards required to meet quality or design considerations imposed as defined stakeholder requirements or derived to meet enterprise, industry, or domain requirements
- System Boundaries – clearly identify system elements under design control of the project team and/or enterprise and expected interactions with systems external to that control boundary as defined in the negotiated Interface Control Document (ICD). After agreement the ICD is placed under formal change control.
- External Interfaces – functional and design interfaces to interacting systems, platforms, and/or humans external to the system boundary as negotiated in the ICD
- Utilization Environment(s) – identify all environmental factors (natural or induced) that may affect system performance, impact human comfort or safety, or cause human error for each of the operational scenarios envisioned for system use
- Life Cycle Process Requirements – conditions or design factors that facilitate and foster efficient and cost-effective lifecycle functions (i.e. Production, Deployment, Transition, Operation, Maintenance, Reengineering/Upgrade, and Disposal)
- Design considerations – including human systems integration (manpower, personnel, training, human factors engineering, environment, safety, occupational health, survivability, habitability), system security requirements (e.g. information assurance, anti-tamper provisions), and potential environmental impact
- Design constraints – including physical limitations (e.g. weight, form/fit factors), manpower, personnel, and other resource constraints on operation of the system, and defined interfaces with host platforms and interacting systems external to the system boundary, including supply, maintenance, and training infrastructures.
- Define Verification Criteria – concurrent with analysis, to ensure verifiable requirements
- Maintain continuity of configuration control and traceability.

■ *Common approaches and tips:*

- Integrated Product Teams (with acquirer-supplier participation) are an effective practice to bring together the necessary expertise.[4]
- Use failure modes, effects, and criticality analysis (FMECA) or hazard analysis to identify the critical system level requirements. See chapters 7 and 9 for additional discussion about identifying the non-functional requirements.

- Use specially designed requirements management tools.[5]
- Begin from the very beginning to maintain requirements traceability.
- Avoid deriving requirements that are not consistent with other requirements or constraints.
- Create templates for constructing requirements statements.[6]

4.4 Architectural Design Process

4.4.1 Purpose

The purpose of the Architectural Design Process is to synthesize a system architecture baseline that satisfies the requirements. (Appendix K elaborates on the System Architecture Synthesis.)

4.4.2 Description

The Architectural Design Process requires the participation of systems engineers joined by relevant specialists in the system domain. When alternative solutions present themselves, technical analysis and decisions are taken as part of this process to identify a set of system elements. Integration is defined for the system, not the individual system elements. Figure 4.4 is the context diagram for the Architectural Design Process.

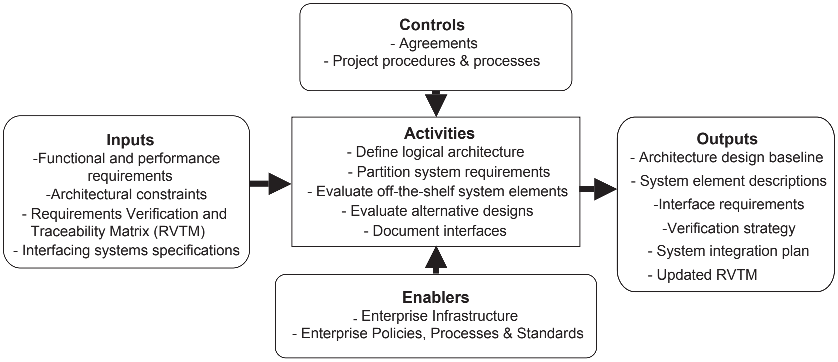


Figure 4-4 Context Diagram for the Architectural Design Process

4.4.3 Inputs

Architectural design begins from the baseline functional and performance requirements, architectural constraints, and traceability matrix. Specifications for enabling systems are used to drive interface design. Specifications for reusable system elements are used when designing for product lines.

4.4.4 Outputs

The result of this process is an architectural design that is placed under configuration management. This baseline includes:

- System element detailed descriptions with documented justification for concept selections
- Requirements assigned to system elements and documented in a traceability matrix
- Interface requirements and a plan for system integration and verification strategy

4.4.5 Process Activities

The following processes contribute to architectural design:

- Define a consistent logical architecture – capture the logical sequencing and interaction of system functions or logical elements
- Partition system requirements and allocate them to system elements and subsystems with associated performance requirements – evaluate off-the-shelf solutions that already exist.
- Evaluate alternative design solutions – see chapter 7 for a discussion of trade studies; consider these selection criteria:
  - Measures of the system's ability to fulfil its objectives as defined by the requirements
  - Its ability to operate within resource constraints
  - Its accommodation of interfaces
  - Costs, economic and otherwise, of implementing and operating the system over its entire life cycle
  - Side effects, both positive and adverse, associated with architectural options
- Identify interfaces and interactions between system elements (including human elements of the system) and with external and enabling systems
- Define the system integration strategy and plan (to include human system integration).
- Document and maintain the Architectural Design and relevant decisions made to reach agreement on the baseline design.
- Establish and maintain the traceability between requirements and system elements.
- Define Verification and Validation Criteria for the system elements.

■ *Common approaches and tips:*

- Modeling techniques such as SysML, discussed in chapter 7, are useful in deriving a logical architecture.
- Use an Integrated Product Team for analysis and review; working together as a team helps break down communications silos and facilitates decision-making.
- Architecture and Design Patterns can be useful for establishing a system framework.

- System elements can be developed in a top-down partitioning exercise that allocates the functional elements to physical or virtual system elements. Ideally, interface requirements between these system elements are minimized. At the same time, off-the-shelf or previously developed system elements are considered within the constraints of the contracting strategy.
- During this process, consider emergent properties, feature interactions, and human-system interactions.[7]

4.5 Implementation Process

4.5.1 Purpose

The purpose of the Implementation Process is to design, create or fabricate a system element conforming to that element's detailed description. The element is constructed employing appropriate technology and industry practices. This process straddles the Development and Production stages.

4.5.2 Description

During the Implementation Process, engineers follow the requirements allocated to the system element to design, fabricate, code, or build each individual element using specified materials, processes, physical or logical arrangements, standards, technologies, and/or information flows outlined in detailed drawings or other design documentation. Requirements are verified and stakeholder requirements are validated. If subsequent configuration audits reveal discrepancies, recursive interactions occur with predecessor activities or processes as required to correct them. Figure 4-5 is the context diagram for the Implementation Process.

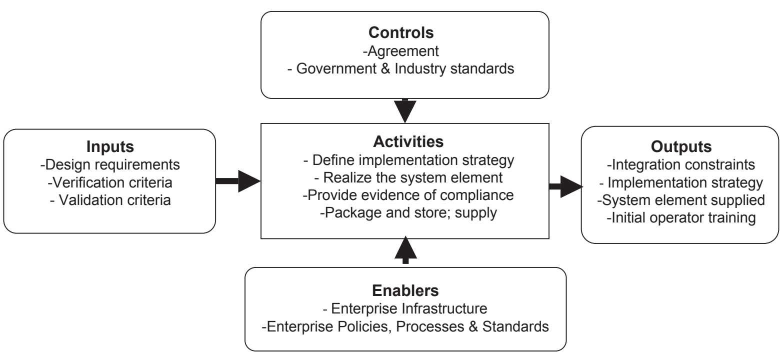


Figure 4-5 Context Diagram for the Implementation Process

4.5.3 Inputs

The system element requirements and the associated verification and validation criteria are input to this process from the Architectural Design Process. Execution of the Implementation Process is governed by

- government and industry standards, terms of any agreements
- conditions of the agreement for packaging, storage, and initial operator training (PHS&T) – see discussion of PHS&T in Chapter 9
- enterprise safety practices and other guidelines

4.5.4 Outputs

Outputs from this process include:

- Refined Implementation Strategy and Integration Constraints
- System element – verified and validated – supplied according to agreement
- Detailed drawings and codes and material specifications
- Updated design documentation – as required by corrective action or adaptations caused by acquisition or conformance to regulations
- Operator/Maintenance training capabilities and documentation and initial staff of trained operators, maintainers, and support personnel, according to agreement

4.5.5 Process Activities

Implementation Process activities begin with detailed design and include developing an Implementation Strategy that defines fabrication/coding procedures, tools and equipment to be used, implementation tolerances, and the means and criteria for auditing configuration of resulting elements to the detailed design documentation. In the case of repeated system element implementations (such as for mass manufacturing or replacement elements) the implementation strategy is defined/refined to achieve consistent and repeatable element production; and it is retained in the project decision database for future use. As required, data for training users on correct and safe procedures for operating and maintaining that element—either as a stand-alone end item or as part of a larger system—are developed and added to the project database. Detailed product, process, material specifications ("Build-to" or "Code-to" documents) and corresponding analysis are completed.

- Conduct peer reviews and unit testing – inspect and test software for correct functionality, white box testing, etc. in accordance with software/hardware best practices
- Conduct hardware conformation audits – compare hardware elements to detailed drawings to assure that each element meets its detailed specifications prior to integration with other elements in higher configuration items or assemblies

- Initial training capability and draft training documentation - to be used to provide the user community with the ability to operate, conduct failure detection and isolation, and maintain system components, subsystems, and the system as appropriate
- Train initial operators and maintainers – on the use of elements that provide a human-system interface or require maintenance actions at the element level

■ *Common approaches and tips:*

- Keep the Integrated Product Team engaged to assist with configuration issues and redesign.
- Inspections are a proactive way to build in quality.[8]
- In anticipation of improving process control, reducing production inspections, and lowering maintenance activities, many manufacturing firms use Design for Six Sigma, or Lean Manufacturing.
- Conduct hardware conformation audits or system element level hardware testing; ensure sufficient software unit testing prior to integration.
- Validate simulations; interface simulator drivers should be representative of tactical environments.

4.6 Integration Process

4.6.1 Purpose

The purpose of the Integration Process is to realize the system-of-interest by progressively combining system elements in accordance with the architectural design requirements and the integration strategy. This process is successively repeated in combination with the Verification and Validation Processes as appropriate. (Appendix N elaborates on the System Integration process.)

4.6.2. Description

The Integration Process includes activities to acquire or design and build enabling systems needed to support the integration of system elements and demonstration of end-to-end operation (system build). System build is bottom-up. That is, parts and components at the bottom of the system hierarchy are integrated and tested first. This process confirms all boundaries between system elements have been correctly identified and described, including physical, logical, and human-system interfaces and interactions (physical, sensory, and cognitive); and confirms that all functional, performance, and design requirements and constraints are satisfied. Interim assembly configurations are tested to assure correct flow of information and data across internal and external interfaces to reduce risk, and minimize errors and time spent isolating and correcting them. Figure 4-6 is the context diagram for the Integration Process.

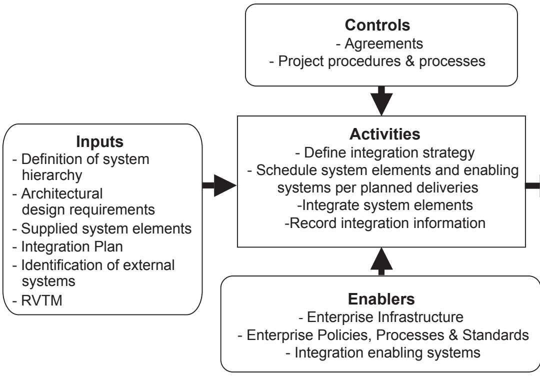


Figure 4-6 Context Diagram for the Integration Process

4.6.3 Inputs

Inputs for this process include:

- Integration plan and assembly requirements
- Integration technology constraints
- Enabling systems – Integration tools, facilities, and test equipment
- Applicable internal and external system element interface specifications

4.6.4 Outputs

Outputs for this process include:

- Integration test and analysis results, including areas of non-conformance
- Updated Interface Control Documents (ICDs) and Interface Specifications
- Updated Requirements Verification & Traceability Matrix (RVTM)
- Validated internal interfaces
- Completed subsystem or system ready for verification
- Product assembly drawings and manufacturing tool drawings

4.6.5 Process Activities:

Activities in the Integration Process include:

- Schedule Integration Testing Tools and Facilities
- Assemble system elements according to the integration plan
- Validate and Verify Interfaces – confirm correct flow of information across internal interfaces through “black box testing” at each successive level of assembly

- Test and analyze Assemblies – confirm correct functionality of assembled products through integration testing and analysis at each successive level of assembly
- Document integration testing and analysis results
- Document and control the architectural baseline – this includes capturing any modifications required during this process

■ *Common approaches and tips:*

- Keep the Integrated Product Team engaged to assist with configuration issues and redesign.
- Maintain configuration control over including drawings, specifications, interface control drawings, and published analyses.
- Define an integration strategy that accounts for the schedule of availability of system elements (including the humans that will use, operate, maintain, and sustain the system), and is consistent with fault isolation and diagnosis engineering practices.

4.7 Verification Process

4.7.1 Purpose

The purpose of the Verification Process is to confirm that all requirements are fulfilled by the system elements and eventual system-of-interest, i.e. that the system has been built right. This process establishes the procedure for taking remedial actions in the event of non-conformance.

4.7.2 Content/Description

The Verification Process confirms that all elements of the system-of-interest perform their intended functions and meet the performance requirements allocated to them. Verification methods include test, inspection, analysis, and demonstration and are discussed in more detail in chapter 8. Verification activities are determined by the perceived risks, safety, and criticality of the element under consideration.

A key outcome of the Planning Process is the creation of project procedures and processes that specify the forms of system assessments (conformation audits, integration testing, verification, and validation) in appropriate project documents (e.g. systems engineering plans, schedules, and specifications). Specification of verification criteria takes place as the requirements are written, but the creation of a procedure to assess compliance is part of this process. Figure 4-7 is the context diagram for the Verification Process.

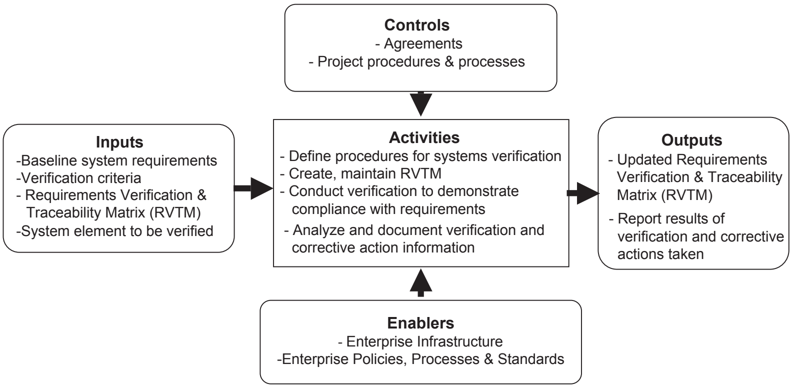


Figure 4-7 Context Diagram for the Verification Process

4.7.3 Inputs

Within the guidelines of enterprise policies and established project directives, system elements are verified against the baseline requirements and the information is maintained in a Requirements Verification and Traceability Matrix.

4.7.4 Outputs

The outputs of this process are documentation of the verification results, a record of any corrective actions recommended, Design Feedback/Corrective Actions taken, and evidence that the system element or system satisfies the requirements, or not. The RVTM is updated.

4.7.5 Process Activities

The Verification Processes include:

- Develop verification procedures
- Schedule/confirm/install verification enabling systems
- Execute verification procedures
- Document verification results and enter data into the RVTM

■ *Common approaches and tips:*

- The Requirements Verification and Traceability Matrix (RVTM) is frequently used as a single point of accountability for tracing a requirement back to the source of the need and forward through the life cycle to assess that the need has been met.
- Beware the temptation to reduce verification activities due to budget or schedule overruns – remember the message of Figure 2-3 – discrepancies and errors are more costly to correct later in the life cycle.[9]

- Avoid conducting verification late in the schedule when there is less time to handle discrepancies, or too early, before development is complete.

4.8 Transition Process

4.8.1 Purpose

The purpose of the Transition Process is to transfer custody of the system and responsibility for system support from one organizational entity to another. This includes (but is not limited to) transfer of custody from the development team to the organizations that will subsequently operate and support the system. Successful conclusion of the Transition Process typically marks the beginning of the Utilization Stage of the system-of-interest.

4.8.2 Description

The Transition Process installs a verified system in the operational environment along with relevant enabling systems, such as, operator training systems, as defined in the agreement. As part of this process, the acquirer accepts that the system provides the specified capabilities in the intended operational environment prior to allowing a change in control, ownership, and/or custody. While this is a relatively short process, it should be carefully planned to avoid surprises and recrimination on either side of the agreement; and transition plans should be tracked and monitored to ensure all activities are completed to both parties' satisfaction. Figure 4-8 is the context diagram for the Transition Process.

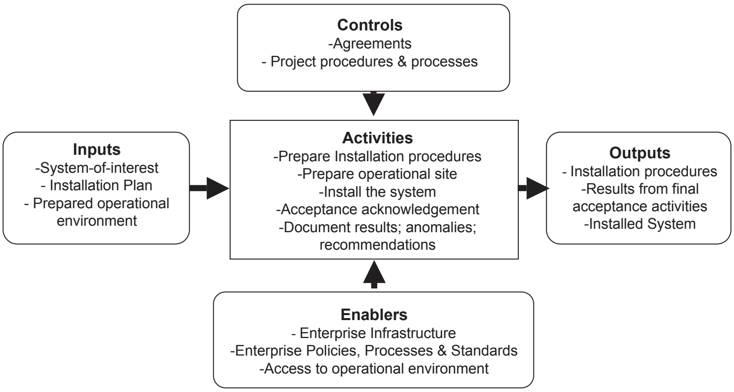


Figure 4-8 Context Diagram for the Transition Process

4.8.3 Inputs

Availability of the system-of-interest together with enabling systems is prerequisite to begin the Transition Process. Installation and commissioning of the system includes the humans that will operate, maintain, and sustain the system, and is conducted according to agreements, and applicable health, safety, security and environmental regulations.

4.8.4 Outputs

At the conclusion of the Transition Process the system is installed, acceptance criteria are met or discrepancies documented with recommended and agreed corrective actions.

4.8.5 Process Activities

The Transition Processes include:

- Prepare a transition strategy including operator training and logistics support.
- Train the users in the proper use of the system; affirm users have the knowledge and skill levels necessary to perform Operation and Maintenance activities.
- Receive final confirmation that the system—as operated and maintained by the intended users—meets their needs. This process typically ends with a formal, written acknowledgement that the system has been properly installed and verified, that all issues and action items have been resolved, and that all agreements pertaining to development and delivery of a fully supportable system have been satisfied fully or adjudicated.
- Post-implementation problems are documented and may lead to corrective actions or changes to the requirements.

■ *Common approaches and tips:*

- When acceptance activities can not be conducted within the operational environment, a representative locale is selected.
- This process relies heavily on quality assurance and configuration management documentation.

4.9 Validation Process

4.9.1 Purpose

The purpose of the Validation Process is to confirm that the realized system complies with the stakeholder requirements. System validation is subject to approval by the project authority and key stakeholders. This process is invoked during the Stakeholders Requirements Definition Process to confirm that the requirements properly reflect the stakeholder needs and to establish validation criteria, i.e. that the right system has been built. This process is also invoked during the Transition Process to handle the acceptance activities.

4.9.2 Description

In-process validation starts with a comparative assessment as a means to determine if stakeholders' requirements and defined measures of effectiveness have been correctly translated into technical design specifications and measures of performance. Validation criteria are selected based upon the perceived risks, safety and criticality. Figure 4-9 is the context diagram for the Validation Process.

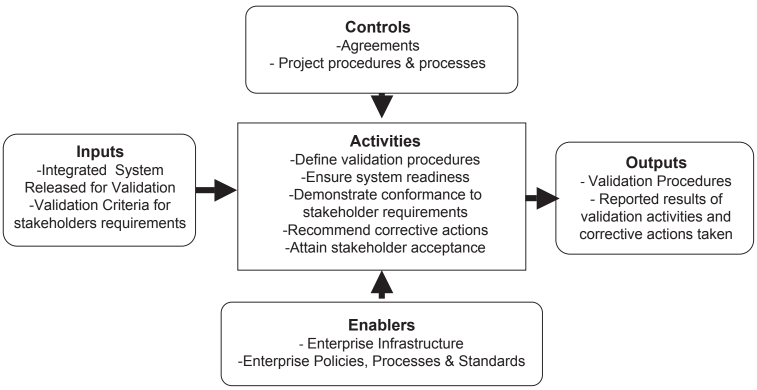


Figure 4-9 Context Diagram for the Validation Process

4.9.3 Inputs

When stakeholder requirements are elicited, validation criteria are applied before proceeding. After the system-of-interest is verified, it is subjected to the validation criteria. The previously established requirements traceability matrix is maintained. The activities are governed by the agreements, project procedures, and appropriate enterprise policies.

4.9.4 Outputs

The primary outputs of the Validation Process are

- Validation activity results
- Design feedback/corrective actions
- Approved system baseline

4.9.5 Process Activities

The following activities are part of the Validation process.

- Develop validation procedures that demonstrate that the system is fit for its purpose and satisfies the stakeholders' requirements.
- Ensure readiness to conduct validation – system, enabling systems, and trained operators.
- Conduct validation to demonstrate conformance to stakeholder requirements.
- Document validation results and enter data into traceability matrix
- If anomalies are detected, analyze for corrective actions, detect trends in failure to find threats to the system and evidence of design errors.

■ *Common approaches and tips:*

- Validation methods during the concept phase include developing assessment scenarios exercising all system modes, and demonstrating system-level performance over the entire operating regime. The system design team uses the results of this activity to forecast success in meeting the expectations of users and the acquirer, as well as to provide feedback to identify and correct performance deficiencies before implementation.[10]

4.10 Operation Process

4.10.1 Purpose

The purpose of the Operation Process is to use the system to deliver its services. This process is often executed concurrent with the Maintenance Process.

4.10.2 Description

The Operation Process sustains system services by supplying personnel to operate the system, monitoring operator-system performance, and monitoring the system performance. When the system replaces an existing system, it may be necessary to manage the migration between systems such that persistent stakeholders do not experience a breakdown in services.

The Operation Stage of a system usually accounts for the largest portion of the total life cycle cost. If system performance falls outside acceptable parameters, this may indicate the need for corrective actions, in accordance with the Concept of Support and any associated agreements. When the system or any of its constituent elements reach the end of their planned or useful life, the team may enter the Disposal Process. Figure 4-10 is the context diagram for the Operation Process.

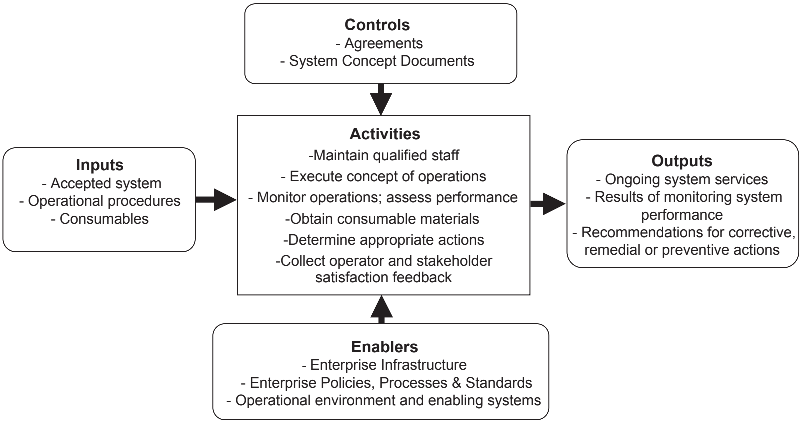


Figure 4-10 Context Diagram for the Operation Process

4.10.3 Inputs

The Operation Process identifies and analyzes operational performance in the context of agreements, stakeholder requirements and organizational constraints. Concept documents generated early in the life cycle are used to direct the activities of this process.

4.10.4 Outputs

Outputs of the Operation Process include:

- Operational strategy – including staffing and sustainment of enabling systems and materials
- System performance reports (statistics, usage data, and operational cost data)
- System trouble/anomaly reports with recommendations for appropriate action
- Operational availability constraints – to influence future design and specification of similar systems or reused systems-elements

4.10.5 Process Activities

Activities of the Operation Process include:

- Provide operator training
- Track system performance and account for operational availability
- Perform operational analysis
- Manage operational support logistics
- Document system status and actions taken
- Report malfunctions and recommendations for improvement

■ *Common approaches and tips:*

- The Operations Process corresponds to a life cycle phase called the Utilization Stage.
- Depending on the nature of agreements between different organizations, the development team may continuously or routinely communicate with users to determine the degree to which delivered services continue to satisfy their needs. The system may exhibit unacceptable performance when system elements implemented in hardware have exceeded their useful life or changes in the operational environment affect system performance. In the event of system failures or anomalies, it may be necessary to conduct engineering investigations to identify the source(s) of the failure and determine appropriate corrective actions. Systems engineers can assist in these activities.

4.11 Maintenance Process

4.11.1 Purpose

The purpose of the Maintenance Process is to sustain the system through its useful life.

4.11.2 Description

The Maintenance Process includes the activities to provide operations support, logistics, and material management. Based on feedback from ongoing monitoring of the operational environment, problems are identified and corrective, remedial or preventive actions are taken to restore full system capability. This process also contributes to the Requirements Analysis Process when considerations of constraints imposed in later life cycle stages are used to influence the system requirements and architectural design.

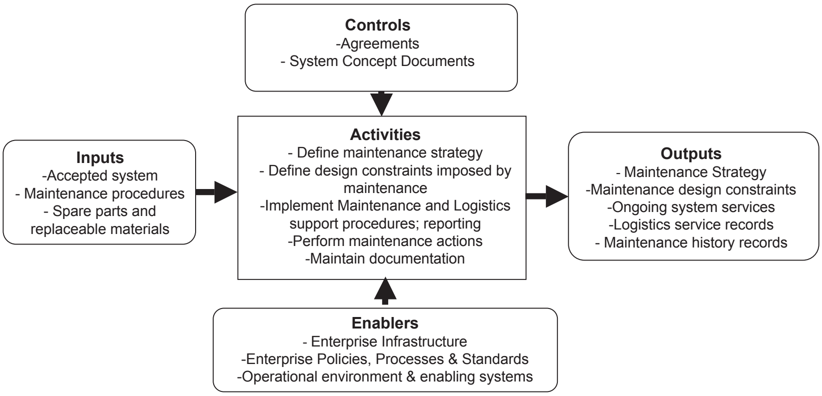


Figure 4-11 Context Diagram for the Maintenance Process

4.11.3 Inputs

The Maintenance Process identifies and analyzes maintenance and support activities in the context of agreements, stakeholder requirements and organizational constraints. Concept documents generated early in the life cycle are used to direct the activities of this process.

4.11.4 Outputs

The following are outputs of the Maintenance Process.

- Maintenance strategy – accounts for the system's technical availability, replacements for system elements and logistical support, maintenance personnel training and staff requirements
- Maintenance constraints (to influence system requirements before design)
- Reporting of failures and recommendations for action
- Failure and lifetime performance data

4.11.5 Process Activities

The following activities occur under the Maintenance Process.

- Establish a maintenance strategy
- Define maintenance constraints on system requirements
- Obtain the enabling systems, system elements and other services used for maintenance of the system, monitor replenishment levels of spare parts, and manage the skills and availability of trained maintenance personnel
- Implement maintenance and problem resolution procedures – including scheduled replacement of system elements prior to failure (preventive maintenance)
- Maintain a history of failures, actions taken, and other trends to inform operations and maintenance personnel, and other projects creating or utilizing similar system elements
- Monitor customer satisfaction with system and maintenance support

■ *Common approaches and tips:*

- The Maintenance Process corresponds to a life cycle phase called the Support Stage.
- Use historic data and performance statistics to maintain high levels of reliability and availability; and provide input to improve the design of operational and future systems.
- Planning for Maintenance begins early in the system life cycle with the development of supportability criteria. These criteria, which include reliability and maintainability requirements as well as personnel, training, facilities, etc. are included in the defined stakeholder requirements or system specification to ensure that they are considered in the system design. Chapter 9 contains a detailed discussion of acquisition logistics.

- Maintain configuration management control throughout the Utilization and Support Stages in support of the Maintenance Process.

4.12 Disposal Process

4.12.1 Purpose

The purpose of the Disposal Process is to remove a system element from the operational environment with the intent of permanently terminating its use; and to deal with any hazardous or toxic materials or waste products in accordance with applicable guidance, policy, regulations, and statutes.

4.12.2 Description

This process is implemented in the Disposal Stage, but Disposal is a life cycle support process because concurrent consideration of disposal during the Development Stage generates requirements and constraints that must be balanced with defined stakeholders' requirements and other design considerations. Environmental concerns are driving the designer to consider reclaiming the materials or recycling them into new systems. Regulatory reporting requirements are addressed by this process. Figure 4-12 is the context diagram for the Disposal Process.

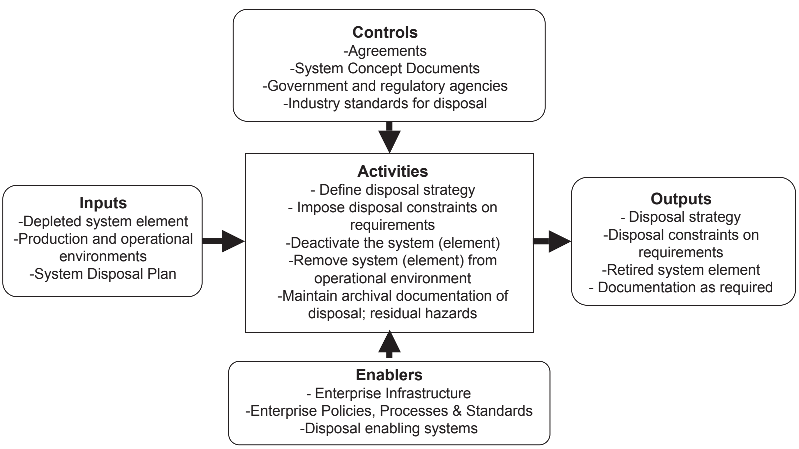


Figure 4-12 Context Diagram for the Disposal Process

4.12.3 Inputs

The Disposal Process works on a depleted system or system elements (for example, batteries) and implements the disposal plan according to applicable environmental laws, regulations, and policy. Stakeholder agreements and industry standards for disposal can also govern decision-making for this process. If production and operational environments must be restored to former conditions, details of the initial state are relevant.

4.12.4 Outputs

The first application of the Disposal Process results in a set of constraints on the system requirements and a strategy for disposal of the system and relevant system elements. The outcome from retirement or disposal may include an inventory of system elements for reuse/storage, and any documentation required by regulation or enterprise standards.

4.12.5 Process Activities

As required, Disposal Process activities include deactivating the elements to be terminated; disassembling the elements for ease of handling; removing the elements and any associated waste products from the operational site; removal of materials from storage sites; and consigning the elements and waste products for destruction or permanent storage. The Disposal Process also includes any steps necessary to return the environment to an acceptable condition; handling all system elements and waste products in an environmentally sound manner in accordance with applicable legislation, organizational constraints, and stakeholder agreements; and documenting and retaining records of Disposal activities as required for monitoring by external oversight or regulatory agencies.

■ *Common approaches and tips:*

- The project team conducts analyses to develop solutions for ultimate disposition of the system, constituent elements, and waste products based on evaluation of alternative disposal methods available. Methods addressed should include storing, dismantling, reusing, recycling, reprocessing and destroying end products, enabling systems, system elements, and materials.
- Disposal analyses include consideration of costs, disposal sites, environmental impacts, health and safety issues, responsible agencies, handling and shipping, supporting items, and applicable federal, state, local, and host nation regulations.
- Disposal analyses support selection of system elements and materials that will be used in the system design; and they should be readdressed to consider design and project impacts from changing laws and regulations throughout the project life cycle.
- Disposal Strategy and design considerations are updated throughout the system life cycle in response to changes in applicable laws, regulations, and policy
- Consider donating an obsolete system; many items, both systems and information, of cultural and historical value have been lost to posterity because

museums and conservatories were not considered as an option during the disposal stage.

- Concepts such as Zero Footprint and Zero Emissions drive current trends toward corporate social responsibility that influence decision-making regarding cleaner production and operational environments and eventual disposal of depleted materials and systems.[11]
- The ISO 14000 series includes standards for Environmental Management Systems and Life Cycle Assessment.[12]
- Instead of designing cradle-to-grave products, dumped in landfills at the end of their 'life,' a new concept is transforming industry by creating products for cradle-to-cradle cycles, whose materials are perpetually circulated in closed loops. Maintaining materials in closed loops maximizes material value without damaging ecosystems.[13]

---

1 Used with permission, Professor Heinz Stoewer.

2 Gilb, T., *Competitive Engineering,* Elsevier, 2005.

3 Hook, I., *Customer-Centered Products: Creating Successful Products Through Smart Requirements Management,* Amacon, 2000.

4 Martin, J. N., *Systems Engineering Guidebook,* CRC Press, 1996.

5 see results of INCOSE vendor survey at www.incose.org/ProductsPubs/products/toolsdatabase.aspx

6 an example is Gilb's Planguage format, see URL www.gilb.com

7 ISO 13407: *Human-centered design processes for interactive systems*

8 Gilb, T., and Dorothy Graham, *Software Inspection*. Addison-Wesley Longman, 1993.

9 *SysTest, Systems Verification, Validation and Testing Methodology Guidelines,* Contract: G1RD-CT-2002-00683, http://www.incose.org/secoe/0105.htm

10 Ibid.

11 see URL: www.zerofootprint.net/ and www.zeri.org/

12 see URL: www.iso-14001.org.uk/

13 see URL: www.mcdonough.com/

5 Project Processes

5.1 Introduction

Within the system life cycle, the creation or upgrade of products and services is managed by the conduct of projects. For this reason, it is important to understand the contribution of systems engineering to the management of the project. Systems engineers continually interact with project management as illustrated in Figure 5-1 below.

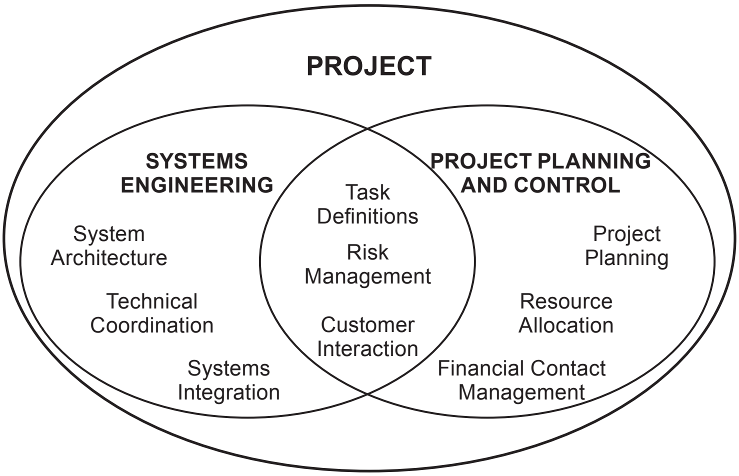


Figure 5-1 Project Management/Systems Engineering Overlap[1]

The processes described in this section are applied according to the risk and complexity of the project. These processes fall in two categories; project specific processes include project planning, project assessment, and project control; life cycle processes, which apply both inside and outside the project context, include decision-making, risk management, configuration management, and information management. Many of these latter processes are found throughout an enterprise as they are essential to generic management practices. This chapter of the handbook focuses on processes relevant to the technical coordination of a project. Table 5-1 contains an acronym list for acronyms that appear in the context diagrams in this chapter.

Table 5-1 Acronym List

CMP	Configuration Management Plan
IMP	Information Management Plan
QMP	Quality Management Plan
RMP	Risk Management Plan
SEP	Systems Engineering Plan
WBS	Work Breakdown Structure

5.2 Project Planning Process

5.2.1 Purpose

Project planning establishes the direction and infrastructure necessary to assess and control the progress of a project. This process identifies the details of the work and the right set of personnel/skills/facilities with a schedule of need for resources from within and outside the enterprise. (Appendix G elaborates on the Systems Engineering Technical Management.)

5.2.2 Description

Project planning starts with a statement of need, often expressed in a project proposal. The planning process is performed in the context of the enterprise. Enterprise processes establish and identify relevant policies and procedures for managing and executing a technical effort; identifying the technical tasks, their interdependencies, risks and opportunities, and providing estimates of needed resources/budgets. This is also the point in the process to determine the need for resources and specialists during the project life cycle to improve efficiency/effectiveness and decrease cost over-runs. For example during product design, various disciplines work together to evaluate parameters such as manufacturability, testability, operability and sustainment against product performance. In some cases, project tasking is concurrent to achieve the best results. Figure 5.2 is the Planning Process Context Diagram.

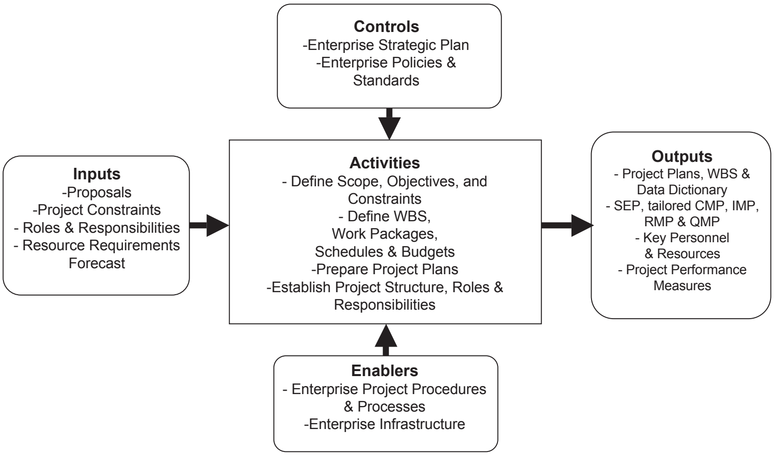


Figure 5-2 Context Diagram for the Project Planning Process

5.2.3 Inputs

The primary inputs to the planning process are the project proposal and technical results from the initial concept exploration stage. Project resources are derived from and require coordination with the enterprise. The enterprise also provides the contextual policies, procedures and standards.

5.2.4 Outputs

Principle outputs from planning are a work breakdown structure (WBS), a data dictionary, project milestones, task descriptions with completion criteria and work authorizations, a detailed budget, a project quality plan, identification of required technical reviews and their completion criteria, methods for controlling changes, risk assessment and methodology, identification of other technical plans and documentation to be produced for the project.

An important outcome of planning is the Systems Engineering Plan. This plan identifies the activities to be accomplished, key events that must be satisfied at decision gates throughout the project, work packages that define the working schedule and the assignment of required resources (people, equipment and facilities) that define the project budget. Each decision gate will have a list of activities or tasks that must be successfully completed prior to entering the decision gate. This plan references other planning instruments that are tailored for use on the project such as the Configuration Management, Risk Management, and Information Management Plans discussed in later sections of this chapter.

5.2.5 Process Activities

Process activities include:

- Analyze project proposal and related agreements to define project scope
- Identify project objectives and project constraints
- Define Key Events that provide structure for the project. Once the key events are established, the tasks and activities that need to be completed prior to each event are identified
- Define top-level work packages for each task and activity identified. Each work package should be tied to resources required including procurement strategies. Develop project schedule based on objectives and work estimates
- Define costs and estimate project budget
- Prepare a Systems Engineering Plan; tailor the quality, configuration, risk and information management plans to meet the needs of the project.
- Tailor the enterprise risk management processes and practices in accordance with the agreements and the Systems Engineering Plan to establish a systematic approach for identifying and handling risk.
- Tailor the enterprise configuration management processes and practices in accordance with the agreements and the Systems Engineering Plan to establish a systematic approach for identifying and handling change requests
- Establish tailoring of enterprise procedures and practices to carry out planned effort. Chapter 10 contains a detailed discussion on tailoring.

■ *Common approaches and tips:*

- Integrated product development teams are used frequently to break down communications and knowledge stovepipes within organizations.[2]

- Creation of the WBS is an activity where systems engineering and project management intersect.[3]
- Skipping or taking shortcuts in the planning process reduces the effectiveness of other project processes.
- Even agile project management methods include planning – the cycles may be shorter and more frequent, but planning is an essential process.
- Incorporate risk assessment early in the planning process to identify areas that need special attention or contingencies. Always attend to the technical risks.
- The Project Management Institute is a source of guidelines for project planning.[4]

5.3 Project Assessment Process

5.3.1 Purpose

This process collects and evaluates the status of the project comparing the results achieved against the plan to assess the maturity of the project. Assessments are scheduled periodically and for all milestones and decision gates. The intention is to maintain good communications within the project team and with the stakeholders, especially when deviations are encountered.

5.3.2 Description

The Project Planning Process identified details of the work effort and expected results. Project assessment collects data to evaluate the adequacy of the project infrastructure, the availability of necessary resources, and compliance with project performance measures. A discussion of the creation and assessment of Technical Performance Measures (TPM) is found in chapter 7. Figure 5-3 is the context diagram for the Project Assessment Process.

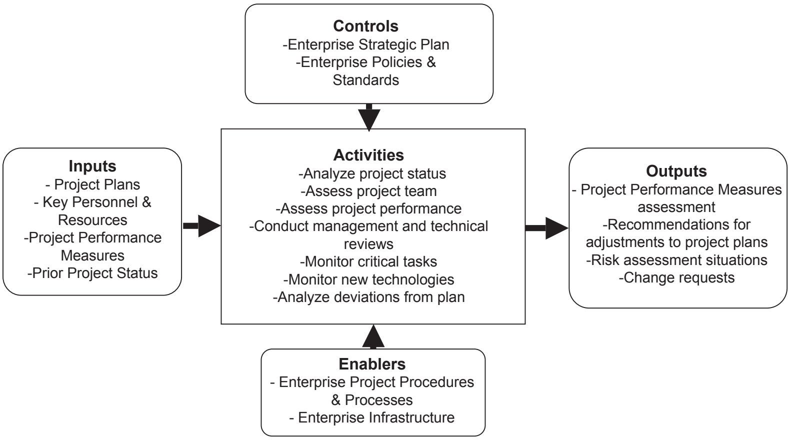


Figure 5-3 Context Diagram for the Project Assessment Process

5.3.3 Inputs

The planning outputs provide the basis for ongoing evaluation of the progress and achievements of a project against performance requirements, costs, schedule, and overall business objectives. The enterprise provides the contextual policies, procedures and standards for project assessment.

5.3.4 Outputs

The outcome of project assessment is information on the health and maturity of the project work effort. Results of this process are used to make decisions regarding future work and technical options.

5.3.5 Process Activities

Activities within this process include:

- Determine actual and projected cost against budget; actual and projected time against schedule and deviations in project quality
- Evaluate the effectiveness of personnel
- Evaluate the adequacy and the availability of the project infrastructure
- Evaluate project progress against established criteria and milestones
- Conduct required reviews, audits, inspections to determine readiness to proceed to next milestone
- Monitor critical tasks and new technologies – see Risk Management section below
- Make recommendations for adjustments to project plans; these are input to the project control process and other decision-making processes
- Communicate status as designated in the agreements, policies and procedures

■ *Common approaches and tips:*

- One way for project management to remain updated on project status is to conduct regular team meetings; short standup meetings on a daily or weekly schedule are effective for smaller groups.
- Prevailing wisdom suggests that "what gets measured gets done," but projects should avoid the collection of metrics that are not used in decision-making and detract from doing real work.
- The Project Management Institute is a source of guidelines for project assessment.

5.4 Project Control Process

5.4.1 Purpose

The Project Control Process uses project assessment to direct the efforts of the project. This includes redirecting the project when deviations from the plan are uncovered, or the project does not reflect the anticipated maturity. (Appendix section G.3 elaborations on Continuous Process Improvement.)

5.4.2 Description

The rigor of the project control process is directly dependent on the complexity of the system-of-interest. Project control involves both corrective and preventive actions taken to ensure that the project is performing according to the plans and schedules, and within projected budgets. Assessments also monitor the technical progress of the project, and may identify new risks or areas that require additional investigation. The Project Control Process may trigger activities within the other process areas in this chapter. See Figure 5-4 for the Project Control Process Context Diagram.

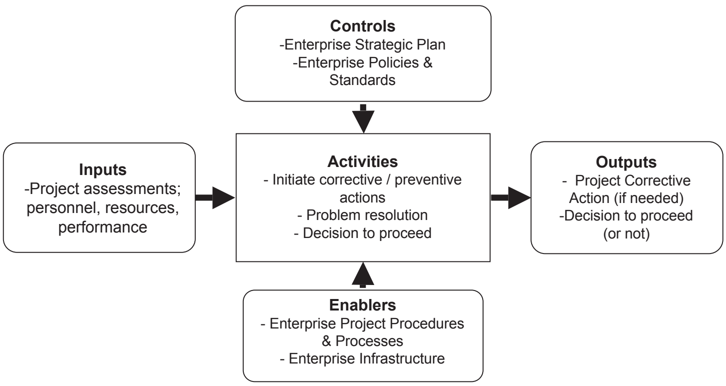


Figure 5-4 Context Diagram for the Control Process

5.4.3 Inputs

Results of the project assessment process are analyzed to determine the maturity of the project and identify any deviations from the plans or technical performance of the product. The project is guided by the enterprise policies, procedures, and standards.

5.4.4 Outputs

New directions are communicated to both project team and customer, when appropriate. If assessments are associated with a decision gate, a decision to proceed, or not to proceed, is taken.

5.4.5 Process Activities

The activities within this process include:

- Analyze assessment results
- Initiate corrective actions when the assessment indicates deviation from approved plan
- Initiate preventive actions when the assessment indicates a trend toward deviation

- Initiate problem resolution when the assessment indicates non-conformance with performance success criteria
- Establish work items and changes to schedule to reflect the actions taken
- Negotiate with suppliers for any goods or services acquired from outside the enterprise
- Make the decision to proceed, or not to proceed, when assessment supports a tollgate or milestone event

■ *Common approaches and tips:*

- Project teams need to identify critical areas and control them through monitoring, risk management or configuration management.
- An effective feedback control process is an essential element to enable the improvement of project performance.
- Agile project management techniques schedule frequent assessments and make project control adjustments on tighter feedback cycles than other plan-drive development models.
- Tailoring of enterprise processes and procedures should not jeopardize any certifications. Processes must be established with effective review, assessment, audit, and upgrade as discussed in chapter 6.

5.5 Decision-Making Process

5.5.1 Purpose

Decisions are made throughout the life cycle of every system whenever alternative courses of action exist. Milestones and decision gates mark the most formal decisions. Less formal decisions require less structure, but documenting all decisions, with their rationale, supports future decision-making. (Appendix L elaborates on decision support.)

5.5.2 Description

As the system progresses from early concept definition throughout sustainment, decisions are needed to direct the focus of all personnel toward the desired result. Every decision involves an analysis of the alternative options, the selection of a course of action, and recording of the eventual decision with supporting documentation.

Decisions come from many sources and range from programmatic to highly technical. Different strategies are appropriate to each category of decision. A more complete discussion of decision gates and decision-making strategies is found in chapter 7. All decisions are taken within the context of an enterprise. Some decisions are made within the context of other processes, for example approval of an engineering change proposal within the Configuration Management Process. See Figure 5-5 for the Decision-making Process Context Diagram.

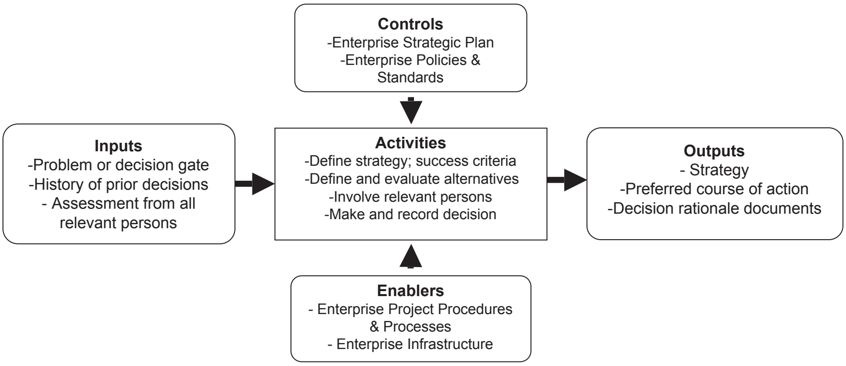


Figure 5-5 Context Diagram for the Decision-making Process

5.5.3 Inputs

Decisions related to decision gates are taken on a pre-arranged schedule. Other requests for a decision may arise from any stakeholder. Decisions should be taken in consideration of prior history and all relevant persons should be involved in the decision-making activities. The enterprise provides the contextual policies, procedures and standards.

5.5.4 Outputs

The approved decision is documented along with rationale, assumptions, constraints and supporting analysis, recorded and communicated.

5.5.5 Process Activities

Decision-making Process activities include:

- Identify the need for a decision and the strategy for making the decision, including desired outcomes and measurable success criteria
- Involve all personnel with knowledge and experience relevant to the decision
- Evaluate the consequences of alternative choices using the selected strategy and optimize the decision
- Record the decision made with the relevant data and supporting documentation
- Communicate new directions from the decision

■ *Common approaches and tips:*

- Decision support systems have been developed to assist decision makers in considering the implications of various courses of action.
- Failure to maintain a history of prior studies and decisions can result in wasted effort when old questions reappear.

5.6 Risk and Opportunity Management Process

5.6.1 Purpose

Risk and opportunity management is a disciplined approach to dealing with uncertainty that is present throughout the entire systems life cycle. The objective is to achieve a proper balance between risk and opportunity. This process is used to understand and avoid the potential cost, schedule, and performance/technical risks to a system, and to take a proactive and structured approach to anticipate negative outcomes, respond to them if they occur; and to identify potential opportunities that may be hidden in the situation. Enterprises manage many forms of risk and the risk associated with system development is managed in a manner that is consistent with the enterprise strategy.

5.6.2 Description

Every new system or modification of an existing system is based on pursuit of an opportunity. Risk always is present in the life cycle of systems, and the risk management actions are assessed in terms of the opportunity being pursued. The system may be intended for technical accomplishments near the limits of the state of the art, creating technical risk. Risk can also be introduced during architectural design caused by the internal interfaces that exits between the system elements. System development may be rushed to deploy the system as soon as possible to exploit a marketing opportunity or meet an imminent threat, leading to schedule risk. All systems are funding-limited so that cost risk is always present. Risk can develop within a project, since (for example) technical risk can create schedule risk, which in turn can create cost risk. Chapter 7 contains a more detailed discussion of these four risk categories; technical, schedule, cost and programmatic.

Ambient risk is often neglected in project management. The ambient risk is defined as the risk caused by and created by the surrounding environment (ambience) of the project[5]. Project participants have no control over ambient risk factors, but can learn to observe the external environment and eventually to take proactive or reactive action to minimize the impact of the environment on the project. The typical issues are time dependent processes, rigid sequence of activities, one dominant path for success and little slack.

Projects are subject to uncertainty; an uncertain event may be harmful if it occurs (threats), another may assist in achieving objectives (opportunities). Dealing with both types of uncertainty under the single heading of "risk management" minimizes process and overhead and expands organizational and personal commitment towards finding and capturing opportunities. Since traditionally, project managers think of risks as threats alone, it may be a change to begin recognizing opportunities. If opportunities are handled along with threats, risk management language needs to be balanced with terms for opportunities such as "exploit", "share", "protect" and "enhance". Project managers may need encouragement to be open to opportunities and to manage both threats and opportunities proactively.

Typical strategies for coping with risk include transference, avoidance, acceptance or taking action to reduce the anticipated negative effects of the situation. Most risk management processes include a prioritization scheme whereby risks with the greatest negative effect and the highest probability of occurrence are handled before those deemed to have lower negative consequences and lower probability of occurrence. The objective of risk management is to balance the allocation of resources such that the minimum amount of resources achieves the greatest risk mitigation (or opportunity realization) benefits. See Figure 5-6 for the Risk Management Process Context Diagram.

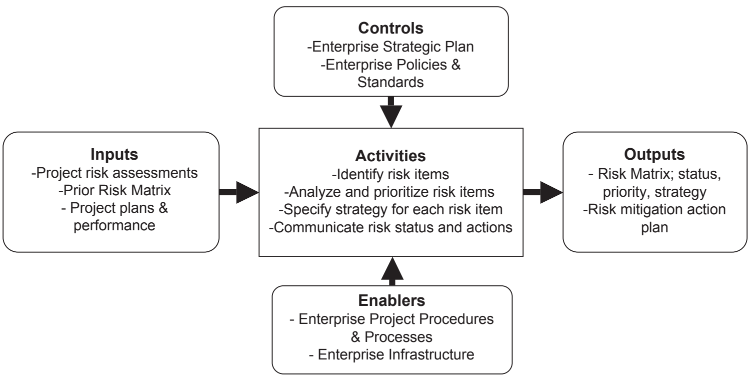


Figure 5-6 Context Diagram for the Risk Management Process

5.6.3 Inputs

In the Project Planning Process, a Risk Management Plan is tailored to satisfy the individual project procedures for risk management. In many cases, risk situations are identified during the project assessment process. A risk management process establishes documentation, often maintained as a risk matrix, which includes a description, priority, mitigation, responsible person, and status of each risk item.

5.6.4 Outputs

The Risk Matrix contains the findings of the Risk Management Process. For selected risks, an action plan is produced to direct the project team to properly respond to the risks. If appropriate, change requests are generated to mitigate technical risk.

5.6.5 Process Activities

Process activities include:

- Identify and define risk situations
- Analyze risks for likelihood and severity in order to determine the magnitude

of the risk and its priority for handling
- Define handling scheme and resources for each risk, including identification of a person who will be responsible for continuous assessment of the status of the situation
- Using the criteria for acceptable and unacceptable risk, generate a plan of action when the risk threshold exceeds acceptable levels
- Maintain a record of risk items and how they were handled
- Maintain transparent risk management communications

■ *Common approaches and tips:*

- One rule of thumb for identifying risks is to pose each risk candidate in a "if <situation>, then <consequence.>" format. This form helps to determine the validity of a risk and to assess its magnitude or importance. If the statement does not make sense or cannot be put in this format, then the candidate is probably not a true risk. For example, a statement that describes a situation but not a consequence implies that the potential event will not affect the project. Similarly, a statement of potential consequence without a clear situation description is worthy of more attention.
- Document everything so if there are unforeseen issues and challenges during execution you can recreate the environment within which the planning decisions were made and know where to update the information to correct the problem.
- Negative feedback toward personnel who identify a potential problem will discourage the full cooperation of engaged stakeholders, and could result in failure to address serious risk-laden situations. Conduct a transparent risk management process to encourage suppliers and other stakeholders to assist in the risk mitigation efforts. Some situations can be difficult to categorize vis-à-vis probability and consequences – involve all relevant stakeholders in this evaluation to capture the maximum variety in viewpoints.
- Many analysis completed throughout the technical processes, such as FMECA, may identify candidate risk elements.
- The metrics for risk management vary by organization and by project. As with any metric, use measurements or statistics that help manage the risk.
- The Project Management Institute is a good source for more information on Risk Management.
- The Institute of Risk Management has generated *The Risk Management Standard*.[6]
- See Forsberg, et. al. for additional reading on opportunity management.[7]

5.7 Configuration Management Process

5.7.1 Purpose

The objective of configuration management is to ensure effective management of the evolving configuration of a system, both hardware and software, during its life cycle. Fundamental to this objective is the establishment, control, and maintenance of software and hardware baselines. Baselines are reference points for maintaining development and control. These baselines, or reference points, are established by review and acceptance of requirements, design, and product specification documents. The creation of a baseline may coincide with a project milestone or decision gate. As the system matures and moves through the life cycle stages the software or hardware baseline is maintained under configuration control. (Appendix G.4 elaborates on Configuration Management.)

5.7.2 Description

Configuration Management ensures that product functional, performance, and physical characteristics are properly identified, documented, validated, and verified (establishing product integrity); that changes to these product characteristics are properly identified, reviewed, approved, documented, and implemented; and that the products produced against a given set of documentation are known. See chapter 8 for additional discussion relevant to configuration management.

Evolving system requirements are a reality that must be addressed over the life of a system development effort, and throughout the Utilization and Support Stages of the system. See Figure 5-7 for the Configuration Management Process context diagram.

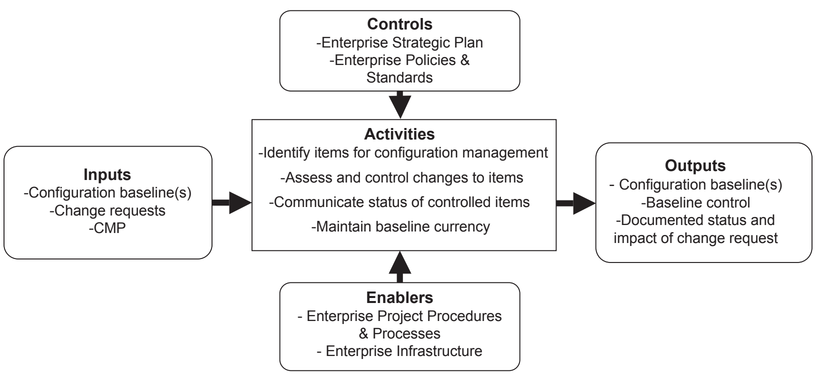


Figure 5-7 Context Diagram for the Configuration Management Process

5.7.3 Inputs

In the Project Planning Process, a Configuration Management Plan is tailored to satisfy the individual project procedures for configuration management. In many cases, the need for change requests is identified during the project assessment process.

5.7.4 Outputs

The primary output of the configuration management process is the maintenance of the configuration baseline for the system and system elements. Items are placed under formal control as part of the decision-making process. The required configuration baseline documentation is developed and approved in a timely manner to support required systems engineering technical reviews, the system's acquisition and support strategies, and production. This documentation is maintained throughout the life of the system. The configuration management process formally documents the impact to any process, organizations, decisions, products, and services affected by a given change request.

5.7.5 Process Activities

Process activities include:

- Identify system elements that are maintained under configuration control
- Maintain enough information to ensure system/product integrity
- Implement a configuration control cycle that incorporates evaluation, approval, validation, and verification of engineering change requests
- Develop and maintain configuration control documentation
- Perform Configuration Audits associated with milestones and decision gates to validate the baselines

■ *Common approaches and tips:*

- Establish a configuration control board with representation from all stakeholders and engineering disciplines participating on the project – see chapter 8 for more details.
- Begin the configuration management process in the infancy phases of the system and continue through until disposal of the system.

5.8 Information Management Process

5.8.1 Purpose

Information Management ensures that information is properly stored, maintained, secured, and accessible to those who need it thereby establishing/maintaining integrity of relevant system life cycle artifacts.

5.8.2 Description

Information exists in many forms, and different types of information have different value within an enterprise. Information assets, whether tangible or intangible, have become so pervasive in contemporary organizations that they are indispensable. The impact of threats to secure access, confidentiality, integrity, and availability of information can cripple the ability to get work done. As information systems become increasingly interconnected, the opportunities for compromise increase.[8] The following are important terms in Information Management:

- Information is what an organization has compiled or its employees know. It can be stored and communicated, and it might include customer information, proprietary information, and/or protected (e.g. by copyright, trademark, or patent) and unprotected (e.g. business intelligence) Intellectual Property.
- Information assets are intangible information and any tangible form of its representation, including drawings, memos, e-mail, computer files, and databases.
- Information security generally refers to the confidentiality, integrity, and availability of the information assets.
- Information security management includes the controls used to achieve information security and is accomplished by implementing a suitable set of controls, which could be policies, practices, procedures, organizational structures, and software functions.
- Information Security Management System is the life cycle approach to implementing, maintaining, and improving the interrelated set of policies, controls, and procedures that ensure the security of an organization's information assets in a manner appropriate for its strategic objectives.

Information management provides the basis for the management of and access to information throughout the system life cycle, including after disposal if required. Designated information may include enterprise, project, agreement, technical, and user information. The mechanisms for maintaining historical knowledge in the prior processes – decision-making, risk and configuration management – are under the responsibility of information management. See Figure 5-8 for the Information Management Process Context Diagram.

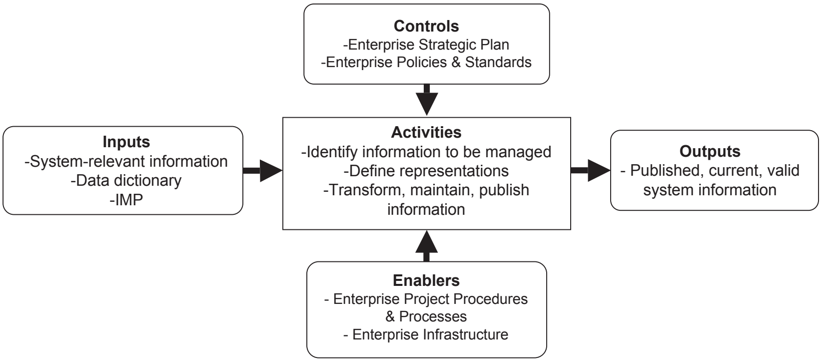


Figure 5-8 Context Diagram for the Information Management Process

5.8.1 Inputs

In the Project Planning Process, an Information Management Plan is tailored to satisfy the individual project procedures for information management. An information management plan identifies the system-relevant information to be collected, retained, secured, and disseminated, with a schedule for retirement.

5.8.2 Outputs

The output of this process is the availability for use and communication of all relevant systems artifacts in a timely, complete, valid and, if required, confidential manner.

5.8.3 Process Activities

Process activities include:

- Establish/maintain a system data dictionary – see project planning outputs
- Define system-relevant information, the storage requirements, access privileges and the duration of maintenance
- Define formats and media for capture, retention, transmission, and retrieval of information
- Identify valid sources of information and periodically obtain artifacts of information
- Maintain information according to security and privacy requirements
- Retrieve and distribute information as required
- Archive designated information for compliance with legal, audit and knowledge retention requirements
- Retire unwanted, invalid or unverifiable information according to organizational policy, security, and privacy requirements

■ *Common approaches and tips:*

- Identify information-rich artifacts and store them for later use even if the information is informal such as a design engineer's notebook.
- Information management delivers value to the organization and the project by using a variety of mechanisms to provide access to the contents of data repositories. Email, web-based access through intranets, and database queries are a few examples.
- ISO 17799 "Code of Practice for Information Security Management" is an international standard that provides a best practices framework for implementing security controls.
- ISO 10303, STandard for the Exchange of Product model data (STEP) includes Application Protocol (AP) 239 Product Lifecycle Support (PLCS), which addresses information requirements for complex systems; this topic is discussed in chapter 8.[9]

1 Kossiakoff, Alexander and William N. Sweet (2003). **Systems Engineering; Principles and Practice**, NJ: Hoboken: John Wiley & Sons, Inc., page 91.
2 Martin, J. N., *Systems Engineering Guidebook*, CRC Press, 1996.
3 Forsberg, K., et. al., *Visualizing Project Management, 3 ed.* John Wiley & Sons, 2005, p. 198-219.
4 see URL: www.pmi.org
5 Fossnes, T., (2005), Lessons from Mt. Everest Applicable to Project Leadership. Proceedings of the 15th Annual INCOSE International Symposium, Rochester, NY.
6 See URL: www.theirm.org
7 Forsberg, K., et. al., *Visualizing Project Management, 3 ed.* John Wiley & Sons, 2005, p. 223-253.
8 STSC Crosstalk, http://www.stsc.hill.af.mil/crosstalk/2003/05/brykczynski.html
9 see URL: www.plcs-resources.org

6 Enterprise and Agreement Processes

6.1 Introduction

Enterprise processes are the purview of the organization and are used to direct, enable, control, and support the system life cycle. This chapter and the ISO/IEC 15288 focus on the capabilities of an organization relevant to the realization of a system; they are not intended to address general business management objectives, although sometimes the two overlap.

Within the enterprise, organizational units cooperate to develop, implement, deploy, operate, maintain and dispose of the system-of-interest. Enabling systems may also need to be modified to meet the needs of new systems; developed or acquired if they do not exist. Examples include development, manufacturing, training, testing, transport, maintenance and disposal systems that support the system-of-interest.

Every enterprise claims interfaces with industry, academia, customers, partners, etc. An overall objective of enterprise processes is to identify these external interfaces and establish the parameters of these relationships, including identifying the inputs required from the external entities and the outputs that will be provided to them. This network of relationships provides the context of the business environment of the enterprise and access to future trends and research. Some relationships are defined by the exchange of products or services.

There are six Enterprise Processes identified by ISO/IEC 15288. They are: Enterprise Environment Management, Investment Management, System Life Cycle Processes Management, Resource Management, and Quality Management. Discussion of these processes and their interfaces is the topic of this chapter. The enterprise will tailor these processes and their interfaces to meet specific strategic and communications objectives.

There are two Agreement Processes identified by ISO/IEC 15288: the Acquisition Process, and the Supply Process. These processes are discussed in the closing sections of this chapter. They are included in this chapter because they conduct the essential business of the enterprise, namely the buying and selling of products and services. They establish the relationships between enterprises relevant to the acquisition and supply of products and services. Agreements may exist between organizational units internal or external to the enterprise. Figure 6-1 illustrates the critical success factors for enterprise and agreement processes.[1]

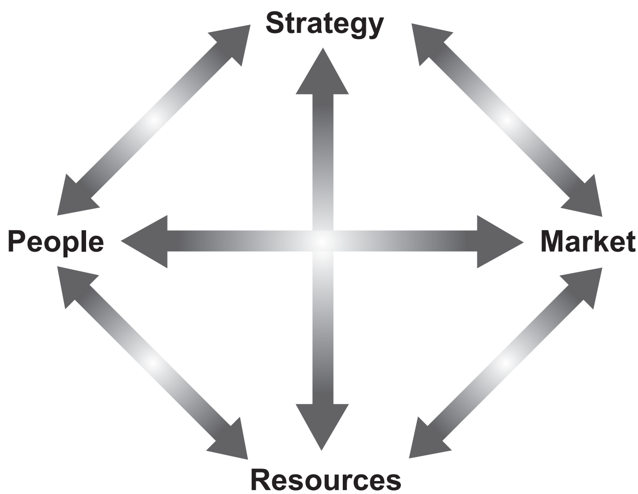


Figure 6-1 Key Success Factors for Enterprise Processes

6.2 Enterprise Environment Management Process

6.2.1 Purpose

The purpose of the Enterprise Environment Management Processes is to establish and maintain a set of policies and procedures at the enterprise level that support the organization's ability to acquire and supply products and services.

6.2.2 Description

The work of the organization is accomplished through projects. These projects are conducted within the context of the enterprise – the enterprise environment. This environment needs to be defined and understood within the organizations and the project to ensure alignment of the working units and achievement of overall enterprise strategic objectives. This process exists to establish, communicate, and continuously improve the system life cycle (SLC) process environment. Figure 6-2 contains the context diagram for the Enterprise Environment Management Process.

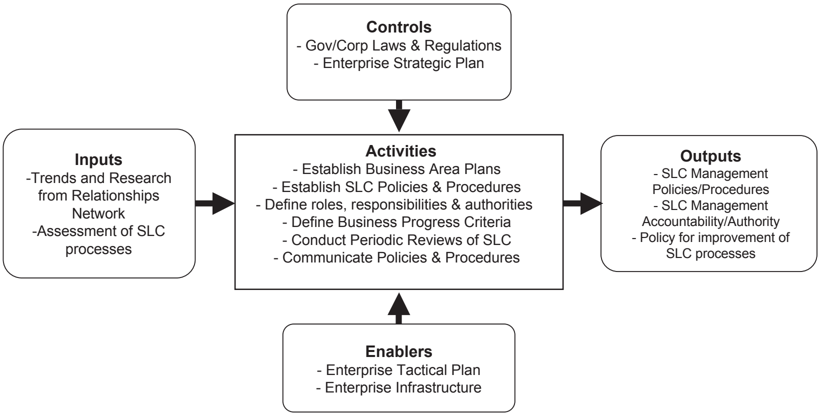


Figure 6-2 Enterprise Environment Management Process Context Diagram

6.2.3 Inputs

The network of relationships include: Government, Industry, and Academia. Each of these external interfaces provide unique and essential information for the enterprise to succeed in business and meet the continued need and demand for improved and effective systems and products for the customers. It is up to the enterprise environment management process to fully define and utilize these external entities and interfaces, i.e. their value, importance and capabilities that are required by the enterprise.

- Legislative, regulatory, and other government requirements
- Industry systems engineering and management related standards, training, capability maturity models
- Academic education, research results, future concepts and perspectives, requests for financial support

The enterprise environment is built on the existing enterprise infrastructure, including facilities, personnel and knowledge. Integration and interoperability of supporting systems, such as financial, human resources, and training, is critically important to execute the enterprise strategic objectives. Feedback from active projects is used to refine and continuously improve the enterprise environment.

6.2.4 Outputs

The essential products from this process support the systems life cycle (SLC) processes in the form of policies, procedures, guidance, and directions needed for the enterprise to provide the most cost effective environment for a portfolio of systems over their life cycle. This includes:

- SLC strategic plans, policies, procedures, directives, guidance – for example, templates for management plans such as configuration, information, and risk management plans
- Accountability & authority for all systems engineering & management processes within the enterprise
- Roles & Responsibilities defined for systems engineering & management processes within the enterprise
- Review criteria for process improvement (including assessments, approvals/disapprovals, recommendations)

6.2.5 Process Activities

This process includes the following activities:

- Establish Business Area plans – use the strategic objectives to identify candidate projects to fulfill them
- Establish SLC policies, processes, procedures – consistent with the enterprise and business area plans
- Define Roles, Responsibilities & Authorities – align the portfolio of projects
- Define the decision-making criteria that determine entering and exiting each stage of the SLC – expressed in terms of business achievements
- Conduct periodic reviews of the SLC models used by projects – use assessments to confirm the adequacy and effectiveness of the SLC processes
- Disseminate policies and procedures throughout the enterprise.

■ *Common approaches and tips:*

- The process of developing the business area plans helps the enterprise to assess where it needs to focus activities and resources to meet present and future strategic objectives. Include representatives from relevant stakeholders in the enterprise community.
- Manage the network of external relationships – assign personnel to identify standards, industry & academia research and other sources of enterprise management information and concepts needed by the enterprise.
- Establish an enterprise architecture – integrating the infrastructure of the enterprise can make the execution of routine business activities more efficient.
- Establish an enterprise communication plan – most of the processes in this handbook include dissemination activities. An effective set of communication methods is needed to ensure that all stakeholders are well-informed.
- Base the policies and procedures on an Enterprise level strategic plan that provides a comprehensive understanding of the Enterprise's goals, objectives, stakeholders, competitors, future business, and technology trends.
- Work continually to improve the SLC processes.

6.3 Investment Management Process

6.3.1 Purpose

The purpose of the Investment Management Process is to initiate and sustain investments in projects that meet the objectives of the organization and to cancel or redirect investments for projects that do not.

6.3.2 Description

Projects create the products or services that generate income for an enterprise. The conduct of successful projects requires an adequate allocation of funding and resources and the authority to deploy them to meet project objectives. Most business entities mange the commitment of financial resources using well defined and closely monitored processes.

The Investment Management Process also performs ongoing evaluation of the projects in its portfolio. Based on periodic assessments, projects are determined to justify continued investment if they

- are progressing toward achieving established goals;
- are complying with project directives from the enterprise;
- are conducted according to approved plans; and,
- are providing a service or product that is still needed and providing acceptable investment returns.

Otherwise, projects may be redirected, or in extreme instances, cancelled. Figure 6-3 shows the context diagram for the Investment Management Process.

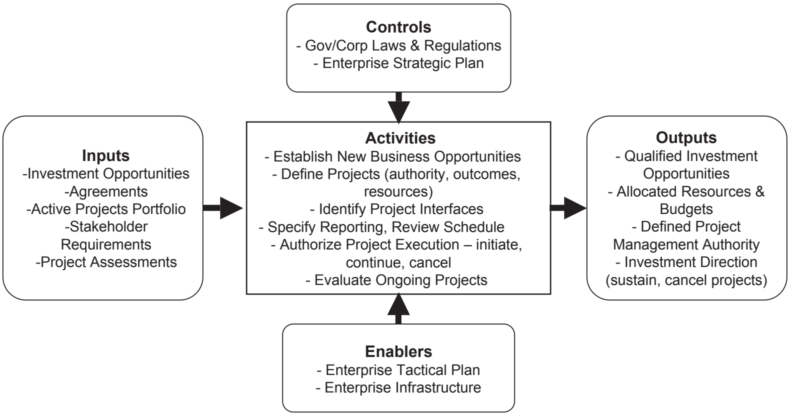


Figure 6-3 Investment Management Process Context Diagram

6.3.3 Inputs

Investment opportunities are not all equal. Enterprises are limited in the number of projects that can be conducted concurrently. Some investments are not well aligned with the overall strategic plan of the enterprise. For these reasons, opportunities are evaluated against the portfolio of existing agreements and ongoing projects, and taking into consideration the attainability of the stakeholders' requirements.

The enterprise investment decisions are built on the existing enterprise infrastructure, including facilities, personnel and knowledge. Efficient use of these resources is achieved by exploiting opportunities to share enabling systems or to use a common system element on more than one project. These opportunities are enabled by good communications within the enterprise infrastructure.

6.3.4 Outputs

The successful implementation of the Investment Management Process results in the following outputs:

- Qualified investment opportunities result in the initiation of new project(s)
- Resources and budgets are identified and allocated to the projects
- Project management accountability and authorities are defined
- Projects meeting assessment criteria are sustained
- Project not meeting assessment criteria are redirected or terminated

6.3.5 Process Activities

This process includes the following activities:

- Identify and assess investment opportunities consistent with the enterprise strategic plan.
- Establish project scope; define project management accountabilities and authorities, identify expected project outcomes.
- Allocate adequate funding and other resources.
- Identify opportunities for multi-project synergies.
- Specify project reporting requirements and review schedule that govern the progress of the project.
- Authorize project execution – initiate, continue, cancel.
- Evaluate ongoing projects to provide rationale for continuation, redirection or termination.

■ *Common approaches and tips:*

- When investment opportunities present themselves, prioritize them based on measurable criteria such that projects can be objectively evaluated against a threshold of acceptable performance.

- The expected project outcomes should be based on clearly defined criteria that are measurable to ensure than an objective assessment of progress can be determined. Specify the investment information that will be assessed for each milestone. Initiation should be a formal milestone that does not occur until all resources are in place as identified in the project plan.
- Establish a program office or other coordination organization to manage the synergies between active projects in the enterprise portfolio. Complex and large enterprise architectures require the management and coordination of multiple interfaces and make additional demands on investment decisions. These interactions occur within and between the projects.
- Include risk assessments in the evaluation of ongoing projects. Projects that contain risks that may pose a challenge in the future might require redirection. Cancel or suspend projects whose disadvantages or risks to the organization outweigh the investment.
- Include opportunity assessments in the evaluation of ongoing projects. Addressing project challenges may represent a positive investment opportunity for the enterprise. Avoid pursuing opportunities that are inconsistent with the capabilities of the organization and its strategic goals and objectives or contain unacceptably high technical risk, resource demands, or uncertainty.
- Allocate resources based on the requirements of the projects; otherwise, the risk of cost and schedule overruns may have a negative impact on quality and performance of the project.
- Establish effective governance processes that directly support investment decision-making and communications with project management.

6.4 System Life Cycle Processes Management Process

6.4.1 Purpose

The purpose of the System Life Cycle Process Management Process is to establish a set of proven and effective system life cycle processes and make them available for use by the enterprise.

6.4.2 Description

This process provides integrated, system life cycle (SLC) processes necessary to meet the enterprise's strategic plans, policies, goals and objectives for all projects and all system life cycle stages. The processes are defined, adapted and maintained to support the requirements of the enterprise, systems engineering organizational units, individual projects and personnel. SLC processes are supplemented by recommended methods and tools. The resulting guidelines in the form of enterprise policies and procedures are still subject to tailoring by projects, as discussed in chapter 10. Figure 6-4 is the context diagram for this process.

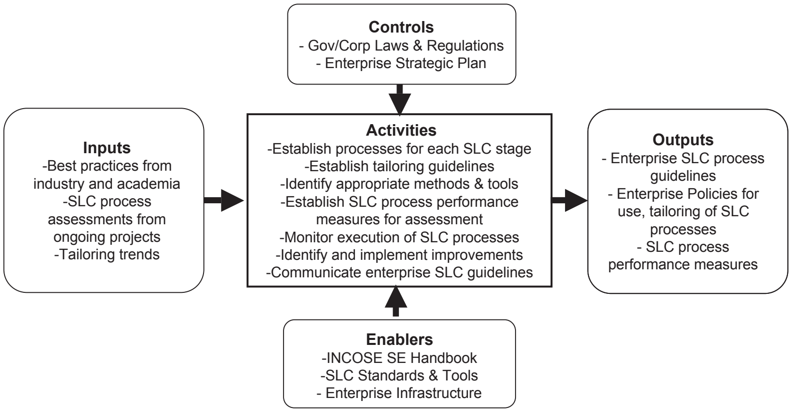


Figure 6-4 SLC Processes Management Process Context Diagram

6.4.3 Inputs

This handbook and relevant standards, new knowledge from research, and industry sponsored knowledge networks are examples of the sources from which the SLC processes are extracted. Enterprise strategy plans and infrastructure are used to ensure consistency in the eventual recommendations. Assessments from projects and trends collected from tailoring processes provide constructive input for improvements to an enterprise's SLC implementation.

6.4.4 Outputs

The successful implementation of this process results in the following outputs:

- Enterprise SLC process guidelines in the form of enterprise policies and procedures
- Enterprise policies and procedures for applying the SLC processes and adapting them to meet the needs of individual projects
- Performance criteria and measures that indicated the degree to which SLC processes are used
- Enterprise SLC Processes Improvement Plan

6.4.5 Process Activities

The process activities include:

- Identify sources (enterprise, corporate, industry, academia, stakeholders and customers) of SLC process information.
- Distill the information from multiple sources into an appropriate set of SLC processes that are aligned with the enterprise plans and infrastructure.

- Establish SLC Process Guideline in the form of Plans, Policies, Procedures, Tailoring Guidance, Models, and Methods and Tools for controlling and directing the SLC processes.
- Define, integrate, and communicate SLC process roles, responsibilities, authorities, requirements, measures, and performance criteria based on the SLC process guidelines.
- Identify opportunities to improve the enterprise SLC guidelines on a continuing basis based on individual project assessments, individual feedback, and changes in the enterprise strategic plan.
- Communicate with all relevant organizations regarding the creation of and changes in the SLC guideline.

■ *Common approaches and tips:*

- Development of an SLC Processes intranet and information database with essential information provides an effective mechanism for disseminating consistent guidelines, providing announcements about enterprise related topics as well as industry trends, research findings, and other relevant information. This provides a single point of contact for continuous communication regarding the SLC Guidelines and encourages the collection of valuable feedback and the identification of enterprise trends.
- Establish an enterprise center of excellence for SLC Processes. This organization can become the focal point for the collection of relevant information, dissemination of guidelines, and analysis of assessments and feedback. They develop checklists and other templates to support project assessments to ensure that the pre-defined measures and criteria are used for evaluation.
- Methods and Tools for enabling the application of SLC Processes must be effective and tailored to the implementation approach of the enterprise and its projects. Create a responsible organization to coordinate the identification and development of partnerships and/or relationships with tool vendors and working groups. They can recommend the use of methods and tools that are intended for the purpose to help personnel avoid confusion and frustration, and wasting valuable time and money. These experts may establish an integrated tool environment between interacting tools to avoid cumbersome (and inaccurate) data transfer – see chapter 8.4.
- Including stakeholders, such as engineering and project management organizations, as participants in developing the SLC guidelines increases their commitment to the recommendations and incorporates a valuable source of enterprise experience.
- Develop alternative SLC guidelines based on the type and scope, complexity, and risk of a project decreases the need for tailoring by engineering and project organizations.
- Provide clear guidelines for tailoring and adaptation.
- Estimating techniques for life cycle cost are described in chapter 8.5.

6.5 Resource Management Process

6.5.1 Purpose

The purpose of the Resource Management Process is to create and maintain a pool of resources for projects.

6.5.2 Description

Projects all need resources to meet their objectives. Financial resources are addressed under the Investment Management Process, but all other resources are provided under this process. Resource is a generic term for all materials, services, facilities, and personnel needed by a project. Non-human resource services include tools, databases, communication systems, financial systems and information technology.

Project planners determine the resources needed by the project. They attempt to anticipate both current and future needs. The Resource Management Process provides the mechanisms whereby the enterprise infrastructure is made aware of project needs and the resources are scheduled to be in place when requested. While this can be simply stated, it is less simply executed. Conflicts must be resolved, personnel must be trained, employees are entitled to vacations and time away from the job, equipment must be purchased and sometimes repaired, buildings are refurbished, and information technology services are in a state of constant change.

The enterprise resource management organization collects the needs, negotiates to remove conflicts, and is responsible for providing the enabling enterprise infrastructure without which nothing else can be accomplished. Since resources are not free, their costs are also factored into investment decisions. Figure 6-5 is the context diagram for the Resources Management Process.

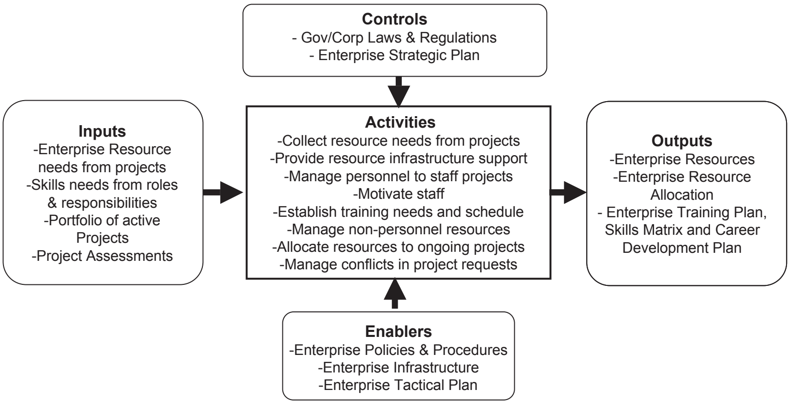


Figure 6-5 Resource Management Process Context Diagram

6.5.3 Inputs

Resource management collects the needs of all the projects in the active portfolio. Personnel needs are evaluated against available people with the pre-requisite skills to determine if training or hiring activities are indicated. Other assets are scheduled or if necessary acquired. Trends in the market may suggest changes in the composition of project teams and the supporting IT environment. The availability and suitability of the enterprise infrastructure is one of the critical project assessments and provides feedback for improvement – and reward – mechanisms. All enterprise processes require mandatory compliance with government and corporate laws and regulations. Decision-making is governed by the enterprise strategic plan.

6.5.4 Outputs

The primary objective of this process is to provide a resource pool to the enterprise. This is complicated by the number of sources for requests, the need to balance the skills of the labor pool against the other infrastructure elements (e.g. computer-based tools), the need to maintain a balance between the budgets of individual projects and the cost of resources, the need to keep apprised of new or modified policies and procedures that might influence the skills inventory, and myriad unknowns.

Resources are allocated based on requests and conflicts are negotiated. The goal is to provide personnel, materials and services to a project when they are needed to keep the project on target and on budget. A key concern is keeping project personnel from becoming over-committed, especially persons with specialized skills. Skills inventory and career development plans are important documentation that can be validated by engineering and project management.

6.5.5 Processes Activities

This process includes the following activities:

- Manage resource availability to ensure enterprise goals and objectives are met. Conflicts and resource shortfalls are managed with steps for resolution.
- Provide resources to support all projects – the enterprise infrastructure.
- Keep employees motivated, content in their career progression, current with their training, and appropriately allocated using techniques that are within acceptable enterprise and corporate guidelines and constraints.
- Control multi-project resource management communications to effectively allocate resources throughout the enterprise, identify potential future or existing conflict issues and problems with recommendations for resolution.

■ *Common approaches and tips:*

- Motivate staff; consider using an integrated product development team environment as a means to reduce the frequency of project rotation; recognize progress and accomplishments and reward success; establish apprentice and mentoring programs for newly hired employees and students.

- Maintain a pipeline of qualified candidates that are interested in joining the organization as employees or temporary staff. Focus recruitment, training and retention efforts on personnel with experience levels, skills and subject matter expertise demanded by the projects. Personnel assessments should review proficiency, motivation, ability to work in a team environment, as well as the need to be retrained, reassigned or relocated.
- Qualified personnel and other resources may be hired temporarily/leased – insourced or outsourced – in accordance with the investment strategy.
- Encourage personnel to engage in external networks as a means of keeping abreast of new ideas and attracting new talent to the organization.
- Maintain an enterprise career development program that is not sidetracked by project demands. Develop a policy that all personnel receive training or educational benefits on a regular cycle. This includes both undergraduate and graduate studies, in-house training courses, certifications, tutorials, workshops, and conferences.
- Remember to provide training on enterprise policies and procedures and SLC processes.
- Establish a resource management information infrastructure with enabling support systems and services to maintain, track, allocate and improve the resources for present and future enterprise needs. Computer-based human resource, equipment tracking, facilities allocation and other systems are recommended for organizations over 50 people.
- Attend to physical factors, including facilities and human factors, such as ambient noise level and computer access to specific tools and applications.
- Begin planning in early life cycle stages for utilization and support resource requirements for system transition; manpower, facilities, infrastructure, information/data storage and management.
- Use the slack time in the beginning of a project to obtain and train the necessary people to avoid a shortfall of skilled engineers, technologists, managers, and operations experts.

6.6 Quality Management Process

6.6.1 Purpose

The purpose of the Quality Management Process is to make visible the goals of the enterprise toward customer satisfaction. Enterprise policies and procedures govern the products, services, and implementations of the system life cycle (SLC) processes to assure that they meet quality objectives and customer requirements.

6.6.2 Description

Since primary drivers in any project are time, cost, and quality, inclusion of a Quality Management Process is essential to every organization. Many of the SLC processes are concerned with quality issues, and this forms some of the justification for exerting time, money, and energy into establishing these processes in the organization. Application of this handbook is one approach toward inserting a quality discipline into an organization.

The Quality Management Process establishes, implements, and continuously improves the focus on customer satisfaction and enterprise goals and objectives. There is a cost to managing quality as well as a benefit. The effort and time required to manage quality should not exceed the overall value gained from the process. Chapter 8 contains additional discussion of the importance to the organization, and activities for implementing this process. Figure 6-6 is the context diagram for the Quality Management Process.

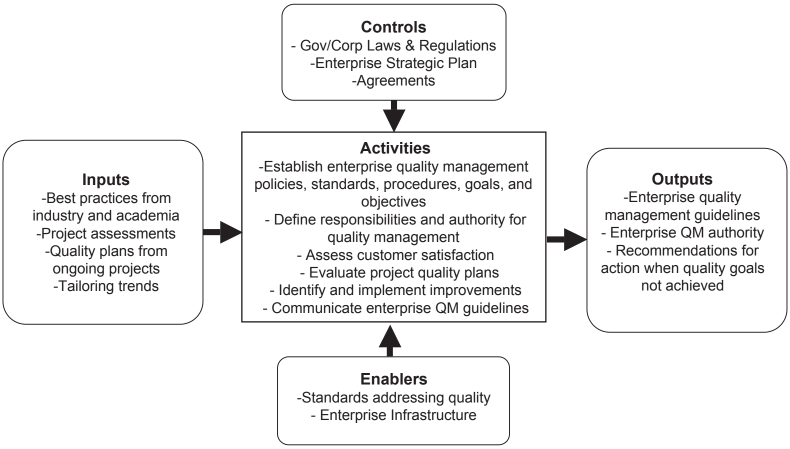


Figure 6-6 Quality Management Process Context Diagram

6.6.3 Inputs

Enterprise strategic documentation including quality policy, mission, strategies, goals and objectives are essential inputs for analysis and synthesis of quality impacts, requirements and solutions. Existing agreements also provide direction regarding the appropriate level of attention given to quality within the organization.

Project assessments include measurements that can be evaluated to determine the performance of a project team and the progress toward a quality outcome. Trends in tailoring of project-specific quality plans provide clear indications of potential improvements in the overall enterprise guidelines.

The team of people working in this process will also find a wealth of material in ISO standards and other sources.

6.6.4 Outputs

The successful implementation of the Quality Management Process results in the following outcomes:

- Enterprise Quality Management Guidelines include the set of policies and procedures that apply to quality practices within the organization, within individual projects, and as part of the execution of SLC processes. These guidelines define goals and quality objectives for processes and systems that are measurable and objective.
- Accountability and authority for quality management is assigned within the organization and realistic resources are provided.
- Customer satisfaction is closely monitored and appropriate actions are taken when quality goals are not achieved.

6.6.5 Process Activities

This process includes the following activities:

- Establish Quality Management Guidelines – Policies, Standards, and Procedures
- Establish enterprise and project quality management goals & objectives
- Define Responsibilities & Authorities
- Assess Customer Satisfaction against compliance with requirements and objectives
- Evaluate project assessments and recommend appropriate action when indicated
- Continuously improve the Quality Management Guidelines
- Maintain open communications within the organization

■ *Common approaches and tips:*

- Management commitment to quality is reflected in the strategic planning of the enterprise – the rest of the organization will follow. Everyone in the Enterprise should know the Enterprise's quality policy.
- Quality is a daily focus – not an afterthought!
- Development of a Quality Management intranet and information database with essential information provides an effective mechanism for disseminating consistent guidelines, providing announcements about enterprise related topics as well as industry trends, research findings, and other relevant information. This provides a single point of contact for continuous communication regarding the Quality Management Guidelines and encourages the collection of valuable feedback and the identification of enterprise trends.
- Analyze statistics from process audits, tests and evaluations, product discrepancy

reports, customer satisfaction monitoring, accident and incident reporting, and the implementation of changes to items of a product (e.g. recalled product and/or production lines).

- Quality Management is big business. A plethora of standards, methods, and techniques exist to help an organization. A short list includes the ISO 9000 series, Total Quality Management (TQM), and Six-Sigma (statistical process control). Quality according to ISO 9000 is the "Ability of a set of inherent characteristics of a product, system, or process to fulfill requirements of customers and other interested parties." [2]
- A successful strategy is to aim at achieving customer satisfaction primarily by preventing non-fulfillment of requirements. Ideally, customer satisfaction is linked to compliance with requirements – two signals that the process is not working are situations where the project is compliant but the customer is unhappy, or the project is not compliant and the customer is happy.
- The consistent involvement and commitment of top management with timely decision-making is mandatory for the quality program. This is reflected in staffing and training of project auditors.

6.7 Acquisition Process

6.7.1 Purpose

The Acquisition Process is invoked to establish an agreement between two enterprises under which one party acquires products or services from the other. The acquirer experiences a need for an operational system, for services in support of an operational system, for elements of a system being developed by a project, or for services in support of project activities. The goal is to find a supplier that can meet that need. (Appendix F elaborates on the Acquisition and supply.)

6.7.2 Description

The role of the acquirer demands familiarity with the Enterprise, Project, and Technical Processes as it is through them that the supplier will execute the agreement. An acquirer enterprise applies due diligence in the selection of a supplier to avoid costly failures and impacts to the enterprise budgets and schedules. This section is written from the perspective of the acquirer enterprise. Figure 6-7 is the context diagram for the Acquisition Process.

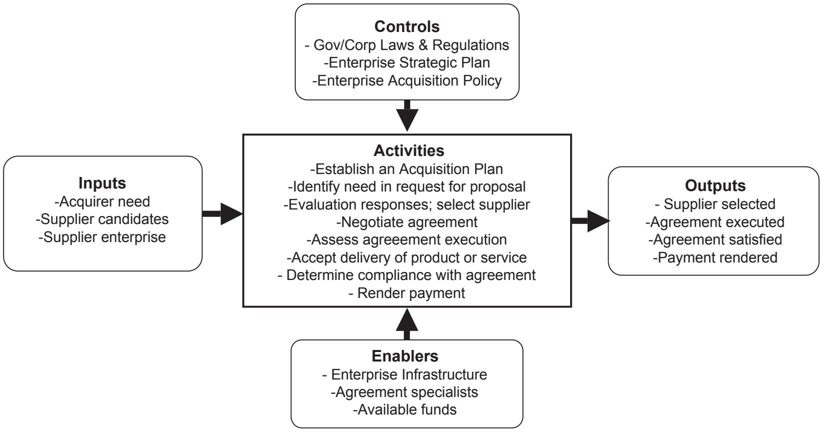


Figure 6-7 Acquisition Process Context Diagram

6.7.3 Inputs

The Acquisition Process begins with the identification of a need that can not be met within the organization encountering the need, or a need that can be met in a more economical way by a supplier. The organization identifies candidate suppliers that could meet this need, and the acquisition personnel identify a plan for procuring the system-of-interest. During creation of a request for proposal, inputs are received from the project management and engineering personnel in the organization with the need. These same personnel participate in the evaluation of the responses and offer recommendations for the selection of supplier. The selection criteria documented in the acquisition plan are used to drive this decision.

Legally executed agreements must comply with government and other directives. The acquirer enterprise will adopt standard practices related to the negotiation of agreements. It is important to note that availability of adequate funding is essential to beginning the acquisition process.

6.7.4 Outputs

The following are outcomes of the Acquisition Process.

- Acquisition Plan – include objective selection criteria and a schedule of milestones
- List potential suppliers – suppliers may be internal or external to the acquirer enterprise
- Recommendations from evaluation of responses to request for proposal – formal documentation, or less formal inter-organizational interactions, e.g. between design engineering and marketing

- Supplier selected – using selection criteria; rank suppliers by their suitability to meet the overall need; establish supplier preferences and corresponding justifications
- Negotiated agreement – formal contracts, or less formal inter-organizational work orders
- System-of-interest (product or service) delivered according to delivery conditions of agreement
- Acquirer payments or other compensations rendered
- Responsibility for system-of-interest transferred from supplier to acquirer
- Communication between acquirer and supplier
- Acquisition strategies regarding agreements

6.7.5 Process Activities

The following activities take place under the Acquisition Process:

- Manage Acquisition Process activities –decision-making for agreements, relationship building and maintenance, interaction with Enterprise management, responsibility for the development of plans and schedules, final approval authority for deliveries accepted from supplier. When an acquisition process cycle concludes, conduct a final review of performance to extract lessons-learned for continued process performance.
- Develop and maintain Acquisition Plans, Strategies, Policies, Procedures to meet the enterprise goals and objectives and the needs of the project management and technical systems engineering organizations.
- Identify needs in Request for Proposal (RFP)
- Select appropriate suppliers – willing to conduct ethical negotiations, able to meet technical obligations, willing to maintain open communications throughout the acquisition process.
- Evaluate supplier responses to request for proposal – system-of-interest meets acquirer needs and complies with industry and other standards. Assessments from the Investment and Quality Management Processes, and recommendations from the requesting organization are necessary to determine the suitability of each response and the ability of the supplier to meet the stated commitments.
- Select the preferred supplier based on acquisition criteria.
- Negotiate agreement – acquirer commits to specify requirements for system-of-interest, participate in verification, validation and acceptance activities, render payment according to the schedule, participate in exception and change control procedures, and contribute to transparent risk management procedures. The agreement will establish criteria for assessing progress toward final delivery.
- Maintain communications with supplier, stakeholders, and other organizations regarding the project.
- Assess execution of agreements to identify risks and issues, progress towards mitigation of risks, adequacy of progress toward delivery, evaluation cost and schedule performance, and determine potential undesirable outcomes for the enterprise. Amend agreements when impacts on schedule, budget, or

performance are identified.
- Accept delivery in accordance with all agreements and relevant laws and regulations.
- Render payment or other agreed consideration in accordance with agreed payment schedules.
- Accept responsibility in accordance with all agreements and relevant laws and regulations.

■ *Common approaches and tips:*

- Establish acquisition guidance and procedures that inform acquisition planning, including recommended milestones, standards, assessment criteria and decision gates. Include approaches for identifying, evaluating, choosing, negotiating, managing and terminating suppliers.
- Establish a point of responsibility within the enterprise for monitoring and controlling individual agreements. This person maintains communication with the supplier, and is part of the decision-making team to assess progress in the execution of the agreement. The possibility of late delivery or cost overruns should be identified and communicated into the enterprise as early as noted.
- Define and track metrics that measure the progress on agreements. Appropriate metrics requires the development of tailored metrics/measures that do not drive unnecessary and costly efforts but do provide the information needed to assure the progress is satisfactory and key issues and problems are identified early to allow time for resolution with minimal impact to the delivery and quality of the product and service.
- Include technical representation in the selection of the suppliers to critically assess the capability of the supplier to perform the required task; this helps reduce the risk of contract failure and its associated costs, delivery delays, and increased resource commitment needs. Past performance is highly important but changes to key personnel should be identified and evaluated.
- Communicate clearly with the supplier about the real needs; avoid conflicting statements, or making frequent changes in the statement of need that introduce risk into the process.
- Maintain traceability between the supplier's responses to the acquirer's solicitation; this can reduce the risk of contract modifications, cancellations or follow-on contracts to fix the product or service.

6.8 Supply Process

6.8.1 Purpose

The Supply Process is invoked to establish an agreement between two enterprises under which one party supplies products or services to the other. Within the supplier enterprise, a project is conducted according to the recommendations of this handbook with the objective to provide a product or service that meets the contracted requirements. In the case of a mass produced product or service, a marketing function may represent the acquirer and establish customer expectations.

6.8.2 Description

The supply process is highly dependent upon the Enterprise, Project, and Technical Processes as it is through them that the work of executing the agreement is accomplished. This means that the supply process is the larger context in which the other processes are applied under contract. This section is written from the perspective of the supplier enterprise. Figure 6-8 is the context diagram for the Supply Process.

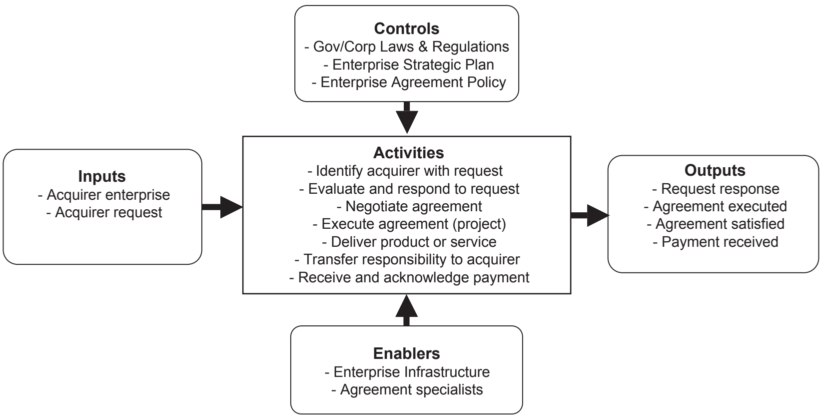


Figure 6-8 Supply Process Context Diagram

6.8.3 Inputs

The products and services available for acquisition are determined within the enterprise strategic plan. Legally executed agreements must comply with government and other directives. A supplier connects with another party with the desire to acquire their products or services. The supplier enterprise will adopt standard practices related to the negotiation of agreements. During the evaluation of the request from the acquirer, inputs are received from project management and engineering organizations, including potential sub-suppliers.

6.8.4 Outputs

The following are outcomes of the Supply Process.

- Identification of potential acquirers
- Response to acquirer request for proposal – formal documentation, or less formal inter-organizational interactions, e.g. between design engineering and

marketing

- Negotiated agreement – formal contracts, or less formal inter-organizational work orders
- System-of-interest (product or service) delivered according to delivery conditions of agreement
- Acquirer payments or other compensations received and acknowledged
- Responsibility for system-of-interest transferred between acquirer and supplier
- Communication between acquirer and supplier
- Supplier strategies regarding agreements

6.8.5 Process Activities

The following activities take place under the Supply Process:

- Manage Supply Process activities – decision-making for agreements, relationship building and maintenance, interaction with Enterprise management, responsibility for the development of plans and schedules, final approval authority for deliveries made to acquirer. When a supply process cycle concludes, conduct a final review of performance to extract lessons-learned for continued process performance.
- Develop and maintain Supply Plans, Strategies, Policies, Procedures to meet the enterprise goals and objectives and the needs of the project management and technical systems engineering organizations.
- Select appropriate acquirers – willing to conduct ethical negotiations, able to meet financial obligations, willing to maintain open communications throughout the supply process.
- Evaluate acquirer requests and prepare a response – a satisfactory response proposes a system-of-interest that meets acquirer needs and complies with industry and other standards. Assessments from the Investment, Resource Management, and Quality Management Processes are necessary to determine the suitability of this response and the ability of the enterprise to meet these commitments.
- Negotiate agreement – supplier commits to meet requirements for system-of-interest, meet delivery milestones, verification, validation and acceptance conditions, accept payment schedule, execute exception and change control procedures, and maintain transparent risk management procedures. The agreement will establish criteria for assessing progress toward final delivery.
- Execute the agreement – start a project and invoke the other processes defined in this handbook.
- Maintain communications with acquirer, sub-suppliers, stakeholders, and other organizations regarding the project.
- Assess execution of agreements to identify risks and issues, progress towards mitigation of risks, adequacy of progress toward delivery, evaluation cost and schedule performance, and determine potential undesirable outcomes for the

enterprise.

- Deliver in accordance with all agreements and relevant laws and regulations.
- Receive and acknowledge payment or other agreed consideration in accordance with agreed payment schedules.
- Transfer responsibility in accordance with all agreements and relevant laws and regulations.

■ *Common approaches and tips:*

- Agreements fall into a large range from formal to very informal based on verbal understanding. Contracts may call for a fixed price, cost plus fixed fee, incentives for early delivery, penalties for late deliveries, and other financial motivators.
- Relationship building and trust between the parties is a non-quantifiable quality that, while not a substitute for good processes, makes the human interactions agreeable.
- Develop technology white papers or similar documents to demonstrate and describe to the (potential) acquirer the range of capabilities in areas of interest. Use traditional marketing approaches to encourage acquisition of mass produced products.
- Maintain an up-to-date internet presence, even if the enterprise does not engage in electronic commerce.
- When expertise is not available within the enterprise (e.g. legal and other governmental regulations, laws, etc.), retain subject matter experts to provide information and specify requirements related to agreements.
- Invest sufficient time and effort into understanding acquirer needs before the agreement. This can improve the estimations for cost and schedule and positively affect agreement execution. Evaluate any technical specifications for the product or service for clarity, completeness and consistency.
- Involve personnel who will be responsible for agreement execution to participate in the evaluation of and response to the acquirer's request. This reduces the start-up time once the project is initiated, which in turn is one way to recapture the cost of writing the response.
- Make a critical assessment of the ability of the organization to execute the agreement; otherwise, the high risk of failure and its associated costs, delivery delays, and increased resource commitment needs will reflect negatively on the reputation of the entire enterprise.
- Institute for supply management has useful guidance for purchasing and marketing.[3]

---

1 ©2001 Jack Ring and A. Wayne Wymore

2 http://www.iso.ch/iso/en/iso9000-14000

3 http://www.napm.org/

THIS PAGE
INTENTIONALLY
LEFT BLANK

7 Enabling Systems Engineering Process Activities

Systems Engineering Process Activities are covered in three chapters:

- Chapter 7 – Enabling Systems Engineering Activities
- Chapter 8 – Systems Engineering Support Activities
- Chapter 9 – Specialty Engineering Activities

Chapter 1 (Table 1-1) provides summary guidance to the process activities covered in these chapters for those who may not be familiar with all of them.

From a sampling of the literature[1] in Systems Engineering from 1962 to the present, a common set of process activities emerges. This is the same set as that identified by a joint project between The American Institute of Aeronautics and Astronautics (AIAA) and INCOSE in which they are referred to as "enabling[2]" processes. The enabling processes are Decision Management, Requirements Management, and Risk and Opportunity Management. When tailoring a project, most of these activities will be essential to every system life cycle.

7.1. Decision Management

This section on Decision Management gives a glimpse into decision gates, making difficult decisions, and trade studies.

7.1.1. Decision Gates

A decision gate is an approval event in the project cycle, sufficiently important to be defined and included in the schedule by the project manager, executive management, or the customer. Entry and exit criteria are established for each gate at the time they are included into the project management baseline. Decision gates ensure that new activities are not pursued until the previously scheduled activities, on which new ones depend, are satisfactorily completed. Proceeding beyond the Decision Gate before the project is ready entails risk, as comically illustrated in Figure 7-1. The project manager may decide to accept that risk, as is done, for instance, with long-lead item procurement.

Figure 7-1 Decision Gates synchronize project activities[3]

The project business case issues of market demand, affordability, and realistic schedules are important decision criteria influencing concept selection, and they should be updated and evaluated at every decision gate. Inadequate checks along the way can set up subsequent failures – usually a major factor in cost over-runs and delays. At each gate the decision options are:

Acceptable: Proceed with the next stage of the project;

Or *Acceptable with reservations*: Proceed and respond to action items;

Unacceptable: Do not proceed; continue this stage and repeat the review when ready;

Unacceptable: Return to a preceding stage;

Unacceptable: Put a hold on project activity;

Unsalvageable: Terminate the project.

Upon successful completion of a decision gate, some artifacts (documents, models, or other products of a project cycle stage) have been approved as the basis upon which future work must build. If the project is large or long enough, or entails high risk, these artifacts are placed under configuration management.

Decision gate descriptions should identify the:

- Purpose of the decision gate
- Host and chairperson
- Attendees
- Location
- Agenda and how the decision gate is to be conducted
- Evidence to be evaluated
- Actions
- Method of closing the review

Decision gate approval must involve the necessary disciplines and stakeholders and must be based on hard evidence of compliance. One of the underlying principles for the agile development and extreme programming movements is to substantially reduce (but not eliminate) the frequency and elaborate (and they would claim pro-forma) content of decision gates for software development. Balancing the formality and frequency of decision gates is seen as a critical success factor for all systems engineering process areas. On large or lengthy projects, decisions and their rationale are maintained using an information management process.

7.1.2 Making Difficult Decisions

Making good decisions requires adequate information, experience, and good judgment. The techniques discussed in the following paragraphs are found in the literature, and have proven to be effective aids in making good decisions. In some cases, a technique may use mathematics to produce a result useful in the decision-making process, such as the hydrographical models used to assess the environmental restrictions in the Øresund Strait. People make decisions based on intuition and judgment; these techniques are aides to decision-making.[4]

Decision analysis is a method of identifying the best option from a set of alternatives, under uncertainty, using the possible outcomes of each alternative and their probabilities of occurrence to calculate the expected value of the outcome. Decision analysis has been a subject of interest for centuries,[5,6,7] and can be applied to a wide-range of problems and problem domains.

Skinner[8] states, “Real world decisions often involve a high degree of ambiguity, conflicting goals due to multiple objectives, complex trade-offs, more than one decision maker, or several sequential decisions. It is these types of situations where decision analysis is most valuable. By carefully decomposing the problem into smaller more manageable problems and by focusing on what is truly important, we can develop clear objectives and defensible courses of action.”

Skinner also lists ten principles of good decision making:

1. Use a value creation lens for developing and evaluating opportunities.
2. Clearly establish objectives and trade-offs.
3. Discover and frame the real problem.
4. Understand the business situation.
5. Develop creative and unique alternatives.
6. Identify experts and gather meaningful and reliable information.
7. Embrace uncertainty as the catalyst of future performance.
8. Avoid “analysis paralysis” situations.
9. Use systemic thinking to connect current to future situations.
10. Use dialog to foster learning and clarity of action.

Advocates of Lean Manufacturing would add one more suggestion to this list; delay commitment until the last responsible moment. Lean software development delays freezing all design decisions as long as possible, because it is easier to change a decision that is not made.[9]

Decision trees are a graphical and quantitative method for thinking through a decision. The first step is to create a decision "tree" diagram that represents the situation in question. Starting on the left with the initial decision point and proceeding to the right, the decision diagram must accurately represent each point where a decision is to be made and all the possible consequences of that decision. Figure 7-2 illustrates a situation where management needs to decide on whether or not to bid on a contract. The team estimates that their company has a 60% chance of winning. If they propose a unit price of $30, the company will earn $4.25 M, where M is 106. They further estimate that there is a 10% chance that the unit cost will be $28, which would result in an increase in income to $6 M. There is a 40% chance that the unit cost could be as high as $38, and the project will lose about $3 M. The unit costs and probabilities of each chance outcome are based on the best judgment of the team.

The expected value of winning the contract is the sum of the expected value for each branch at the chance node times the probability for each branch. So

Contract Win Expected Value = 10% * $6.05 M + 50% * $4.25 M – 40% * $2.95M or $1.55M

The expected value of making the bid is 60% * $1.55 – 40% * 0.25 or $0.83 M

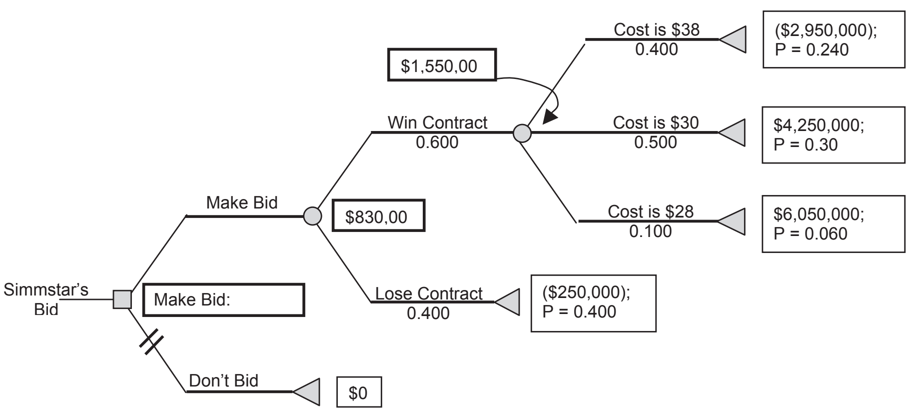


Figure 7-2 Decision Tree for a "Bid – No Bid" Decision[10]

This technique can be extended to include multiple decision points and multiple outcomes as long as every possible outcome has a value and a probability of occurrence associated with it.

Additional decision analysis techniques include:

a. Sensitivity analysis, which looks at the relationships between the outcomes and their probabilities to find how “sensitive” a decision point is to the relative numerical values.
b. Value of information methods, whereby expending some effort on data analysis and modeling can improve the optimum expected value.
c. Multi-attribute Utility Analysis, which is a method that develops equivalencies between dissimilar units of measure.

There are many tools available to support the decision management area. The INCOSE website maintains suggestions from the Tools Working Group.

7.1.3 Trade Study and Sensitivity Analysis

Trade study describes a process for comparing the appropriateness of different technical solutions. (Appendix L elaborates on a number of decision support methods). The characteristics of each option are traded against each other. Once a best alternative has been identified, the stakeholders in the decision will want to know how sensitive the recommended selection is to differently evaluated criteria or to different estimates of the alternatives’ characteristics — perhaps a different best alternative would result. Therefore, a good trade study provides a disciplined process that justifies the selected approach, and includes sensitivity analysis.

A recent study reported that the following activities can be found in most trade study processes: [11]

1. Frame the problem context, scope, constraints.
2. Establish communications with stakeholders.
3. Define evaluation criteria and weights where appropriate.
4. Define alternatives and select candidates for study.
5. Define measures of merit and evaluate selected candidates.
6. Analyse the results and select best alternative.
7. Review results with stakeholders and re-evaluate.
8. Investigate the consequences of implementation.
9. Use scenario planning to test assumptions about the future.

The utility or value of each feature of an alternative is determined or estimated, and often a weight is defined that assigns a relative importance of each feature across all alternatives. The weighted value of utility is simply the product of the utility and the weight for each feature, and the weighted total is the sum of the weighted utility values summed over all the features of an alternative. The selected alternative nominally is the one with the best weighted total. Involvement of the stakeholders in this process gives them confidence in the eventual choice and imparts useful insights to the whole team.

A sensitivity analysis involves varying each utility and each weight and re-computing the weighted total for each alternative to ascertain what would change if the values of the utilities or weights were different. The significance of the change is best determined through conversations with stakeholders and subject matter experts. All evaluations and decisions are reviewed to address the concerns and opinions of stakeholders.

A final evaluation of the consequences of implementing the selected alternative may help identify unintended consequences of an otherwise "best" solution. The highest score does not always win.

Trade studies support decisions in all phases of system development, from conceptualization to deployment. Requirements can be traded against constraints; architecture features can be traded against dictated equipment or interface requirements; alternative functional or performance choices can be traded to determine an optimal configuration. In the case of the Øresund Bridge, trade studies helped determine many of the final elements of the bridge configuration, e.g. length of main span.

7.2 Requirements Management

There is near unanimous agreement that successful projects depend on meeting the needs and requirements of the stakeholder/customer. When the requirements are for a complex system, or for a system that may take many years to realize, a formal requirements management process is justified. Requirements management concerns the collection, analysis, and validation of requirements with all the communications and negotiations inherent in working with people.

A great deal of literature exists on how to write and manage requirements. This overview amplifies four major elements of this process area discussed in Section 4.3; elicit and capture requirements, generate a concept of operations, define system capabilities and performance objectives, and define non-functional requirements. (Appendix I elaborates on the requirements definition process, and Appendix F provides information on capturing user needs and user requirements.)

7.2.1 Elicit and Capture Requirements

Within the context of ISO/IEC 15288, requirements are specifically mentioned in two of the technical processes, and are drivers for many of the system life cycle processes. Depending on the system development model, requirements capture may be done nominally once near the beginning of the development cycle, or as for agile methods, be a continuous activity. The reason for eliciting requirements is the same, understand the needs of the stakeholders well enough to support the architecture design process.

One of the biggest challenges in this activity is the identification of the set of stakeholders from whom requirements should be elicited. Customers and eventual end-users are relatively easy to identify, but regulatory agencies, and other interested parties that may reap the consequences of the system-of-interest should also be sought out and heard. In sustainable development this includes finding representation for future generations. Figure 7-3 illustrates the range of potential stakeholders.

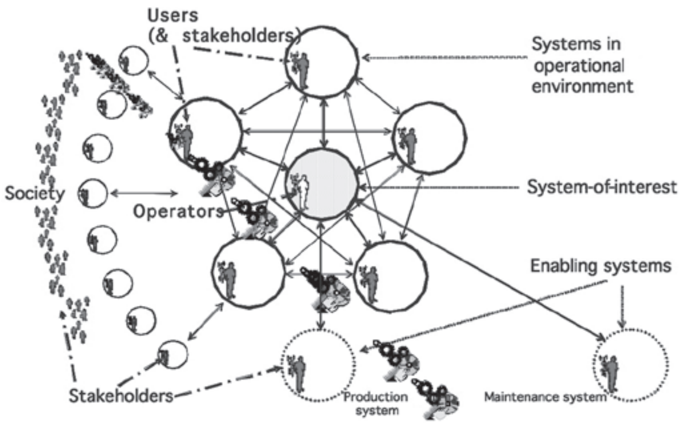


Figure 7-3 Requirements elicitation captures the needs of stakeholders, operators and users across systems boundaries[12]

Requirements elicitation is an iterative activity and benefits from continuous communication and validation with stakeholders. Techniques for requirements elicitation include interviews, focus groups, the Delphi technique, and soft systems methodology. Tools for capturing and managing requirements are many and varied. The INCOSE Tools Database Working Group evaluates the relative merits of different products and maintains a database that is available from the INCOSE website.

7.2.2 Systems Modeling Language (SysML)

SysML is used to model complex systems and is an extension of the family of UML-based standards that are intended to provide standard representations with well defined semantics that can support model and data interchange. SysML has been developed as part of a joint initiative between the Object Management Group and INCOSE.[13, 14, 15]

SysML includes diagrams that can be used to specify system requirements, behaviour, structure, and parametric relationships. The modelling elements represented in the diagram facilitate integration among the various diagrams and views. The SysML diagram types in Figure 7-4 are summarized below.

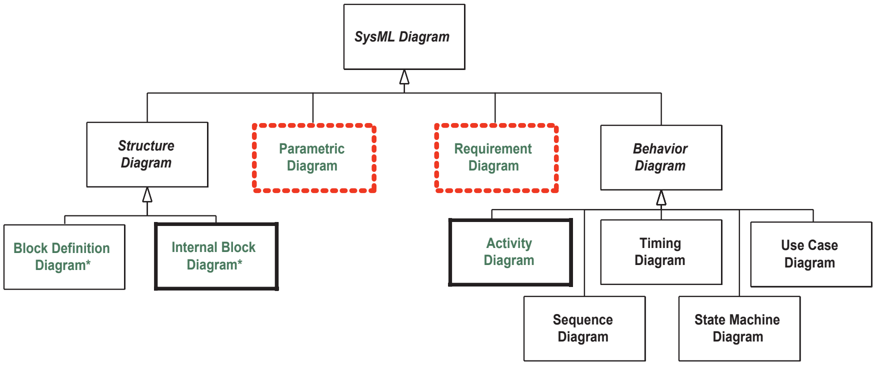


Figure 7-4 SysML Diagram Types

The system structure is represented as block definition diagrams and internal block diagrams. A block definition diagram describes the system hierarchy and system/component classifications. The internal block diagram describes the internal structure of a system in terms of its parts, ports, and connectors.

The behavior diagrams include the use case diagram, activity diagram, sequence diagram, state machine diagram, and timing diagram. A use-case diagram provides a high-level description of the system functionality. The activity diagram represents the flow of data and control between activities. A sequence diagram represents the interaction between collaborating parts of a system. The state machine diagram describes the state transitions and actions that a system or its parts perform in response to events. The timing diagram represents parameters, functions or states as a function of time.

The requirements diagram captures requirements hierarchies, and the derivation, satisfaction, and verification relationships. It provides a bridge between requirements and system design models. The parametric diagram represents constraints on system parameter values such as performance, reliability, and mass properties to support engineering analysis. SysML includes an allocation relationship to represent allocation of functions to components, allocation of logical to physical components, and other types of allocations.

7.2.3 Concept of Operations

Scenarios and what-if thinking are essential tools for planners who must cope with the uncertainty of the future. Scenario thinking can be traced back to the writings of early philosophers such as Plato and Seneca.[16] As a strategic planning tool, scenario techniques have been employed by military strategists throughout history. Building scenarios serves as a methodology for planning and decision-making in complex and uncertain environments. The exercise makes people think in a creative way, observations emerge which reduce the chances of overlooking important factors, and the act of creating the scenarios enhances communications within and between organizations.

Creation or upgrade of a system shares the same uncertainty regarding future use and emergent properties of the system. The Stakeholder Requirements Definition Process suggests capturing the understanding of stakeholder needs in a series of concept documents, each focused on a life cycle stage, see chapter 4.2. The goal of a concept document is to capture an implementation-free understanding of stakeholders' needs by defining what is needed, without addressing how to satisfy the need. It captures behavioral characteristics required of the system in the context of other systems with which it interfaces, and captures the manner in which people will interact with the system for which the system must provide capabilities.

If the system is for a military customer, there may be several required operational views of the system driven by architectural frameworks. These are defined, for example, in the United States Department of Defense Architecture Framework (DODAF) and in the Ministry of Defense (UK) Architecture Framework (MODAF).

One objective is to ensure that operational needs are clearly understood and the rationale for performance requirements is incorporated into the design decision database. Other objectives are:

a. To provide traceability between operational needs and the captured source requirements.

b. To establish a basis for requirements to support the system over its life, such as personnel requirements, support requirements, etc.

c. To establish a basis for test planning, system-level test requirements, and any requirements for environmental simulators.

d. To generate operational analysis models to test the validity of external interfaces between the system and its environment, including interactions with external systems.

e. To provide the basis for computation of system capacity, behavior under/overload, and mission-effectiveness calculations.

f. To validate requirements at all levels and to discover implicit requirements overlooked from other sources.

Since a concept of operations describes system behavior, a starting point for building up the concept is to begin by identifying outputs generated by external systems (modified as appropriate by passing through the natural system environment) which act as stimuli to the system-of-interest, causing it to take specified actions and produce outputs, which in turn are absorbed by external systems. These single threads of behavior eventually cover every aspect of operational performance, including logistical modes of operation, operation under designated conditions, and behavior required when experiencing mutual interference with multi-object systems. Aggregation of these single threads of behavior represents a dynamic statement of what the system is required to do.

Scenario building is an essentially human activity that may involve interviews with operators of current/similar systems, potential end users, and meetings of an Interface Working Group. The results of this exercise can be captured in many graphical forms using modeling tools and simulations.

7.2.4 Define systems capabilities and performance objectives

The concepts of production, deployment, operations, and support serve as an excellent foundation from which the systems engineer can discern the required capabilities of the system-of-interest and the relevant performance objectives of the system. Together with identified system constraints, these will drive the architecture design activities.

Typical constraints on the system may include:

- Cost and schedule
- Mandated use of Commercial Off-The-Shelf (COTS) equipment
- Operational environment and use of pre-existing facilities and system elements
- Operational interfaces with other systems or organizations

As a result of this activity, a number of performance requirements will be identified. These may include areas such as power, propulsion, communications, data processing, environmental, and human interaction and intervention. In the Maglev Train, the desire to cover large distances in a brief time established train speed parameters, and the need to carry people suggested safety and maximum noise tolerances. It is advisable to place the assumptions, constraints, and analyses associated with derived requirements in the decision and/or requirements database(s).

The concept of operations will also help identify adverse consequences of derived requirements:

- Is unnecessary risk being introduced?
- Is the technology producible?
- Are sufficient resources available to move forward?
- Are trade studies needed to determine appropriate ranges of performance?

Large systems may justify the development of a high-level system simulation evolved from the system architecture. The simulation should contain sufficient functional elements that the interactions can be properly assessed. The purpose of the simulation is to establish measurable parameters for the functional requirements. This provides the necessary guidance to the designers on the size and capability required of their equipment. In addition, these parameters will be used as an integral part of the verification process in establishing the capability of the equipment (and the system) to satisfy user needs.

When time permits, use of an interdisciplinary team to audit the requirements may help ensure the clarity, completeness and consistency of the set. Such a team can also assess that the requirements are verifiable. Unfortunately, it is possible to write reasonable-sounding requirements which in fact cannot be met. In the 1980s, a government/industry team created requirements for a weather satellite program, with eight satellites to be built and launched, one every three years. The specifications were – and still are – so tight that the last satellite scheduled to be launched in 2008 must have a waiver, since it cannot meet the original specifications.

If there is uncertainty associated with a requirement, it should be identified as needing further attention, and even proposed for monitoring as part of the project risk management. Resolution of uncertainty should be assigned as a responsibility to an individual, and progress and eventual resolution recorded in the decision database.

7.2.5 Technical Performance Measures

Technical Performance Measures (TPM) express the objective performance requirements, are evaluated at decision gate reviews, and may be used to assess the risk position of the project. TPM provide visibility into important project technical parameter status to enable effective management enhancing the likelihood of achieving the technical objectives of the project. Limit the number of TPM being monitored to critical issues. Collecting too many measures without knowing how they can be used wastes time and resources, and, even worse, the really useful values may become lost in the ocean of data. The Systems Engineering Plan (SEP) should define the approach to TPM.

Without TPM, a project manager could fall into the trap of relying on cost and schedule status alone, with perhaps the verbal assurances of technical staff to assess project progress. This can lead to a product developed on schedule and within cost that does not meet all key requirements. Values are established to provide limits that give early indications if a TPM is out of tolerance, as illustrated in Figure 7-5.

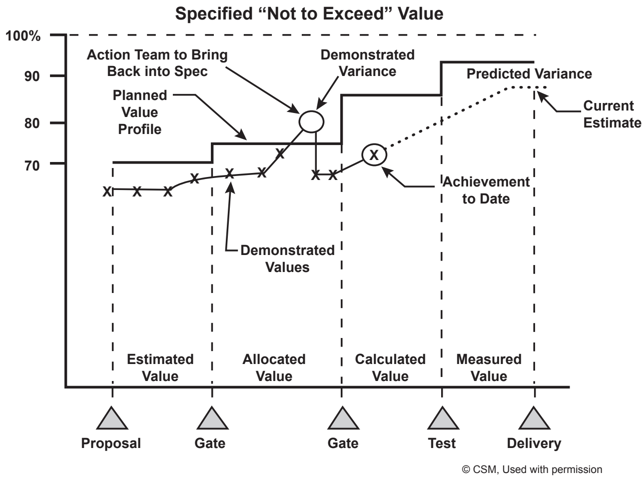


Figure 7-5 TPM Monitoring

Periodic recording of the status of each TPM provides the continuing verification of the degree of anticipated and actual achievement of technical parameters. Measured values that fall outside an established tolerance band will alert management to take corrective action.

7.2.6 Define other non-functional requirements

The concept documents will also suggest requirements that are not directly related to the primary capability provided by the system-of-interest. Many of these are discussed further in chapter 9.2, such as availability, supportability, security, and training. The Øresund Bridge case illustrated the avoidance of negative environmental impact by establishing constraints on the construction practices. Addressing non-functional requirements from the earliest stages is a good way to ensure that they are not forgotten and that they are satisfied.

7.3 Risk and Opportunity Management

Most projects are executed in an environment of uncertainty. Uncertainty influences the ability of the project team to achieve the project objectives. Uncertainty includes events that could harm the project (threats) and those that could help the project (opportunities). Well-established techniques exist for managing threats. There is some debate whether the same techniques are applicable to recognizing opportunities. In an optimal situation, opportunities are maximised at the same time as threats are minimised, resulting in the best chance to meet project objectives.[17] The Øresund Bridge case illustrates this; the man-made Peberholm Island was created from the materials dredged from the Strait to meet environmental requirements and is now a sanctuary for a rare species of tern.

7.3.1 Concepts

Definitions: Risks are events that if they occur can jeopardize the successful completion of the project. Risks should be identified and assessed for probability of occurrence and impact on the project.[18]

"Traditionally, risk has been defined as the likelihood of an event occurring coupled with a negative consequence of the event occurring. In other words, a risk is a potential problem — something to be avoided if possible, or its likelihood and/or consequences reduced if not."[19] As a corollary, opportunity is "the potential for the realization of wanted, positive consequences of an event."[20]

Fundamentals: the measurement of risk has two components as illustrated in Figure 7-6:

- The likelihood that an event will occur.
- The undesirable consequence of the event if it does occur.

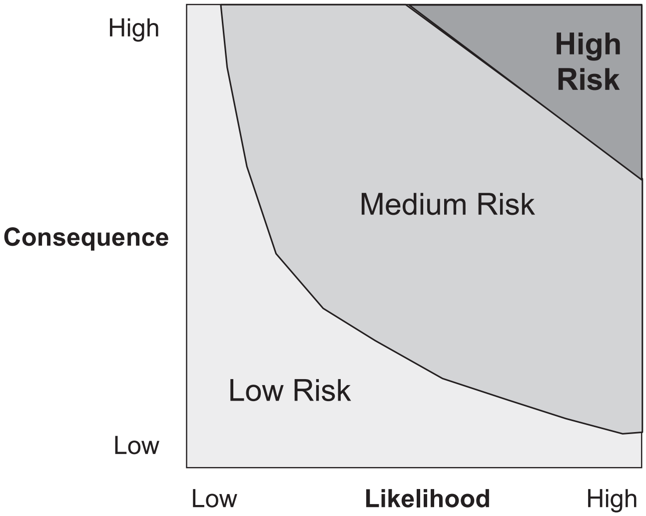


Figure 7-6 Level of risk depends upon both likelihood and consequences

The likelihood that an undesirable event will occur often is expressed as a probability. The consequence of the event is expressed in terms that depend on the nature of the event (e.g. lost investment, inadequate performance). The combination of low likelihood and low undesirable consequences gives low risk, while high risk is produced by high likelihood and highly undesirable consequences.

By changing the adjective from undesirable to desirable the noun changes from risk to opportunity, but the diagram remains the same. As suggested by the shading, most projects experience a comparatively small number of high risk or high opportunity events.

Risk pervades the life cycle of systems. The system may be intended for technical accomplishments near the limits of the state of the art, creating technical risk. System development may be rushed to deploy the system as soon as possible to meet an imminent threat, leading to schedule risk. All systems are funding-limited so that cost risk is present. Risk can be introduced by external constraints or can develop from within the project, since technical risk can create schedule risk, which in turn can create cost risk.

There is no alternative to the presence of risk in system development. The only way to remove risk is to set technical goals very low, to stretch the schedule, and to supply unlimited funds. None of these events happen in the real world. No realistic project can be planned without risk. The challenge is to define the system and the project which best meet overall requirements, which allow for risk, and which achieve the highest chances of project success.

Figure 7-7 illustrates the major interactions between the four risk categories; technical, cost, schedule and programmatic. The arrow names indicate typical risk relationships; others certainly are possibe.

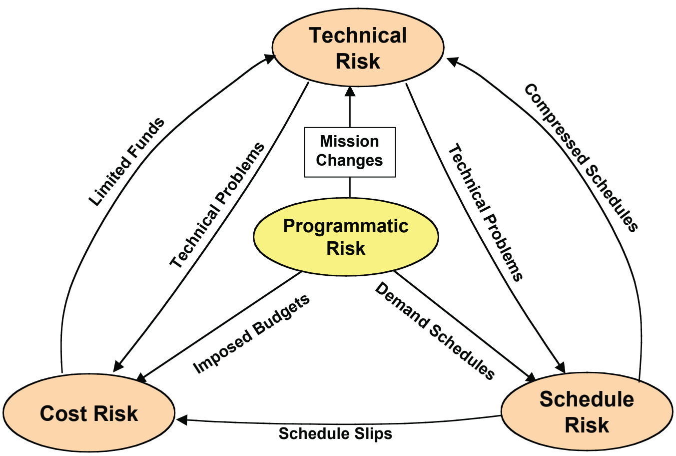


Figure 7-7 Typical Relationship among the Risk Categories

Technical risk is the possibility that a technical requirement of the system may not be achieved in the system life cycle. Technical risk exists if the system may fail to achieve performance requirements; to meet operability, producibility, testability, integration requirements; or to meet environmental protection requirements. A potential failure to meet any requirement which can be expressed in technical terms is a source of technical risk.

Cost risk is the possibility that available budget will be exceeded. Cost risk exists if the project must devote more resources than planned to achieve technical requirements;

if the project must add resources to support slipped schedules due to any reason; if changes must be made to the number of items to be produced; or, if changes occur in the enterprise or national economy. Cost risk can be predicted at the total project level or for a system element. The collective effects of element-level cost risks can produce cost risk for the total project.

Schedule risk is the possibility that the project will fail to meet scheduled milestones. Schedule risk exists if there is inadequate allowance for acquisition delays. Schedule risk exists if difficulty is experienced in achieving scheduled technical accomplishments, such as the development of software. Schedule risk can be incurred at the total project level for milestones such as deployment of the first system element. The cascading effects of element-level schedule risks can produce schedule risk for the total project.

Programmatic risk is produced by events which are beyond the control of the project manager. These events often are produced by decisions made by personnel at higher levels of authority. Programmatic risks can be produced by reductions in project priority, by delays in receiving authorization to proceed with a project, by reduced or delayed funding, by changes in enterprise or national objectives, etc. Programmatic risk can be a source of risk in any of the other three risk categories.

The Risk Management process activities are Risk Identification, Risk Assessment, Risk Handling, Risk Tracking and Control, and Risk Mitigation.

7.3.2 Risk Identification

The purpose of risk identification is to identify risks and evaluate their relative severity. The basis for this evaluation may be qualitative or quantitative. All stakeholders and project personnel should feel welcome to contribute to risk identification. The objective is to set priorities and focus attention on areas of risk with the greatest consequences to the success of the project.

If a project is unprecedented, brainstorming using SWOT (Strength-Weakness-Opportunity-Threat) or Delphi techniques may be appropriate. But most projects represent a new combination of existing systems or system elements or represent the insertion of incremental advances in technology. This means that key insights can be gained concerning a current project's risk by examining the successes, failures, problems, and solutions of similar prior projects. The experience and knowledge gained, or lessons learned, can be applied to identify potential risk in a new project and to develop a strategy for risk management.

The first step is to determine the information needs in this phase of risk management. This could vary from assessing the risk in development of a custom computer chip to identifying the risks associated with a major system development. The second step is to define the basic characteristics of the new system. This is necessary to identify past projects that are similar in technology, function, design, etc. Then, based on the availability of data, analogous systems or subsystems are selected and data gathered. Often the data collection process and initial assessment lead to a further definition of the system for the purposes of comparison. After this has been accomplished, the last step in the process is the analysis and normalization of the historic data. Comparisons to prior systems may not be exact or the data may need to be adjusted to be used as a basis for estimating the future. The desired output is insight into cost, schedule, and technical risks of a project based on observations of similar past projects.

7.3.3 Risk Assessment

Uncertainty is characterized by a distribution of outcomes based on likelihood of occurrence and severity of consequences. Risk involves both the probability and consequences of the possible outcomes. In its most general form, risk assessment should capture the spectrum of outcomes relative to the desired project technical performance, cost, and schedule requirements. Risk generally needs to be assessed subjectively because adequate statistical data are rarely available.

Expert Interviews: Efficient acquisition of expert judgments is extremely important to the overall accuracy of the risk management effort. The expert interview technique consists of identifying the appropriate experts, questioning them about the risks in their area of expertise, and quantifying these subjective judgments. One result is the formulation of a range of uncertainty or a probability density function (with respect to cost, schedule, or performance) for use in any of several risk analysis tools.

Since expert interviews result in a collection of subjective judgments, the only real "error" can be in the methodology for collecting the data. If it can be shown that the techniques for collecting the data are not adequate, then the entire risk assessment can become questionable. For this reason, the methodology used to collect the data must be thoroughly documented and defensible. Experience and skill are required to encourage the expert to divulge information in the right format. Typical problems encountered include identification of the wrong expert, obtaining poor quality information, unwillingness of the expert to share information, changing opinions, getting biased viewpoints, obtaining only one perspective, and conflicting judgements. When conducted properly, the expert interviews provide very reliable qualitative information. However, the transformation of that qualitative information into quantitative distributions or other measures depends on the skill of the analyst.

Models: Risk is often expressed only in qualitative terms or by a single value. However, it is very important to quantify risk in some methodical way to assure a good allocation of resources for risk reduction. Ideally, risk would be characterized by using cumulative probability curves with the probability of failure and the consequences expressed quantitatively in measurable terms, but given the inherent lack of data and limited analysis, this is usually impractical. It is very important to properly quantify risk because an invalid assessment could lead to an improper conclusion with misapplication of resources.

Expected Value Model: A somewhat subjective, relative rating of risk is developed, where risk is expressed as:

Expected consequence = Probability of failure (Pf)* Consequences of failure (Cf).

For illustration purposes, consider a proposal to develop a new light-weight and compact power supply with an operating life of 8,000 hours. The consequences of failing to meet at least 6,000 hours are assessed to be critical, so the consequence of failure is assigned a value of 0.8. Given the present state of technology, cost and schedule, the probability of failing to achieve an operating life of 6,000 hours is judged to be relatively low and is estimated as 30% (0.3).

Applying the equation to the above example yields

Risk = 0.3*0.8 = 0.24

This would suggest a relatively low risk situation. Intuitively, the described scenario represents a low/moderate risk (subjective judgment); therefore this approach appears to yield a valid relative ranking of risk.

7.3.4 Risk Handling

Risk handling approaches need to be established for the moderate and high-risk items identified in the risk assessment effort. These activities are formalized in the Risk Management Project Plan, produced within the Risk Management Process, chapter 5.6. There are basically four approaches to handle risk:

- avoid the risk through change of requirements or redesign
- accept the risk and do no more;
- control the risk by expending budget and other resources to reduce likelihood and/or severity;
- transfer the risk by agreement with other party that it is in their scope to mitigate; or,

7.3.5 Risk Tracking and Control

Project management uses metrics to simplify and illuminate the risk management process. Each risk category has certain indicators, which may be used to monitor project status for signs of risk. Tracking the progress of key system technical parameters can be used as an indicator of technical risk.

The typical format in tracking technical performance is a graph of a planned value of a key parameter plotted against calendar time. A second contour showing actual value achieved is included in the same graph. Cost and schedule risk are monitored using the products of the Cost/Schedule Control System or some equivalent technique. Normally cost and schedule variances are used, along with a comparison of tasks planned to tasks accomplished.

7.3.6 Risk Mitigation

Risk mitigation activities conform to the risk handling options. There are some steps that can be taken to avoid unnecessary risks.

Requirements scrubbing – Requirements that significantly complicate the system can be scrutinized to ensure that they deliver value equivalent to their investment. Find alternative solutions that deliver the same or comparable capability.

Selection of most promising options – In most situations several options are available, and a trade study can include project risk as a criterion when selecting the most promising alternative.

Staffing and team building – Projects accomplish work through people. Attention to training, teamwork, and employee morale can help avoid risks introduced by human errors.

For high-risk technical tasks, risk avoidance is insufficient and can be supplemented by the following approaches:

- Early procurement
- Initiation of parallel developments
- Implementation of extensive analysis and testing
- Contingency planning

The high-risk technical tasks generally imply high schedule and cost risks. Cost and schedule are impacted adversely if technical difficulties arise and the tasks are not achieved as planned. Schedule risk is controlled by early procurement of long-lead items and provisions for parallel-path developments. However, these activities also result in increased early costs. Testing and analysis can provide useful data in support of key decision points. Finally, contingency planning involves weighing alternative risk mitigation options.

In China, the authorities built the short Maglev train line in Shanghai as a proof-of-concept. In spite of the high investment, this represented lower risk to the project than attempting a longer line with an unproven technology. This results collected from this project are inspiring others to consider Maglev alternatives for greater distances.

A number of references exist on the topic of risk management. [21, 22, 23, 24, 25]

---

1 Literature sampling included a – k

a) Armstrong, J. E. and A. P. Sage. *Introduction to Systems Engineering,* John Wiley & Sons, 2000.

b) Aslaksen, E. W. *The Changing Nature of Engineering.* McGraw Hill, 1996.

c) Blanchard, B. *Logistics Engineering and Management,* 4 ed. Prentice Hall, 1992.
d) Chase, W. P., *Management of System Engineering.* John Wiley & Sons, 1974.
e) Chestnut, H., Systems Engineering Methods. New York: Wiley, 1967.
f) Forsberg, K., et. al., *Visualizing Project Management, 3 ed.* John Wiley & Sons, 2005.
g) Hall, A. D. *A Methodology for Systems Engineering.* Princeton, NJ: Van Nostrand, 1962.
h) Kossiakof, A and W. Sweet. *Systems Engineering Principles and Practice.* John Wiley & Sons, 2003.
i) Martin, J. N. *Systems Engineering Guidebook,* CRC Press, 1996.
j) Sage, A. P. *Methodology for Large-scale Systems.* McGraw Hill, 1977.
k) Stevens, R. et. al. *Systems Engineering: coping with complexity.* Prentice Hall Europe, 1998.

2 AIAA and INCOSE, Systems Engineering Primer, 1997, www.incose.org/ProductsPubs/products/seprimer.aspx

3 Used with permission from United Media

4 Teale, M., et. al., *Management Decision-making.* Prentice Hall, 2003.

5 Bayes, the Reverend Thomas (1702 – 1761). "Essay Toward Solving a Problem in the Doctrine of Chances." Philosophical Transactions of the Royal Society of London, 1763.

6 Raiffa, Howard, Decision Analysis: Introductory Lectures on Choices Under Uncertainty., Addison-Wesley, New York, 1968.

7 Schlaiffer, Robert, Analysis of Decisions Under Uncertainty, McGraw-Hill Book Company, New York, 1969.

8 Skinner, David C., Introduction to Decision Analysis, Second Edition, Probabilistic Publishing, Gainsville, FL 2001. p 34, 88.

9 Poppendieck. Mary, "Morphing the Mold", *Software Development,* Jul 07, 2003.

10 Skinner, p. 34.

11 Emes, Michael, "Strategic Multi-stakeholder Trade Studies," *Proceedings of EuSEC, Edinburgh, 18-20.sep 2006.*

12 Arnold, Stuart and Harold Lawson, "Viewing Systems from a Business Management Perspective: The ISO/IEC 15288 Standard," *Journal of Systems Engineering,* 7:3, 2003.

13 Friedenthal, Sanford, Burkhart, *Extending UML from Software to Systems,* Proceedings *Thirteenth Annual International Symposium INCOSE.* Arlington, VA, USA, June 30- July 1, 2003.

14 Friedenthal, Sanford, and Kobryn, "Extending UML to Support a Systems Modeling Language", INCOSE Symposium, Toulouse, France June 2004

15 OMG Systems Engineering Domain Special Interest Group (SE DSIG) Website, http://syseng.omg.org

16 Heijden, K. vd, et. al., *The Sixth Sense,* John Wiley & Sons, 2002; p. 121.

17 Project Management Institute, 2000. *A Guide to the PMBOK.* Upper Darby, USA: p. 176.

18 Widerman, R. Max. Comparative Glossary of Common Project Management Terms v3.1, March 2002.

19 Conrow, E. H., Effective Risk Management, 2nd Edition, American Institute of Aeronautics and Astronautics, Inc., VA, 2003, p 435.

20 Ibid. p. 436.

21 *AT&T Engineering Guides for Managing Risk: Design to Reduce Technical Risk; Design's Impact on Logistics; Moving a Design into Production; Testing to Verify Design and Manufacturing Readiness,* from the McGraw-Hill Professional Book Group (1-800-842-3075)

22 Barton, Thomas L., William G. Shenkir, and Paul L. Walker, *Making Enterprise Risk Management Pay Off: How Leading Companies Implement Risk Management,* Upper Saddle River, NJ: Financial Times/Prentice Hall PTR/Pearson Education Company, 2002

23 Michel, Robert Mark, and Dan Galai, *Risk Management,* NY: McGraw Hill, 2001

24 Shaw, T.E. and Lake, J. G., Ph.D., “Systems Engineering: The Critical Product Development Enabler,” APICS Conference, April 1993

25 Wideman, R. Max, Editor, Project and Program Risk Management: A Guide to Managing Project Risks and Opportunities, Newtown Square, PA: Project Management Institute, 2004.

8 Systems Engineering Support Activities

The topics in chapter 8 are covered in alphabetical order by topic title to avoid giving more weight to one topic over another. See Table 1-1 for an overview. The objective of this chapter is to present the most frequently recommended set of activities within the systems engineering effort. Depending on the nature of the project, its complexity, size, and duration, some or most of these activities will be included in the project execution. More information about each activity can be found in references to external sources and the IPAL.

8.1 Acquisition and Supply

The initiation of a project begins with user need. Once a need is perceived and resources are committed to establish a project it is possible to define the parameters of an acquisition and supply relationship. This relationship exists whenever an organization with a need does not have the ability to satisfy that need without assistance. Acquisition is also an alternative for optimising investment when a supplier can meet the need in a more economical or timely manner. The Acquisition and Supply Processes are the subject of chapters 6.7 and 6.8, respectively.

The acquisition and supply processes are two sides of the same coin. Each process establishes the contractual context and constraints under which the other system life cycle processes are performed. The unique activities for the agreement processes are related to contracts and managing business relationships. An important contribution of the ISO/IEC 15288 is the recognition that systems engineers are relevant contributors in this domain.[1] The Maglev train case is an example where the government representatives of China and Germany participated in the relationship.

Contract negotiations are handled in various ways depending on the specific organization. In a process that is widely used, the contracts organization in industry (or the contracting officer in the government) is responsible for negotiating contracts, including the contract terms and conditions. Key parameters such as profit target and acceptable contract type (firm fixed price, cost plus fixed fee, cost plus award fee) are established by the business area manager or by enterprise management.

Project managers rarely lead contract negotiations, however, the lead contract negotiator should only agree to any changes in scope, cost, or schedule with the project manager's approval. The systems engineer is in a supporting role to the project manager during negotiations.

The lead contract negotiator may need, within minutes or a few hours, an assessment of the impact of customer-proposed changes; for major changes, the team may need a few days. In preparation for contract negotiations, systems engineers often perform preliminary trade studies on a range of cost, schedule, and technical performance options that might be proposed by the customer during negotiations (or, for subcontract negotiations, changes that might be presented by the supplier). Of particular importance is the impact to project risk. What is needed is accuracy – not precision – so the team is prepared for anything reasonable that might arise. A team that is prepared will always have a more favorable outcome in negotiations, and the buyer will be pleased to work with a knowledgeable provider.

A critical element to each party is the definition of acceptance criteria. These criteria protect both sides of the business relationship – the acquirer from being coerced into accepting a product with poor quality; and the supplier, from the vagaries of a fickle or indecisive buyer.

8.2 Architectural Design

Developing the system architecture is one of the most important responsibilities of the systems engineer. It is a creative process, and there is no unique solution to satisfying user requirements. The system architecture is critical because it provides the framework for system development. The Architectural Design Process is the topic of chapter 4.4.

In his book Wasson states, "System, product, or service architectures depict the summation of a system's entities and capabilities levels of abstraction that support all phases of deployment, operations, and support."[2] In 1987 John Zachman first formally enunciated the opinion that in the modern world we should no longer talk about multiple architectures; that instead we need to talk about multiple views of a single broad-reaching architecture.[3,4] Consider developing architectural alternatives that are significantly different in their approach to meeting stakeholder requirements, as illustrated in Figure 8-1.

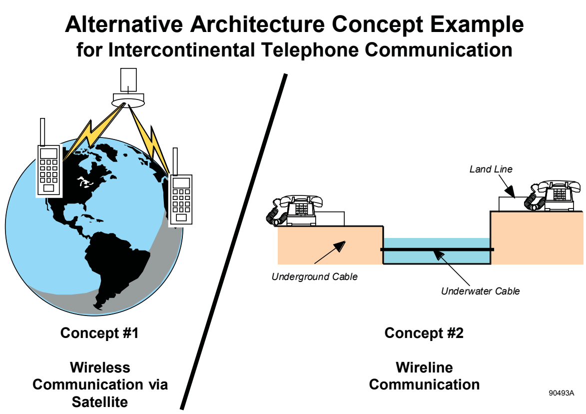


Figure 8-1 Example of Alternative Architectural Concepts

Creating an effective system architecture draws on the experience, intuition, and good judgment of the team to devise an appropriate solution. As Rechtin and Maier define it, systems architecting builds on four methodologies:[5]

- Normative (solution based) such as building codes and communication standards.
- Rational (method based) such as systems analysis and engineering.
- Participative (stakeholder based) such as concurrent engineering and brainstorming.
- Heuristic (lessons learned) such as "Simplify. Simplify. Simplify."

Because systems architecting is a creative process, and because intuition and experience play such an important role, the systems engineer must pay attention to situations where past experience and intuition have been a handicap. The book, *The Innovator's Dilemma*,[6] by Clayton Christensen, clearly sets forth the benefit of making creative use of past experience. He uses the example of the computer hard disk drive industry in which experience had been a key factor to the growth of companies that dominated the hard disk market. Christensen then highlights the transition difficulties for companies making large disks and their inability to capture any market share when the industry moved to smaller size disks. According to Christensen no manufacturer made a successful transition from a 14-inch (35.6 cm) disk to the 8-inch (20.3 cm) disk; a whole new set of companies dominated that market. This was repeated as the sizes dropped from 8 to 5-¼ to 3-½ to 2-½ to 1.8 inches. This sequence started in 1980s and is continuing today with the flash drives. In each of these transitions, the established companies lost out, in part because their established user base was locked-in to the older architecture, and in part, because their entire enterprise from systems engineering to marketing to manufacturing to executive management was unable to see the new vision.

8.2.1 Interoperability Analysis

Interoperability depends on the compatibility of components of a large and complex system (which may sometimes be called a system of systems or a family of systems) to work as a single entity. This feature is increasingly important as the size and complexity of systems continues to grow. Pushed by an inexorable trend toward electronic digital systems and pulled by the accelerating pace of digital technology invention, commercial firms and national enterprises span the world in increasing numbers. As their spans increases, these commercial and national enterprises want to make sure that their sunk investment in legacy components of the envisioned new system is protected and that new components added over time will work seamlessly with the legacy components to comprise a unified system.

Standards have also grown in number and complexity over time, yet compliance with standards remains one of the keys to interoperability. The standards that correspond to the layers of the ISO-OSI Reference Model for peer-to-peer communication systems once fit on a single wall chart of modest size. Today it is no longer feasible to identify the number of standards that apply to the global communications network on a wall chart of any size. Interoperability will increase in importance as the world grows smaller due to expanding communications networks, and as nations continue to perceive the need to communicate seamlessly across international coalitions of commercial enterprises or national defense forces.

The Øresund Bridge demonstrates the interoperability challenges faced when just two nations collaborate on a project; meshing of regulations on health and safety, and the resolution of two power supply systems for the railway.

8.2.2 Manufacturing and Producibility Analysis

The capability to produce a system element is as essential as the ability to properly define and design it. If a designed product can not be manufactured, this causes design rework and program delays with concomitant cost overruns. For this reason, production engineering analysis and trade studies for each design alternative form an integral part of the Architectural Design process. One objective is to determine if existing proven processes are satisfactory since this could be the lowest risk and most cost-effective approach. The Maglev train contractor experienced a steep learning curve to produce an unprecedented system from scientific theory.

Critical producibility requirements are identified during system analysis and design and included in the program risk analysis, if necessary. Long-lead-time items, material limitations, special processes, and manufacturing constraints are evaluated. When production engineering requirements create a constraint on the design, they are communicated and documented.

Producibility analysis is a key task in developing low cost, quality products. Multidisciplinary teams work to simplify the design and stabilize the manufacturing process to reduce risk, manufacturing cost, lead time, and cycle time; and to minimize strategic or critical materials use. Design simplification considers ready assembly and disassembly for ease of maintenance and preservation of material for recycling. The selection of manufacturing methods and processes are included in early decisions.

Manufacturing analyses draw upon the Concept of Production, Concept of Deployment, and Concept of Maintenance. Manufacturing test considerations are shared with the engineering team and are taken into account in Built-In-Test and Automated Test Equipment.

IKEA is often used as an example of supply chain excellence. IKEA has orchestrated a value creating chain that begins with motivating customers to perform the final phases of furniture assembly in exchange for lower prices and a fun shopping experience. They achieve this through designs that support low cost production and transportability – the bookcase that comes in a flat package and goes home on the roof of a car.

8.3 Configuration Management

The purpose of Configuration Management (CM) is to establish and maintain control of requirements, documentation, and artifacts produced throughout the system's life cycle. The Configuration Management Process is the topic of chapter 5.7.

Change is inevitable, as indicated in Figure 8-2, and managing the impact of change on a project is the work of CM. Systems engineers ensure that the change is necessary, and that the most cost-effective recommendation has been proposed. It is important also to ask: "What is the impact of not making the change?" especially as the system matures, since changes made later in the life cycle have an increasing risk of hidden impacts which can adversely affect system cost, schedule, and technical performance.

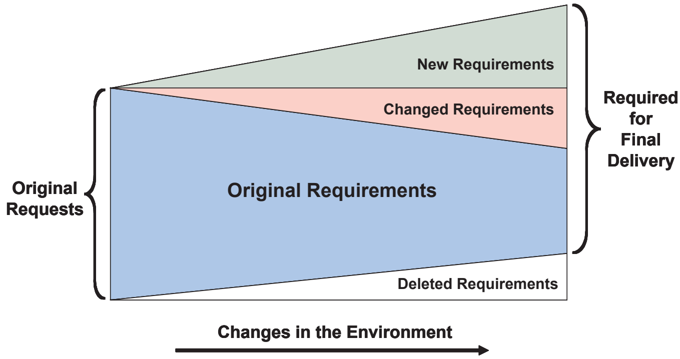


Figure 8-2 Requirements changes are inevitable

CM is the practice of applying technical and administrative direction, surveillance, and services to:

- Identify and document the characteristics of system elements such that they are unique and accessible in some form; assign a unique identifier to each version of each system element;
- Establish controls to allow changes in those characteristics; ensure consistent product versions;
- Record, track, and report status pertaining to change requests or problems with a product; maintain comprehensive traceability of all transactions.

The initial planning efforts for CM are defined in the Configuration Management Plan that establishes the scope of items that are covered by the plan, the resources and personnel skill level required, defines the tasks to be performed, identifies CM tools and processes, as well as methodology, standards and procedures that will be used on the project. Configuration control maintains integrity by facilitating approved changes and preventing the incorporation of unapproved changes into the items under configuration control. Such activities as check-in and check-out of source code, versions of system elements, and deviations of manufactured items, are part of the CM. Independent configuration audits assess the evolution of a product to ensure compliance to specifications, policies, and contractual agreements. Formal audits may be performed in support of decision gate review.

A request to change the current configuration of a system is typically made using an engineering change proposal (ECP). An ECP may originate in a number of ways. The customer may request an ECP to address a change in requirements or a change in scope; an unexpected breakthrough in technology may result in the supplier of a system element proposing an ECP; or a supplier may identify a need for changes in the system under development, and an ECP may originate to address those changes. Circumstances like these that will potentially change the scope or the requirements are appropriate reasons to propose an ECP and to conduct an analysis to understand the effect of the change on existing plans, costs, and schedules. The ECP must be approved before the change is put into effect. It is never appropriate to propose an ECP to correct cost or schedule variances absent of change in scope. A minor change that falls within the current project scope usually does not require an ECP but should be approved and result in the generation of an engineering notice (EN).

The most desirable outcomes of an ECP cycle are (1) that system functionality is altered to meet a changing requirement; or (2) that new technology or a new product extends the capabilities of the system beyond those initially required in ways that the customer desires, (3) that the costs of development, or of utilization, or of support are reduced; or (4) that the reliability and availability of the system are improved. Outcomes (3) and (4) reduce life cycle costs, and potentially save more money than is invested to fund the proposed change.

Evolving system requirements are a reality that must be addressed over the life of a system development effort, and throughout the utilization and support stages of the system. ECPs and ENs help ensure that a system evolves in ways that allow it to continue to satisfy its operational requirements and its objectives and that any modification is known to all relevant personnel. The camera system illustrated in Figure 2-2 is an example of a product family that depends on accurate identification of system elements and characteristics to support the mix & match consumer market.

8.4 Information Management

The purpose of Information Management (IM) is to maintain an archive of information produced throughout the system's life cycle. The Information Management Process is the topic of chapter 5.8.

The initial planning efforts for IM are defined in the Information Management Plan that establishes the scope of project information that is maintained, the resources and personnel skill level required, defines the tasks to be performed, identifies IM tools and processes, as well as methodology, standards and procedures that will be used on the project. Typical information includes source documents from stakeholders, contracts, project planning documents, test documentation, engineering analysis reports, and the files maintained by CM. Today IM is most often concerned with the integration of databases such as the decision database – with results from decision gate reviews and other decisions taken on the project; requirements management tools databases; computer-based training and electronic interactive user manuals; websites and share information spaces accessed over the internet, such as (for example) INCOSE CONNECT. The standard for the exchange of product model data (STEP - ISO 10303) will provide a neutral computer-interpretable representation of product data throughout the life cycle. ISO 10303-239 (Product Life Cycle Support) is an international standard that specifies an information model that defines what information can be exchanged and represented to support a product through life.[7] INCOSE is a co-sponsor of ISO 10303-233 (AP233) - Systems Engineering Data Exchange. Figure 8-3 shows how AP233 would be used to exchange data between a SysML and other Systems Engineering application and then to applications in the larger life cycle of systems potentially using related ISO STEP data exchange capabilities.

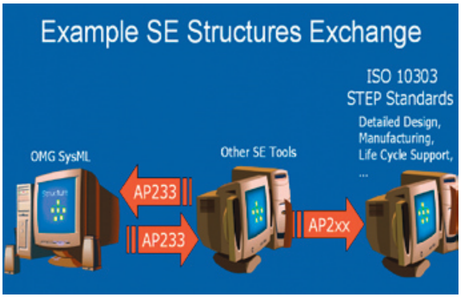


Figure 8-3 AP233 facilitates data exchange

With effective IM, information is readily accessible to authorized project and enterprise personnel. Challenges related to maintaining databases, security of data, sharing data across multiple platforms and organizations, and transitioning when technology is updated are all handled by IM. With all the emphasis on knowledge management, organizational learning and information as competitive advantage, these activities are gaining increased attention.

8.5 Investment Management

The purpose of Investment Management is to balance the use of financial assets within the enterprise. The Investment Management Process is the topic of chapter 6.3.

8.5.1 Define the Business Case

Enterprise management generally demands that there will be some beneficial return for the effort expended in pursuing a project. The business case establishes the scope of required resources (people and money) and schedule, and sets reasonable expectations. An important element of each design gate is a realistic review of the business case as the project matures. The result is re-verification or perhaps restatement of the business case. The Iridium case described in chapter 3.2 illustrates the dangers of failing to keep a realistic perspective. Despite the technological triumph of implementing the world's first Maglev train line, the exorbitant initial cost and the slow return on investment are causing the authorities to question plans to build another line.

The business case may be validated in a variety of ways. For large projects, creation of a sophisticated engineering model, or even prototypes of key system elements, help prove that the objectives of the business case can be met, and that the system will work as envisioned prior to committing large amounts of resources to full-scale engineering and manufacturing development. For very complex systems, such a demonstration can be conducted at perhaps twenty percent of development cost. For smaller projects, when the total investment is modest, proof-of-concept models may be constructed during the Concept Stage to prove the validity of the business case assumptions.

8.5.2 Cost-Effectiveness Analysis

In economics, the term cost-effectiveness applies to the comparison of the relative spending (costs) and outcomes (effects) associated with two or more courses of action. System cost-effectiveness analysis is helpful for deriving critical system performance and design requirements, and supports decision-making. Some examples of critical cost/effectiveness analyses are:

1. Studies of the desirable performance characteristics of commercial aircraft to increase an airline's market share at lowest overall cost over its route structure (more passengers, better fuel consumption)
2. Studies of the desired characteristics of a communications satellite to serve specified markets most economically (placement, coverage)
3. Urban studies of the most cost/effective improvements to a city's transportation infrastructure (buses, trains, motorways, and mass transit routes and departure schedules).

Military and government acquisitions are under the scrutiny of auditing offices to demonstrate that the money spent has delivered the expected benefits.8 A recent concept, Cost as an Independent Variable builds on cost-effectiveness studies to determine an objective cost for the system acquisition. Once the cost is agreed, it becomes a constraint on future decisions regarding project execution.[9]

8.5.3 Life Cycle Cost Analysis

As discussed in chapter 2, decisions made during the early stages of a project inevitably have an impact on future expenditures. New systems are designed, developed, manufactured, and tested over the span of many years, as in the case of a new automobile, or nearly two decades in the case of a submarine. Over such lengths of time, decisions made at the outset may have substantial, long-term effects that are frequently difficult to analyze. (Appendix L.1 elaborates on Life cycle cost analysis.)

Life cycle cost (LCC) analysis is a method of economic evaluation which takes into account all relevant costs of a system over a given period of time adjusting for differences in the timing of those costs. [10] For products purchased off the shelf, the major factors are the cost of acquisition, operation, maintenance, and disposal. Otherwise, it may be necessary to include the costs associated with each of the six life cycle stages. A life cycle cost analysis results in a timetable of expenses so that an organization can cover its costs. If all costs can not be covered, it may not be possible to produce the system.

In some literature, LCC is equated to total-cost-of-ownership. LCC analysis helps the project manager understand the total cost impact of a decision; to compare between program alternatives; and to support trade studies for decisions made *throughout* the system life cycle.

LCC normally includes the following costs, represented in Figure 8-4. Accuracy in the estimates will improve as the system evolves and the data used in the calculation is less uncertain.

1. Research and Development costs
2. Investment (Production/Deployment/Installation) costs
3. Utilization and Support costs
4. Disposal costs

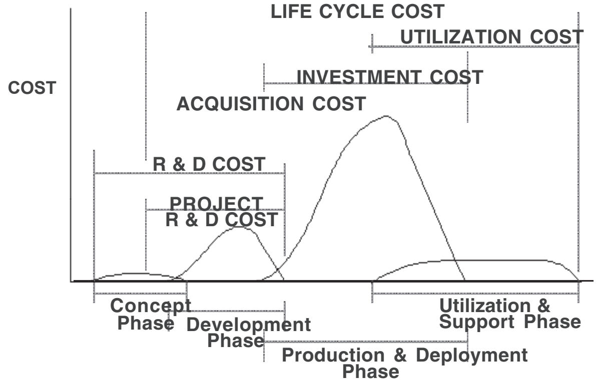


Figure 8-4 Life Cycle Cost Elements (not to scale)

Examples of the justification for LCC analysis

A typical example of a decision with a long-term effect concerns the choice of system elements and parts for a new system. Often, the desire to minimize the initial investment by selecting less expensive parts carries the consequence of a higher probability of failures during the operational life of the system and the corresponding higher maintenance costs.

Therac-25 is an example where the decision to save time and money on software unit testing resulted in undetected design errors that emerged later during operation.

A third example relates to the need for logistics support. Maintenance costs during the operational stage can be reduced by including in the system built-in test equipment that identifies problems, locates their source, and recommends a corrective course of action. Diagnostic testing elements of this type that combine sensors with automated checklists and expert systems logic are expensive to develop, but in the long run decrease maintenance costs and increase availability. A relatively small change in mean-time-to-repair or mean-time-between-failures can result in large cost savings during the Operations Stage.

Advantages of LCC analysis

LCC analysis has three important benefits

- All costs associated with a system become visible: upstream; locked in costs, such as R&D; downstream; customer service.[11]
- Supports an analysis of enterprise interrelationships. Reinforces the importance of locked in costs, such as R&D; low R&D expenditures may lead to high customer service costs in the future.
- Project managers can develop accurate revenue predictions.

Methods / Techniques

1. Wide-band Delphi techniques – estimations from multiple technical and domain experts (estimations only as good as the experts)
2. Analogy - estimating by comparing the proposed project with one or more completed projects that are judged to be similar, with corrections added for known differences (for early estimations)
3. Price-To-Win - focuses on providing an estimate, and associated solution, at or below the price judged necessary to win the contract.
4. Algorithmic (parametric) - uses mathematical algorithms to produce cost estimates as a function of cost driver variables, based on historical data; often supported by commercial tools/models.
5. Design-To-Cost or Cost-As-An-Independent-Variable – based on a design solution that stays within a predetermined set of resources.

8.6 Project Planning

The purpose of Project Planning is to estimate the budget and schedule for a project against which to monitor the progress. This process is the topic of chapter 5.2. As illustrated in Figure 5-1, systems engineers and project managers collaborate in project planning. (Appendix G.1 elaborates on the System Engineering Plan.)

Systems Engineers perform technical management activities consistent with project objectives. Technical management activities include planning, scheduling, reviewing, and auditing the Systems Engineering process as defined in the Systems Engineering Plan (SEP), and the Systems Engineering Master Schedule (SEMS).

The SEP is the top-level plan for managing the Systems Engineering effort. The SEP defines how the project will be organized, structured, and conducted and how the total engineering process will be controlled to provide a system that meets stakeholder requirements. A well-written SEP provides guidance to a project and helps the organization avoid unnecessary discussions about how to perform Systems Engineering. Enterprises generally maintain a template of the SEP suitable for tailoring and reuse. Project-specific appendices are often used to capture detailed and dynamic information such as a schedule of milestones and decision gate reviews, and the methodology to be used in resolving problems uncovered in reviews. Effective project control requires that there be a SEP which the systems engineer keeps current and uses on a daily basis to manage the team's actions.

The SEMS is an essential part of the SEP and a tool for project control. It identifies the critical path of technical activities in the project. Verification activities may also receive special attention in the SEMS. In addition, the schedule of tasks and dependencies helps justify requests for personnel and resources needed throughout the development lifecycle.

The SEP and SEMS are supported by a Work Breakdown Structure (WBS) that defines a project task hierarchy. A description of the enterprise procedures for starting work on a part of the WBS may be defined in the SEP. Under some circumstances, the SEP may address Design to Cost and Value Engineering practices to provide insight into system/cost effectiveness. For example, "Can the project be engineered to have significantly more value with minimal additional cost?" If so, does the customer have the resources for even the modest cost increase for the improvement? The intent is to assure the customer that no obvious cost effective alternatives have been overlooked.

8.7 Quality Management

The purpose of Quality Management is to outline the policies and procedures necessary to improve and control the various processes within the enterprise that ultimately lead to improved business performance. The Quality Management Process is the topic of chapter 6.6.

8.7.1 Process Compliance Reviews

Enterprises that set internal policies and procedures conduct periodic process compliance reviews to assess the effectiveness, strengths, and weaknesses of their processes. Such reviews are conducted on a recurring basis and may be combined with other assessments (such as ISO 9000), to reduce the perceived burden. The review should address defects in the examined process at the time of the review, the improvement process, tailoring of the process, and tailoring of the improvement process (if applicable). Enterprise management prioritizes and approves changes based on requested or recommended areas for improvement in the standard processes. Sometime, it is sufficient to provide additional tailoring guidelines. Results of the review are recorded and maintained.

Occasional benchmark comparisons with other organizations can be useful. Reference processes, practices and other capabilities can be accessed through either direct contact or an intermediary's compilations of benchmarked processes, practices and other capabilities.

This section is not complete without a caution. Leaders in management research advise organizations also to reassess the utility of process management programs and apply them with discrimination.[12] "In the appropriate setting, process management activities can help companies improve efficiency, but the risk is that you misapply these programs, in particular in areas where people are supposed to be innovative. Brand new technologies to produce products that don't exist are difficult to measure. This kind of innovation may be crowded out when you focus too much on processes you can measure."[13]

8.7.2 Quality Assurance

The primary objective of quality assurance (QA) is to produce an end result that meets or exceeds stakeholder expectations. Using a quality system program, manufacturers (for example) establish requirements for each type or family of product to achieve products that are safe and effective. To meet this objective they establish methods and procedures to design, produce, distribute, service, and document devices that meet the quality system requirements. Quality management is the topic of chapter 6.6 and is closely related to the verification and validation processes.

QA is generally associated with activities such as failure testing, statistical control, and total quality control. Many organizations use statistical process control as a means to achieve Six Sigma levels of quality. Traditional statistical process controls use randomly sampling to testing a fraction of the output for variances within critical tolerances. When these are found, the manufacturing processes are corrected before more bad parts can be produced.

Quality experts[14,15] have determined that if quality cannot be measured, it cannot be systematically improved. Assessment provides the feedback needed to monitor performance and make mid-course corrections. It provides data for diagnosing difficulties and pinpointing improvement opportunities. A widely used paradigm for QA management is the Plan-Do-Check-Act approach, also known as the Shewhart cycle.[16]

Quality pioneer W. Edwards Deming stressed that meeting user needs represents the defining criterion for quality and that all members of an organization need to participate actively in "constant and continuous" quality improvement—to commit to the idea that "good enough isn't."[17] His advice marked a shift from inspecting for quality after production to building concern for quality into enterprise processes. As an example, in 1981, Ford launched a quality campaign – see Figure 8-5 – which went beyond getting good workers and supporting them with high-quality training, facilities, equipment, and raw materials. By characterizing quality as a "job," everyone in the organization is motivated to concern themselves with quality and its improvement—for every product and customer.[18]

Figure 8-5 Banner from Ford quality campaign

Total Quality Control deals with understanding what the stakeholder/customer really wants. If the original needs statement does not reflect the relevant quality requirements, then quality can be neither inspected nor manufactured into the product. For instance, the Øresund Bridge consortium included not only the bridge material and dimensions but operating, environmental, safety, reliability and maintainability requirements.

Product certification is the process of certifying that a certain product has passed performance or quality assurance tests or qualification requirements stipulated in regulations such as a building code or nationally accredited test standards, or that it complies with a set of regulations governing quality or minimum performance requirements. Today medical device manufacturers are advised to use good judgment when developing their quality system and apply those sections of the Food and Drug Administration Quality System Regulation that are applicable to their specific products and operations. The regulation, 21-CFR-820.5 is continuously updated since its release in 1996. It ought not to be possible to repeat the errors of the Therac-25 project.

8.8 Resource Management

The purpose of Resource Management is to maintain and manage the people, hardware, and support tools required by the portfolio of enterprise projects. The Resource Management Process is the topic of chapter 6.5.

Resource management is the efficient and effective deployment of an enterprise's resources when and where they are needed. Such resources may include inventory, human skills, production resources, or information technology. An optimised goal is to achieve 100% utilization of every resource, but this is unlikely when providing some minimum level of service while minimizing cost. Resource management relies heavily on forecasts into the future of the demand and supply of various resources.

Project managers face their resource challenges competing for scarce talent in the larger enterprise pool. They must balance access to the experts they need for special studies with stability in the project team with its tacit knowledge and project memory. Today's projects depend on teamwork and optimally multi-disciplinary teams. Such teams are able to resolve project issues quickly through direct communication between team members. Such intra-team communication shortens the decision-making cycle and is more likely to result in improved decisions because the multi-disciplinary perspectives are captured early in the process. However, studies have shown that group decisions are often "riskier" resulting in the potential for greater innovation.

In a multi-disciplinary team, each member comes from a discipline with its own perspective. They are responsible for representing that viewpoint, while at the same time establishing the necessary relationships with the other members. However, team results are condemned to mediocrity unless each member confronts the team with challenging ideas while focusing on the final result. Using concurrent development, see Figure 8-6, it may be possible to finish the work faster and thus return valuable skills back to the personnel pool, having achieved a successful delivery.

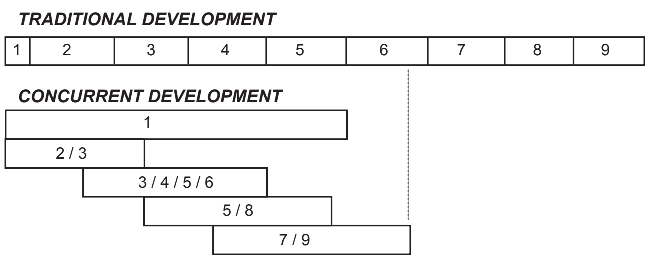


Figure 8-6 Shorter delivery time with concurrent development vs. traditional

As early as 1974, Chase had identified the importance of communications and that "system designs are dependent upon the effective integration of multidisciplinary efforts."[19] He recommended that the organization of a system project should provide opportunity for all disciplinary specialists to work together continuously on a face-to-face basis and, most importantly, to acquire the systems viewpoint and understanding of the role that their specific knowledge can provide in deriving a particular system design.[20]

Chase advocates identifying the lines of communication among tasks in terms of interdependencies and mutual constraints to reveal that different stages of the life cycle call for different tasks and different personnel skills. Properly used, this allows management to acquire and properly utilize the proper combination of specialist and generalist skills. A project avoids "bureaucratization" of the design approach by streamlining the organization and integrating the various specialist backgrounds into common system-oriented task groups with loyalties directed toward the systems design effort.

Modern projects use the concepts of integrated product and process teams to establish a project organization.[21]

8.9 Validation

System validation confirms that the system, as built (or as it will be built), satisfies the stakeholders' stated needs. Validation ensures the requirements and the system implementation provide the right solution to the customer's problem. In other words, "you built the right thing." Validation is the topic of chapter 4.9.

Validation determines that a system does all the things it should and does not do what it should not do. End users and other stakeholders are usually involved in validation activities, but when warranted, an independent third party may be called in to perform validation. Validation may take place either in the operational environment or a simulated operational environment if conditions are hazardous. Both validation and verification activities often run concurrently and may use different portions of the same environment.

Requirements validation is conducted as part of requirements elicitation to provide early assurance that the requirements are the "right" requirements for guiding the development process to a conclusion which satisfies the stakeholders. Requirements validation is often based on requirements analysis; exploration of requirements adequacy and completeness; assessment of prototypes, simulations, models, scenarios, and mock-ups; and by obtaining feedback from customers, users or other stakeholders. Much of the discussion regarding verification (below) also can be applied to validation.

The objects of validation are the designs, prototypes, and final systems elements, as well as the documentation and training materials that describe the system and how to use it. Validation results are an important element of decision gate reviews.

8.10 Verification

System verification addresses whether the system, its elements, and its interfaces satisfy their requirements. Verification ensures the conformance to those requirements; in other words that "you built it right." This is the topic of chapter 4.7.

Verification encompasses the tasks, actions and activities performed to evaluate the progress and effectiveness of the evolving system solutions (people, products and process) and to measure compliance with requirements. The primary function of verification is to determine that system specifications, designs, processes and products are compliant with requirements. A continuous feedback of verification data helps to reduce risk and to surface problems early. The goal is to completely verify system capability to meet all requirements prior to production and operation stages. Problems uncovered at in these stages are very costly to correct, see Figure 2-3. Early discovery of deviations from requirements reduces overall project risk and helps the project deliver a successful, low cost system.[22] Verification results are an important element of decision gate reviews.

Verification analysis can be initiated once a design concept has been established. If a requirements traceability matrix is used, each requirement has a verification activity associated with it. A unique requirements identifier can be used for traceability to the test plans, test procedures, and test reports to provide a closed loop verification process from demonstrated capability back to the requirement. Basic verification activities are:

> **Inspection:** an examination of the item against applicable documentation to confirm compliance with requirements. Inspection is used to verify properties best determined by examination and observation (e.g., - paint color, weight, etc.).
>
> **Analysis**: use of analytical data or simulations under defined conditions to show theoretical compliance. Used where testing to realistic conditions cannot be achieved or is not cost-effective. Analysis (including simulation) may be used when such means establish that the appropriate requirement, specification, or derived requirement is met by the proposed solution.
>
> **Demonstration**: a qualitative exhibition of functional performance, usually accomplished with no or minimal instrumentation. Demonstration (a set of test activities with system stimuli selected by the system developer) may be used to show that system or subsystem response to stimuli is suitable, see Figure 8-7. Demonstration may be appropriate when requirements or specifications are given in statistical terms (e.g., mean time to repair, average power consumption, etc.).
>
> **Test**: an action by which the operability, supportability, or performance capability of an item is verified when subjected to controlled conditions that are real or simulated. These verifications often use special test equipment or instrumentation to obtain very accurate quantitative data for analysis.

A fifth verification method, certification, refers to verification against legal or industrial standards by an outside authority without direction to that authority as to how the requirements are to be verified. For example, this method is used for electronic devices; CE certification in Europe, and UL certification in the US and Canada.

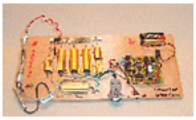


Figure 8-7 A test platform for analyzing battery performance at high loads[23]

The design of the verification activity involves choosing the most cost-effective mix of simulations and physical testing, and integrating test results to avoid unnecessary redundancy. Complete simulation of the system (both performance and design) has become common-place in major system development, and has resulted in reduced development time and cost.

There are four basic test categories. They are:

Development Test: Conducted on new items to demonstrate proof of concept or feasibility.

Qualification Test: Tests are conducted to prove the design on the first article produced, has a predetermined margin above expected operating conditions, for instance by using elevated environmental conditions for hardware.

Acceptance Test: Conducted prior to transition such that the customer can decide that the system is ready to change ownership status from supplier to acquirer.

Operational Test: Conducted to verify that the item meets its specification requirements when subjected to the actual operational environment.

Human Factors Engineers develop task descriptions, operational sequence diagrams, and evaluate the human-system interface to establish the required interactions with the hardware and software. Verification analysis checks that tests have been established using realistic scenarios to demonstrate human reaction times that satisfy operational requirements. Maintainability demonstrations should include a sufficient number of tests and problem areas to provide a high confidence level of meeting maintainability parameters, such as Mean-Time-To-Repair. Production line tests are recommended for items that are new or have not been previously applied to this application. The tests demonstrate producibility and repeatability.

1 Arnold, Stuart and Harold Lawson, "Viewing Systems from a Business Management Perspective: The ISO/IEC 15288 Standard," *Journal of Systems Engineering,* 7:3, 2003.

2 Wasson, Charles S., *System Analysis, Design, and Development,* John Wiley and Sons, NY, NY, 2006. p. 412.

3 Zachman, John A., "A Framework for Information Systems Architecture," *IBM Systems Journal,* vol. 26, no. 3, 1987; IBM Publication G321-5298

4 The Zachman Institute for Framework Advancement (ZIFA) can be found at http://www.zifa.com/.

5 Rechtin, Eberhardt and Mark W. Maier, The Art of Systems Architecting, CRC Press, Boca Raton, FL. Second Edition, 2000. p. 3.

6 Christensen, Clayton M., The Innovator's Dilemma, HarperCollins Publishers, NY, NY. 2000.

7 http://www.plcs-resources.org/

8 http://www.washingtonwatchdog.org/documents/gao/05/GAO-05-966.html

9 Defense Acquisition Deskbook, Version 3.1, 30 September 1999.

10 Fabrycky, W. J. and B. S. Blanchard, *Life-Cycle Cost and Economic Analysis,* Prentice Hall, 1991.

11 Life cycle cost analysis. (2006, May 11). In *Wikipedia, The Free Encyclopedia*. Retrieved May 22, 2006, from http://en.wikipedia.org/w/index.php?title=Life_cycle_cost_analysis&oldid=52726603

12 Womack, J. P., Lean Thinking: Where Have We Been and Where Are We Going?, *Lean Supplement,* Sp 2002, Downloaded May 23, 2006, from http://www.sme.org/gmn/mag/2002/02sem172/02sem172.html

13 Brenner, M. J., TQM, ISO 9000, Six Sigma: Do Process Management Programs Discourage Innovation?, Downloaded May 23, 2006 from http://knowledge.wharton.upenn.edu/index.cfm?fa=printArticle&ID=1321

14 Crosby, Philip B., *Quality is Free,* New York: New American Library, 1979

15 Juran, J. M., ed., *Quality Control Handbook,* Third Edition. New York: McGraw-Hill, 1974

16 *Shewhart, Walter Andrew. Statistical Method from the Viewpoint of Quality Control. NY: Dover, 1939*

17 Deming, W. E., *Out of the Crisis,* M.I.T. Center for Advanced Engineering Study, Cambridge, MA, 1986

18 Scholtes, Peter R., *The Team Handbook, How to Use Teams to Improve Quality,* Joiner Associates, Inc., 1988

19 Chase, Wilton P., *Management of Systems Engineering.* John Wiley & Sons, Inc. 1974.p. 14.

20 Ibid. p. 22.

21 Martin, J. N., *Systems Engineering Guidebook: a process for developing systems and products.* CRC Press. 2000.

22 *Systems Verification, Validation and Testing Methodology Guidelines,* SysTest, Contract: G1RD-CT-2002-00683, http://www.incose.org/secoe/0105.htm

23 Picture downloaded May 22, 2006, from http://www.jk-labs.com/ProdDev.html.

9 Specialty Engineering Activities

The topics in this chapter are covered in alphabetical order by topic title to avoid giving more weight to one topic over another. See Table 1-1 for an overview. The objective of this chapter is to give enough information to systems engineers to appreciate the significance of the engineering specialty area, even if they are not an expert in the subject. It is recommended that subject matter experts are consulted and assigned as appropriate to conduct specialty engineering analysis. More information about each specialty area can be found in references to external sources and the IPAL.

With a few exceptions, the forms of analysis are similar to those associated with systems engineering. Most analysis methods are based on the construction and exploration of models that address specialized engineering areas, such as electro-magnetic compatibility, reliability, safety, and security. Not every kind of analysis and associated model will be applicable to every application domain. Figure 9-1 contains a generic Context Diagram for Specialty Engineering Activities.

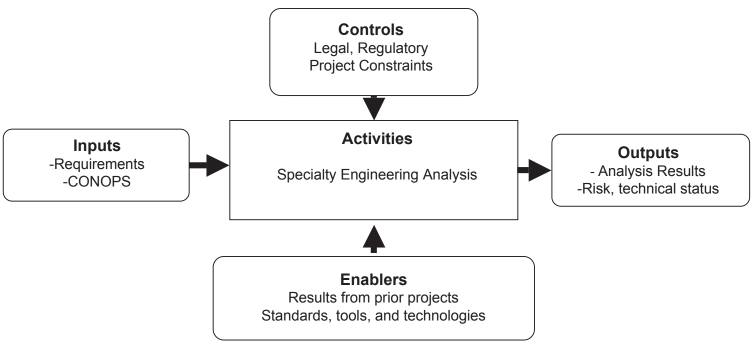


Figure 9-1 Generic Context Diagram for Specialty Engineering Activities

9.1 Design for Acquisition Logistics – Integrated Logistics Support

The Operation and Maintenance Processes are defined in chapters 4.10 and 4.11, respectively. The sustainment of these processes during the Utilization and Support Stages is dependent on actions set in motion during the earlier stages. These activities are known under various titles, such as Integrated Logistics Support (ILS), Supply Chain Management, Product Support, Customer Services, and similar names. The full scope of ILS includes Acquisition Logistics – activities influencing the design of a system and its readiness for utilization and support; and, Operational Logistics – activities that ensure that the right materiel and resources, in the right quantity and quality, are available at the right place at the right time throughout the Utilization and Support Stages. Operational logistics also receives attention under the heading of supply chain management. This handbook uses the term ILS, and this section focuses on Acquisition Logistics. Strategies and tactical plans for ILS are established as part of the Enterprise

Processes, and will drive the basic considerations to be applied during the Acquisition Logistics activities.

Many different analyses are used to consider whether it is more cost effective to influence the initial design of the system or to plan for spare parts and repairs during utilization. When initial acquisition costs are fixed, this can have a downstream impact on the funding that will be needed in future years.

9.1.1 "-ilities" influencing the system design

Acquisition Logistics focus on design requirements criteria, applicable to all system elements, and comprise (but is not limited to) the following list of engineering specializations: Affordability (Life Cycle Cost); Cost/System Effectiveness; Disposability (Recycling / Retirement); Maintainability; Packaging, Handling, Storage and Transportation (PHS&T); Producibility (Manufacturability); Reconfigurability (Flexibility / Standardization); Reliability; Security; Supportability (Serviceability); Survivability; and, Vulnerability – sometimes referred to as the "-ilities". Figure 9-2 illustrates the relationship between ILS analysis activities.

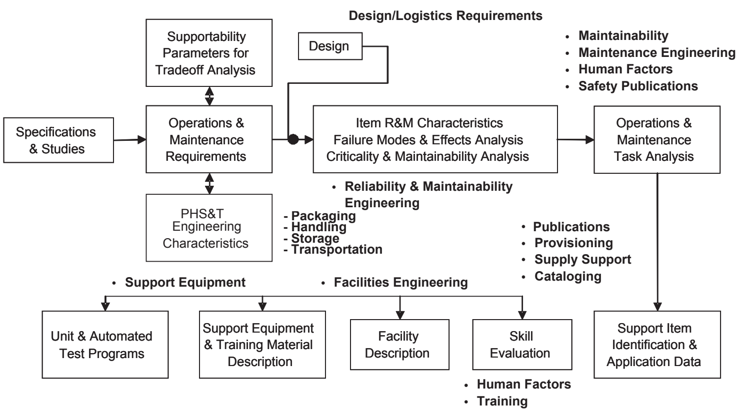


Figure 9-2 Acquisition Logistics Activities

The Availability, Reliability and Maintainability of a system are major drivers in the use of support resources and the related in-service costs. The probability that the system, when used under stated conditions, will operate satisfactorily is often expressed as Availability, which in turn, is dependent on the design for Maintainability and Reliability as well as the support arrangements during the Utilization and Support Stages. However, the term Availability in itself is imprecise as it may, or may not include logistics and administrative delay time, corrective and preventive maintenance. Therefore, we categorize Availability into: Inherent, Achieved, or Operational Availability. Reliability is concerned with the probability of the system-of-interest working when it should. Maintainability is concerned with keeping the system working and the ease of putting things right once they have gone wrong.[1]

A summarizing denomination for Availability, Reliability, Maintainability and Supportability, may also be known as Dependability. The Dependability of a system describes its ability to fulfill the required performance under given conditions, taking degradation of performance due to failure and maintenance into consideration. Under certain conditions, it may be necessary to introduce redundancy into the design of a system to enhance the system's reliability by providing two or more functional paths or physical objects in areas that are critical to successfully achieve specified Availability under given conditions. Likewise, it is important to eliminate single points of failure during the design of a system.

In the discussion of the Øresund Bridge case, the survivability of the bridge in the event of ship collisions, and analysis of the design to ensure that traffic flow is not interrupted by accidents are two examples of ILS-related trade studies that were conducted.

Another important factor to consider during the design of a system is the Packaging, Handling, Storage and Transportation (PHS&T), which includes all special provisions, materials and containers, and how the system or the parts thereof shall be handled, distributed and stored. In addition to the system itself, PHS&T also covers spares and consumables.

Packaging can occasionally be more expensive than the product itself. Packaging requirements often have conflicting user objectives. A merchant wants the package to be so tough that a shoplifter cannot steal parts from inside the package while it is on the store shelf. The consumer wants the package to be easy to open when they get it home. Experience as a consumer suggests that the merchant won; many packages are almost impossible to open without using tools.

Handling can be a source of failure. A warehouse worker can – and sometimes does – drive a folk-lift through a package. Airline baggage handlers are known to throw boxes onto carts, even though they are marked "fragile."

The storage environment can impose significant constraints on packaging and the system itself. Humidity, dust, and temperature are typical environmental concerns. In addition, warehousing places constraints on size and density of objects.

Transportation is often overlooked when designing a system. Use of freight trains or cargo planes imposes limits on height, width, length and weight. For some systems, these constraints may force building multiple system elements that are not assembled until delivered to the operational site. For example after completing a new post office building in a major USA city at a cost of $140 million, the city engineers found that post office trucks would not fit into the enclosed loading dock. Similarly, food trucks that normally service Denver Airport's restaurants are too high to fit under the elevated walkways that cross over their access roads.

Acquisition Logistics / ILS is important and relevant to systems engineers. As consumers themselves, systems engineers have experienced the frustration of unreliable systems. Those who have maintained systems at some point in their careers know that failure to acknowledge that humans maintain systems can cause significant problems in an operational environment. When availability gets too bad, systems are sometimes simply shut down in frustration, which can cause significant rework at considerable cost. The huge variety of Acquisition Logistics analyses are best carried out by subject matter experts, because they are familiar with the mathematics that underlie the techniques, with the tools that are available to support these analyses, and with the factors that influence the outcome of these analyses.

9.1.2 "-ilities" analysis methods

Within the fields of Reliability, Maintainability and Supportability Engineering there are several analyses that are performed iteratively and recursively as they are co-dependent on results of other analyses. This section briefly addresses some of the most useful and common analyses techniques.

Failure Modes Effects and Criticality Analysis (FMECA) should be performed early enough to influence equipment design. The aim is to minimize maintenance requirements and thereby cost. FMECA indicates that potential failures may occur that either: cannot be removed through re-design but can be avoided through preventive maintenance; or have a non-critical impact and therefore can be allowed to occur, with subsequent rectification through corrective maintenance.

FMECA is a means of recording and determining what functions the equipment is required to perform, and

- How these functions could fail
- Possible causes of the failures
- Effects the failures would have on the equipment or system
- The criticality of the failures

Level of Repair Analysis is the process of evaluating system elements to first determine (in most cases from an economic point of view, only) if the element or system should be discarded or repaired. If repairing the item is feasible, establish where the repair should take place; e.g. at home, locally, or at the factory. This is expressed as an organizational level. This analysis is conducted throughout the system life cycle. The handling of a system element may change based on experiences from prior decisions.

Logistic Support Analysis (LSA) / Supportability Analysis is a structured method of analyzing the support implications of system elements as they are being developed, with the aim of identifying features of the design that could result in excessive expense during the operational life of the system. Once identified, these items can be the subject of trade-offs to revise the design in order to reduce later costs. Once the design is more fully defined, during the late activities of the Development stage, the LSA can identify all the logistic resources necessary to support the equipment and the impact on the existing support infrastructure. LSA is only cost effective where it is likely to generate benefit in terms of a more supportable design or better defined support requirements and hence reduced life cycle cost.

Reliability Centered Maintenance Analysis (RCM) can be performed to assess the most cost efficient preventive maintenance program for the system. RCM is best initiated very early in the Development Stage and evolves throughout the Production Stage. As such it can also successfully be introduced for systems already in operation, as it can be accomplished using a decision tree, leading the analyst through a logical sequence of the nature and frequency of applicable preventive maintenance tasks.

Survivability Analysis is performed when items must perform critical functions in a hostile operational environment. Threats to be considered include conventional, electronic, nuclear, biological, chemical, and other weapons, and terrorism or sabotage, erratic human behavior, and harsh environmental conditions, such as ocean salinity. Critical survivability characteristics are identified assessed, and analyzed to evaluate their impact on system performance and effectiveness.[2] A system is said to be survivable if it can fulfill its purpose in a timely manner, even in the presence of attacks or failures. Because of the severe consequences of failure, enterprises increasingly focus on system survivability as a key risk topic.

The Spitfire (see Figure 9-3) was designed with an elliptical wing, giving greater speed and maneuverability (perhaps the most critical -ility of all for a warplane). This was at a price: 13000 man-hours per airframe. Willy Messerschmitt had optimized for speed and manufacturability: only 4000 man-hours; but the Bf 109 was no faster than the Spitfire, and was consistently out-turned by it. The elliptical wing had been considered, but rejected it as too difficult to manufacture. [3]

Figure 9-3 The Spitfire: A perfect balance of -ilities?

System Security Analysis identifies and evaluates system vulnerabilities to known or postulated security threats, and recommends means to eliminate the vulnerabilities or to at least reduce the susceptibility to compromise, damage, or destruction to an acceptable level of risk.

9.2 Electromagnetic Compatibility Analysis

Electromagnetic compatibility (EMC) analysis is performed on electric or electronic items to ensure that they can perform in their intended electromagnetic environments. Analysis also ensures that items that are intentional radiators of radio frequency energy comply with commercial, governmental, and relevant international policies for radio frequency spectrum management and do not interfere with other signals – Electromagnetic interference (EMI). Even cable or speaker wire routing for home devices, such as a television, must consider EMI/EMC to achieve maximum performance and ensure safety of the users.

9.3 Environmental Impact Analysis

Europe, the USA and many other nations recognize regulations that control and restrict the environmental impact that a system may inflict on the biosphere. The ISO 14000 series, Environmental Management[4] standards are an excellent resource for analysis and assessment methods for the protection of the environment. Failure to comply with environmental protection laws carries penalties and may result in the system not being approved for development. The issue is discussed in several references.[5, 6]

The focus of environmental impact analysis is on potential deleterious effects of a proposed system's development, construction, use, and disposal. All countries that have legally expressed their concern for the environment restrict the use of hazardous chemicals and components (e.g. mercury, lead, cadmium, chromium 6, and radioactive materials) with a potential to cause human disease or to threaten endangered species through loss of habitat or impaired reproduction. Concern extends over the full life cycle of the system to be developed, as is made evident by the European Union's resolution to adopt within 2006 a legal restriction that system developers and their component suppliers retain lifetime liability for decommissioning systems that they build and sell.

The Øresund Bridge is an example of how early analysis of potential environmental impacts ensures that measures are taken in the design and construction to protect the environment with positive results. Two key elements of the success of this initiative were the continual monitoring of the environmental status and the integration of the environmental concern into the requirements from the Owner.

Disposal analysis is a significant analysis area within Environmental Impact Analysis. Traditional landfills for non-hazardous solid wastes have become less available within the large city areas and the disposal often involves transporting the refuse to distant landfills at considerable expense. The use of incineration for disposal is often vigorously opposed by local communities and citizen committees, and poses the problem of ash disposal; the ash from incinerators is sometimes classified as hazardous waste. Local communities and governments around the world have been formulating significant new policies to deal with the disposal of non-hazardous and hazardous wastes.

One goal of the architecture design is to maximize the economic value of the system-elements residue and minimize the generation of waste materials destined for disposal. Because of the potential liability that accompanies the disposal of hazardous and radioactive materials the use of these materials is carefully reviewed and alternatives used where and whenever possible. The basic tenet for dealing with hazardous waste is the "womb-to-tomb" control and responsibility for preventing unauthorized release of the material to the environment. This may include designing for reuse, recycling, or transformation (e.g. composing, bio-degradation).

In accordance with United States and European Union laws, system developers and component manufacturers must analyze the potential impacts of the systems that they construct, and must submit the results of that analysis to government authorities for review and approval to build the system. Failure to conduct and submit the environmental impact analysis can result in severe penalties for the system developer, and may result in an inability to build or to deploy the system. It is best when performing environmental impact analysis to employ subject matter experts who are experienced in conducting such assessments and submitting them for governmental review.

9.4 Human Systems Integration

Human Systems Integration (HSI) is the interdisciplinary technical and management processes for integrating human considerations within and across all system elements; an essential enabler to systems engineering practice. HSI focuses on the human, an integral element of every system, over the system life cycle. It is an essential enabler to systems engineering practice as it promotes a "total system" approach which includes humans, technology (hardware, software), the operational context and the necessary interfaces between and among the elements to make them all work in harmony. The "human" in HSI includes all personnel (system owners, users/customers, operators, maintainers, support personnel, trainers, etc.) who interact with the system in any capacity. HSI brings human-centered disciplines and concerns into the SE process to improve the overall system design and performance.

HSI ensures that the individual, but very independent, human-centered domains are integrated throughout system design, development, manufacturing and coding, operation, sustainment, and disposal. Currently, the most widely recognized human-centered domains identified for consideration in the HSI process are: Manpower, Personnel, Training, Human Factors Engineering, Environment, Safety, Occupational Health, Survivability, and Habitability. These interdependent HSI domains must be considered simultaneously because decisions made in one domain can have significant impacts on other domains.

Further details on HSI are provided in Appendix M: Human Systems Integration.

Human Performance and Human Engineering Design Requirements

The Human Engineering or Human Systems Engineering effort affects every portion of the system that has a human-machine interface. It is essential to integrate human system factors into the design of items.

During requirements analysis, requirements from a variety of sources and disciplines are analyzed to resolve conflicts. The human factors engineer is primarily responsible for two types of requirements; human performance requirements and human engineering design requirements. Human performance requirements include times and accuracies for tasks assigned to humans. The human factors engineer ensures that the proposed requirements are in fact achievable by the intended operators and users. The human engineer may in some cases, define the human performance requirements based on external requirements, specifications of other system elements, or the capabilities and limitations of the prospective operators and users.

Human performance requirements are frequently derived from or at least bounded by other performance requirements within the system. The accuracy, response time, and other attributes of the operator tasks will affect similar attributes at the system level. The implementation of the requirements needs to be verified, and additional design decisions need to be made as the design progresses.

The human engineering design requirements concern specific aspects of the hardware and software that are necessary to fit the operators and assist them in their assigned tasks. These requirements define what must be designed and constructed to permit the operators and users to interact with one another and the rest of the system. Such requirements commonly address topics that make the work area more effective (use of colors, button and knob design and layout, etc.). It is generally good practice to minimize characteristics that require extensive cognitive, physical, or sensory skills; require the performance of unnecessarily complex tasks; require tasks that result in frequent or critical errors.

Ergonomics is the name of the engineering discipline concerned with the elimination of aspects of a system design that could cause temporary or permanent injury to people who operate, maintain, or otherwise use the system. This may include identification of steps people can take to reduce the risk of injury when operating, maintaining, or otherwise using the system after it is deployed. It is also a matter of ethics that systems do not present undue risks to the people who will use them. The ergonomics engineering process begins during the Concept Stage of the system life cycle and continues throughout the life of the system. Figure 9-4 identifies a three-step process to reduce the risk that a system will require costly rework in order to be authorized to deploy, or fail to be allowed to deploy at all; (1) identify the key design considerations and address them in step 3, during development of the system, (2) build the right team, and (3) manage the human factors engineering process.

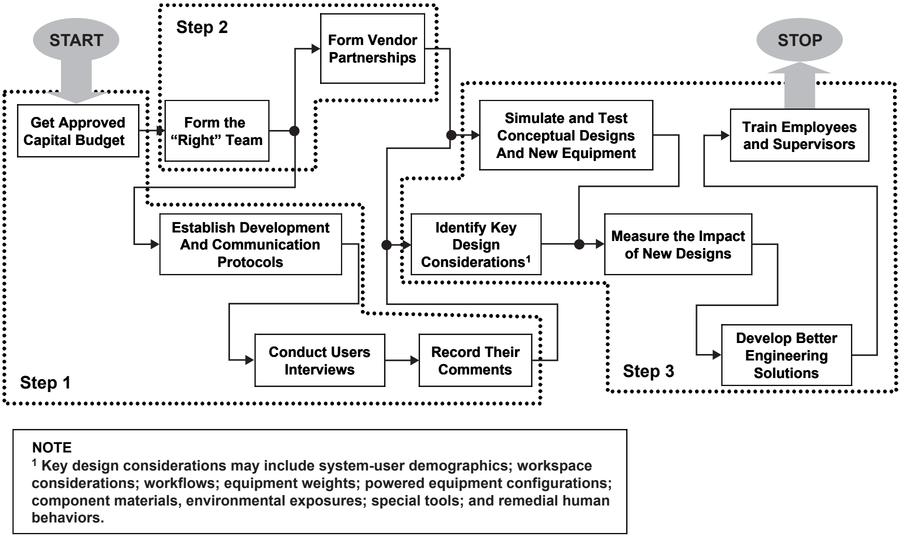


Figure 9-4 Ergonomics Engineering Minimizes Risks to System Stakeholders[7]

The Occupational Safety and Health Agency has a website[8] that features eight eTools that address ergonomics for a number of industries and occupations, including baggage handling, beverage delivery, computer workstations, grocery warehousing, health care, poultry processing and sewing.

The design shown in Figure 9-5 may meet simple requirements for a teapot, but no human would want to use it. The author discusses many examples of implemented systems from calculator keypads with keys in a non-intuitive location to department store door handles where there is no indication where to push to open the door. The consistent issue in these instances is a lack of human factor considerations in the design. In many cases, styling and appearance are allowed to override good engineering, such as the car dash board design in which the identically shaped handles for brake release and hood release are placed side-by-side under the dash with no easily distinguished marking on either handle. Rental car drivers beware!

Figure 9-5 Unique Teapot Shown on Book Cover[9]

9.5 Mass Properties Engineering Analysis

Mass Properties Engineering[10] (MPE) is done to assure that the system or system element has the appropriate mass properties to meet the requirements. The mass properties include weight, the location of center of gravity, inertia about the center of gravity, and product of the inertia about an axis.

Typically, the initial sizing of the physical system is derived from other requirements, such as minimum payload, maximum operating weight, or human factors restrictions. Mass properties estimates are done at all stages of the system life cycle, based on the information that is available at the time, which may range from parametric equations to a three-dimensional product model, to actual inventories of the product in service. A risk assessment is done using techniques such as uncertainty analysis or Monte Carlo simulations to verify that the predicted mass properties of the system will meet the requirements, and that the system will operate with in its design limits. Validation is usually done at the end of the Production Stage to assure all parties that the delivered system meets the requirements, and then several times during the Utilization Stage to assure safety of the system, component or human operator. For a multi-billion dollar project such as oil platform or warship the MPE level of effort is significant.

One trap in MPE is that design mangers may believe that their 3-D modeling tools can be used to estimate the mass properties of the system or system element. This is problematic because: (1) not all parts are modeled on the same schedule, (2) most parts are modeled neat, that is without such items as manufacturing tolerances, paint, insulation, fittings etc. which can add from 10 to 100% to the system weight. For example, the liquid in piping and tanks can weigh more than the structural tank or metallic piping that contain it.

MPE usually includes a reasonableness check of all estimates by using an alternative method. The simplest method is to justify the change between the current estimate and any prior estimates for the same system, or the same system element on another project. Another approach is to use a simpler estimating method to repeat the estimate then justify any difference.

9.6 Modeling, Simulation, and Prototyping

Modeling, simulation, and prototyping used during architecture design can significantly reduce the risk of failure in the finished system. These techniques enable the development of complex and costly enabling systems, such as a flight simulator or a high-volume production line, which allow validation of the system's concepts, or supports training of personnel in ways that would otherwise be cost prohibitive. Systems engineers use modeling and simulation on large complex projects to manage the risk of failure to meet system mission and performance requirements. This form of analysis is best conducted by subject matter experts who develop and validate the models, conduct the simulations, and analyze the results.

9.6.1 Modeling and Simulation

Modeling and simulation are an effective and usually efficient way to address technical risk on a project, especially a large project, because they represent a cost-effective means to find and fix problems before development is concluded and production begins. Modeling helps generate data in the domain of the analyst or reviewer, not available from existing sources, in a manner that is affordable and timely to support decision-making. The objective of modeling is to obtain information about the system before significant resources are committed to its design, development, construction, testing, or operation. Consequently, development, validation, and operation of the model must consume time and resources not exceeding the value of the information obtained through its use. (Appendix L.3 elaborates on systems modeling.)

A model is a mapping of the system-of-interest onto a simpler representation which approximates the behavior of the system-of-interest in selected areas. Models may be used to represent the system under development, the environment in which the system operates, or interactions with enabling systems and interfacing systems.

Models can be used within most Systems Life Cycle Processes, such as the following.

- Requirements Analysis: determine and assess impacts of candidate requirements
- Architectural Design: evaluate candidate options

- Verification: simulate the system's environment and evaluate test data
- Operations: simulate operations in advance of execution for planning and validation

The result of modeling is to predict characteristics (performance, reliability, operations, and cost, etc.) across the spectrum of system attributes throughout its life cycle. The predictions are used to guide decisions about the system's design, construction, and operation, or to verify its acceptability. Standard tools for all types of modeling are now available commercially for a wide range of system characteristics.

9.6.2 Types of Models

Models fall into one of two general categories – representations and simulations. Representations employ some logical or mathematical rule to convert a set of inputs to corresponding outputs with the same form of dependence as in the represented system, but do not mimic the structure of the system. Validity depends on showing, through analysis or empirical data, that the representation tracks the actual system in the region of concern.

Simulations, on the other hand, mimic the detailed structure of the simulated system. They are composed of representations of system elements, connected in the same manner as in the actual system. The validity of a simulation depends on validity of the representations in it and the faithfulness of its architecture to the actual system. Usually the simulation is run through scenarios in the time domain to simulate the behavior of the real system. An example might be the simulation of a fluid control system made up of representations of the piping, pump, control valve, sensors, control circuit, and the fluid running through the system.

The type of model selected depends on the particular characteristics of the system which are of interest. Generally, it focuses on some subset of the total system characteristics such as timing, process behavior, or various performance measures. Representations and simulations may be made up of one or several of the following types: Physical, Graphical, Mathematical (deterministic), and Statistical.

Physical models exist as tangible, real-world objects which are identical or similar in the relevant attributes to the actual system. The physical properties of the model are used to represent the corresponding properties of the actual system. Examples of physical models include: wind tunnel, testbed, and breadboard/brassboard.

Graphical models are a mapping of the relevant attributes of the actual system onto a graphical entity with analogous attributes. The geometric or topological properties of the graphical entity are used to represent geometric properties, logical relationships, or process features of the actual system. Examples of graphical models include: functional flow block diagrams, N^2 diagrams, logic trees, blueprints, schematics, and maps.

Mathematical (deterministic) models use closed mathematical expressions or numerical methods to convert input data to outputs with the same functional dependence as the actual system. Mathematical equations in closed or open form are constructed to represent the system. The equations are solved using appropriate analytical or numerical methods to obtain a set of formulae or tabular data defining the predicted behavior of the system. Examples of mathematical models include: operational or production throughput analysis; thermal analysis; vibration analysis; load analysis; stress analysis; eigen value calculations; and linear programming.

Statistical models are used to generate a probability distribution function for expected outcomes, given the input parameters and data. Statistical models are appropriate whenever truly random phenomena are involved as with reliability estimates, whenever there is uncertainty regarding the inputs such that the input is represented by a probability distribution, or whenever the collective effect of a large number of events may be approximated by a statistical distribution. Examples of statistical models include: Monte Carlo; logistical support; discrete and continuous models.

A simulation can used to quickly examine a range of sizes and parameters, not just a "Point Design." This will insure that the "best" solution is obtained – the system is the proper size throughout, with no choke points. Exercise the simulation using scenarios extracted from the concept of operations with inputs based on system requirements. Monte Carlo runs may be made to get averages and probability distributions. In addition to examining nominal conditions, non-nominal runs should also be made to establish system reactions or breakage when exposed to extraordinary (out-of-spec) conditions.

9.6.3 Prototyping

Prototyping is a technique that can significantly enhance the likelihood of providing a system that will meet the user's need. In addition, a prototype can facilitate both the awareness and understanding of user needs and stakeholder requirements. This section will discuss briefly two types of prototyping; rapid and traditional.

Rapid prototyping is probably the easiest and one of the fastest ways to get user performance data and evaluate alternate concepts. A rapid prototype is a particular type of simulation quickly assembled from a menu of existing physical, graphical, or mathematical elements. Examples include tools such as laser lithography or computer simulation shells. They are frequently used to investigate form and fit, human-system interface, operations, or producibility considerations. Rapid prototypes are widely used, and are very useful, but, except in rare cases they are not "prototypes."

Traditional prototyping is a tool that can reduce risk or uncertainty. A partial prototype is used to verify critical elements of the system-of-interest. A full prototype is a complete representation of the system. It must be complete and accurate in the aspects of concern. Objective and quantitative data on performance times and error rates can be obtained from these higher fidelity interactive prototypes.

The original use of a prototype was as the first of a kind from which all others were replicated. However, prototypes are not "the first draft" of production entities. Prototypes are intended to enhance learning and should be set aside when this purpose is achieved. Once the prototype is functioning, changes will often be made to improve performance, or reduce production costs. Thus, the production entity may require different behavior. The Maglev train system may be considered a prototype (in this case, proof-of-concept) for longer distance systems that will exhibit some but not all of the characteristics of the short line. Scientists and engineers are in a much better position to evaluate modifications that will be needed to create the next system.

9.7 Safety & Health Hazard Analysis

Safety and health hazards are hazards to the well-being of human operators, maintainers, administrators, or other users of a system. They are a major concern[11] wherever hazardous materials are employed, such as the chemical industries, building enterprises, medical and radiological equipment supply concerns, energy production, and aviation and space. The objective of the system safety effort is to ensure that the system meets an accepted level of risk. Assessing safety risk is almost always necessary for contractual and legal reasons.

A systems engineer in one of the cited industries or any number of other industries that deal with hazardous materials, processes or human activities must be aware that subject matter experts and system safety engineers are available to perform the analyses that can identify these hazards and their attendant risks, and can help identify means to eliminate or at least mitigate the risks to acceptable levels. The system safety program begins in the conceptual stage, and continues throughout the system life cycle, as illustrated in Figure 9-6.

Concept Stage		**Development Stage**		**Production Stage**	**Utilization Stage / Support Stage**		**Retire-ment Stage**
Establish Safety Objectives	Initial Hazard Analysis	Update Hazard Analysis	Safety Verifi-cation	Sys Safety Assessment & Certification	Maintenance of Safety Baseline	Mishap Investigation and Correction	Safe Disposal

Figure 9-6 System safety focus during the system life cycle

Safety risks are associated with such processes as complex machinery used in a manufacturing plant, or high-temperature metals in a steel plant, or coal mining, or maintenance of deep sea platforms (among others); or with activities such as flying, or space travel or deep sea fishing (among others). While a safety decision tree can be a useful starting place to analyze processes and activities as well as physical components of systems, it is likely that the means to eliminate or reduce process and activity risks will be different. Construction of safety cages can protect people in a complex manufacturing work cell; "kill" buttons can be installed; and barriers can be constructed to make sure a person cannot fall (for example) into molten steel; specialized training and back-up safety equipment can available to (for example) divers that maintain off-shore oil rigs. The Therac-25 case illustrates the cost in human life that may result when adequate measures are not taken to build safety measures into potentially dangerous equipment. The specially designed windows for the Maglev train dampen the noise level that would otherwise present a hazard to passengers.

When the hazards are caused by materials used within the system, it is crucially important to isolate the materials by some safe means as they are used in the system, and to plan for their eventual substitution by non-hazardous materials as material science advances. See Figure 9-7 for examples of protective clothing.

Figure 9-7 Protective clothing for Hazmat Level A and bird flu

Many governments have regulations that mandate that all hazards to human safety and health be reduced as far as is possible, and that all safety and health hazards that can not be eliminated are mitigated by other than system means to reach acceptable levels of risk. This means avoiding wherever possible the use of hazardous materials, containing hazardous material that cannot be eliminated, and addressing the hazards associated with process and human activities that are required to support and maintain the system in its operational environment. It also means planning for the safe handling and disposal of hazardous materials, and including such effort in the life cycle cost models and cost forecasts for the system being developed.

9.8 Sustainment Engineering Analysis

Sustainment engineering helps ensure that a system continues to satisfy its objective over its intended lifetime. In that timeframe, system expectations will expand, the environments in which the system is operated will change, technology will evolve, and elements of the system may become unsupportable and need to be replaced. The desktop computing environment is a case in point. Today it is nearly impossible to find cables to support parallel port printers since the introduction of the Universal Serial Bus (USB).

Sustainment Engineering is an integrated effort designed to address industry needs regarding aging systems, and a need to maintain those systems in operation. A sustainment program may include re-engineering electronic and mechanical components to cope with parts obsolescence, the development of automated test equipment, and extending the life of aging systems through technology insertion enhancements, and proactive maintenance. These changes will have significant impact on ILS analyses.

9.9 Training Needs Analysis

Training needs analyses support the development of products and processes for training users and maintainers of a system. Training analysis includes the development of personnel capabilities and proficiencies to accomplish tasks at any point in the system life cycle to the level they are tasked. These analyses address initial and follow-on training necessary to execute required tasks associated with system use and maintenance. An effective training analysis begins with a thorough understanding of the concept documents and the requirements for the system-of-interest. A specific list of functions or tasks can be identified from these sources, and can be represented as learning objectives for operators, maintainers, administrators and other users of the system. The learning objectives then determine the design and development of the training modules and their means of delivery.

Important considerations in the design of training include who, what, under what conditions and how well each user must be trained and what training will meet the objectives. Each of the required skills identified must be transformed into a positive learning experience and mapped onto an appropriate delivery mechanism. The formal classroom environment is rapidly being replaced with or augmented by simulators, computer-based-training, internet-based distance delivery, and in-systems electronic support, to name a few. Updates to training content use feedback from trainees after they have some experience to improve training effectiveness.

1 Blanchard, Ben and Wolter Fabrycky, *Systems Engineering and Analysis, 3rd ed.,* Prentice Hall, 1998. Ch. 12-13 include complete discussions of the metrics and calculations for Availability, Reliability and Maintainability.

2 For discussion of Survivable Systems Analysis Methods see http://www.cert.org/archive/html/analysis-method.html

3 Alexander, Ian, Systems Engineering: -ilities for Victory, Downloaded from http://easyweb.easynet.co.uk/iany/consultancy/systems•engineering/ilities•for•victory.htm

4 The ISO 14000 family of International Standards, www.iso.ch/iso/en/prods-services/otherpubs/iso14000/index.html

5 Botkin, Daniel B. and Edward A. Keller, *Environmental Science: Earth as a Living Planet, 2nd edition,* New York: John Wiley Sons, 1998.

6 Mary Edwards, an Assistant Professor at the University of Wisconsin, Madison, developed a guide that includes a chapter on environmental impact analysis. It can be found at http://www.lic.wisc.edu/shapingdane/facilitation/all•resources/impacts/analysis•environmental.htm

7 Copyright© 2004 by Marsh Inc., www.marshriskconsulting.com/st/PDEv•C•371•SC•228135•NR•306•PI•233074.htm.

8 The OSHA web site can be found at www.osha.gov/SLTC/ergonomics/

9 Norman, Donald A., *The Design of Everyday Things,* Doubleday, New York, NY 1988

10 See Recommended Practices from the Society of Allied Weight Engineers at www.sawe.org.

11 US Department of Labor, Job Hazard Analysis, Washington, DC: OSHA 3071, 2002 (available on-line at http://www.osha.gov/Publications/osha3071.pdf).

THIS PAGE
INTENTIONALLY
LEFT BLANK

10 Tailoring Overview

10.1 Introduction

Standards and handbooks are written to address generic practices that may, or may not, apply to a given organization or system-of-interest. Most are accompanied by a recommendation to adapt the processes and activities to the situation at hand. This adaptation is called tailoring.

Throughout this handbook, advice has appeared about the formal use of these processes. Formality is highly dependent on the sophistication of the system, the organizations and the work to be accomplished.

Tailoring scales the rigorous application of these processes to an appropriate level based on need and the system life cycle stage. For example, tighter assessment and control cycles are typical of earlier stages of the system life cycle.

The principle behind tailoring is to establish an acceptable amount of process overhead committed to activities not otherwise directly related to the creation of the system. Oppressive overhead, with no visible value-added contributions, is demoralizing, and may result in a system that costs more than it is worth. Insufficient process results in uncoordinated human effort and thrashing [1] – which also adds cost.

This chapter describes the process of tailoring this handbook to meet your needs.

10.2 Tailoring Process

Figure 10-1 is a notional graph for balancing formal process against the risk of cost and schedule overruns. As discussed in Chapter 2, insufficient systems engineering effort is generally accompanied by high risk. McConnell[1] describes the improvements in efficiency realized by adding process. But as the graph illustrates, too much formal process also introduces high risk.

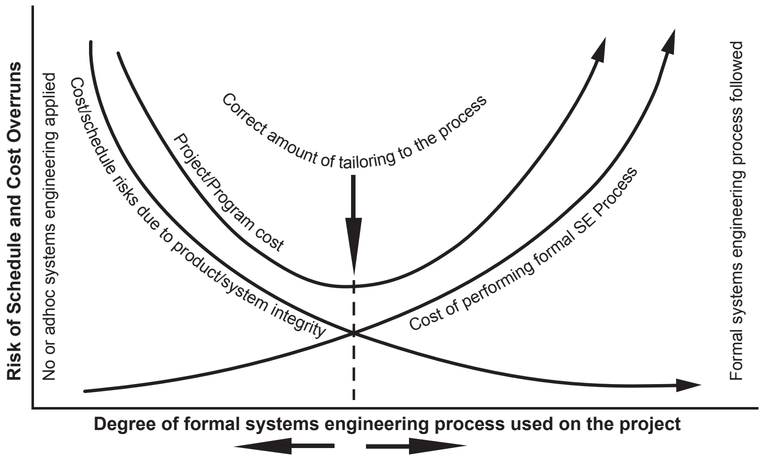


Figure 10-1 Tailoring requires balance between risk and process[2]

If too many or unnecessary processes are performed, increased cost and schedule impacts will occur with little or no added value to the integrity of the system. Tailoring processes is dynamic over the system life cycle depending on risk and the situational environment and should be continually monitored and adjusted as needed. Figure 10-2 is the Context Diagram for the Tailoring Process described in this chapter.

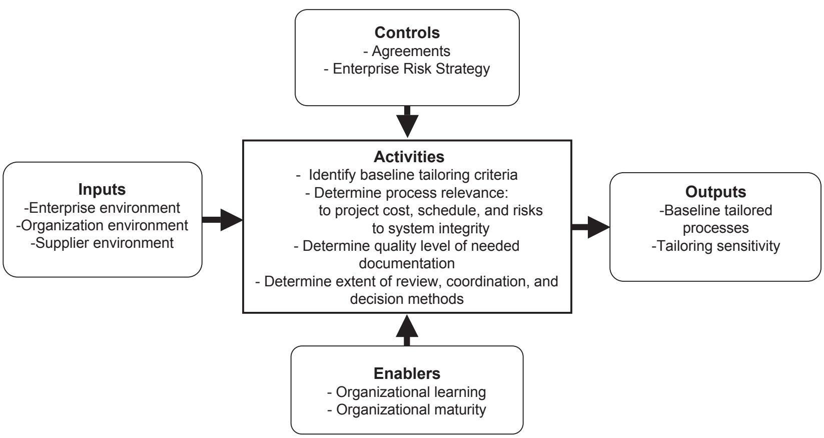


Figure 10-2 Tailoring Process Context Diagram

10.2.1 Inputs to the Tailoring Process

Tailoring is driven by the environment of the system life cycle stages. These environments determine the criteria for process tailoring:

Enterprise environment
As indicated in chapter 6, the enterprise environment provides the context for most processes. Enterprise processes establish standards, policies, processes, goals, and objectives that are based on market environment and opportunities, governmental and other external laws and regulations, and enterprise strategies in response to these factors. The enterprise context includes the industry domain.

Organization environment
The work of an enterprise is conducted through organizations that execute strategy through programs and projects. Factors that influence tailoring at the organizational level include:

- Stakeholders and customers – number of stakeholders, quality of working relationships
- Project budget, schedule, and requirements
- Risk tolerance
- Complexity and precedence of the system

Supplier environment
Today's systems are more often an integration of many systems and system elements to create an operational environment. This demands that cooperation transcends the boundaries of any one organization or enterprise. Harmony between multiple suppliers is often best maintained by agreeing to follow a set of consistent processes and standards. In such environments, consensus on a set of practices is helpful but adds complexity to the tailoring process.

10.2.2 Tailoring Process Activities

Tailoring process activities should be conducted at least once for each stage of the system life cycle.

Identify tailoring criteria for each stage – This activity establishes the criteria for including or excluding any process in the formal conduct of a given stage. Some essential processes, such as configuration management, build cumulatively throughout the system life cycle and may determine a set of permanent activities. Other processes, such as project planning, have a more limited range of applicability.

Determine process relevance to cost, schedule, and risks – This activity analyzes the various environments, including their decision processes, relationships, and sensitivity to risks. The results define the appropriate tailoring of the review, decision and coordination methods for each process activity in each stage.

Determine process relevance to system integrity – This activity analyzes the system features, intended environment, criticality of product/system use, reliability, and availability. It defines the appropriate tailoring of the process activities such as verification, qualification, level of analysis needed, and review and decision gate criteria.

Determine quality of documentation needed – This activity analyzes the support environment, system evolution, criticality of system functions, and internal and external interfaces. It defines the extent of detail needed in documentation for the project.

Determine the extent of review, coordination and decision methods – This activity analyzes the project issues such as stakeholder diversity, extent of their involvement, nature of working relationships, (e.g. single, unified, or conflicting customer needs). These factors influence tailoring of formal reviews, coordination and decision methods, and communications to fit the situation.

10.2.3 Control of the Tailoring Process

Elaboration of the control activities shown in Figure 10-2 are expanded upon in the following paragraphs.

- ***Agreements*** – Agreements between enterprises create constraints on tailoring.
- ***Stakeholder/Customer policy/legal*** – Issues of compliance to stakeholder, customer, and Enterprise policies, objectives, and legal requirements will sometime control the extent of tailoring. Certain documents and procedures may be mandatory in some situations.
- ***Enterprise issues*** – The Enterprise environment controls the processes used in the development, determines who needs to approve certain products, defines what form and content the product takes, and what information can (or cannot) be shared between entities, both internal and external.
- ***Contracting Requirements*** – Methods of procurement or intellectual property will influence the extent of tailoring of the agreement process activities. Tolerance for formal processes is influenced by the contracting method – fixed price, cost plus fixed fee, time and material.
- ***Life cycle process/model used*** – The Life Cycle process/model used determines the extent and nature of System Engineering process application, such as the number of reviews, development iterations, or decision points.

Enterprise Risk Strategy

Each participating enterprise will bring their tolerance for risk to the tailoring process. Risk adverse enterprises may need more detailed information than what the system requires, in order to build confidence in the processes. In such instances, tailoring may introduce extra activities that are removed as the level of trust builds between parties.

10.2.4 Enablers of the Tailoring Process

The following paragraphs elaborate on the enablers shown in Figure 10-2.

- ***Organizational learning*** – A key enabler in the tailoring process is experience with similar systems or familiarity between the participating parties. Beginning with less formal process structure for well-understood systems and established teams may yield significant cost savings without jeopardizing performance or quality.
- ***Organizational maturity*** – Established and well documented processes that are used frequently among parties can contribute to successful outcomes. In such instances, it may be more disruptive and add cost to remove such a process. Consideration of the maturity of the participating parties, both individually and as a whole is an important enabler for tailoring.

10.2.5 Outputs from the Tailoring Process

The following paragraphs elaborate on the output shown in Figure 10-2.

- ***Baseline of tailored processes*** – At the end of the tailoring process a set of formal processes and activities are identified. This plan includes, but is not limited to, a documented set of tailored processes, identification of the system documentation required, the identified reviews, decision methods and criteria, and the analysis approach to be used.
- ***Tailoring sensitivity*** – The tailoring plan, processes, documentation and analyses are sensitive to change and increased knowledge from experience. By identifying the assumptions and criteria for tailoring, the tailoring process can be conducted throughout the life cycle to optimize the use of formal processes.

10.3 Traps in Tailoring

The following discussion reveals traps in the tailoring process.

1) ***Reuse of a tailored baseline from another system without repeating the tailoring process***

It is fallacious to assume that previously tailored baselines are appropriate for all systems. Prior successes are not a guarantee of future success. There is something unique in each system.

2) ***Using all processes and activities, "just to be safe"***

The trap is that the each process carries an overhead cost. If this approach is taken, the quality of the system may actually degrade because of application of an inappropriate process. It can not be called tailoring if there is not a clear justification for the inclusion of every process in the plan.

3) ***Using a pre-established tailored baseline***

Enterprise shortcuts to create templates of baselines that can be taken off the shelf and applied to work based on arbitrary categorizations such as high, medium, and low risk systems can be counter-productive. They carry the same hazards as traps #1 and #2 above. Tailoring is important because the emphasis is placed on the system and only processes that support attainment of the objective in terms of quality and performance should be retained.

4) ***Failure to include relevant stakeholders***

The tailoring process itself can become a unifying activity that establishes shard visions and understanding of the objectives. Suppliers, or other organizations, that are identified and not included in the process may feel disenfranchised with the result that they feel a lower level of commitment to the process baseline. When new parties are added, they should be familiarized with the baseline and asked to make constructive contributions.

1 McConnell, Steven (1998). "The Power of Process," *IEEE Computer,* www.stevemcconnell.com/articles/art09.htm

2 Adapted from a presentation given by Ken Salter, at the Jet Propulsion Laboratory in Pasadena CA. (2003)

Appendix A: System Life Cycle Process N-squared chart

Chart A-1. System Life Cycle Process N-squared chart per ISO/IEC 15288

n-squared chart illustrating input-output dependencies between the System Life Cycle Processes

1	X	X	X		X		X	X	X	X				X	X	X	X					X
	2	X					X							X	X	X	X					X
		3	X	X	X		X							X	X	X	X					X
			4	X			X							X	X	X	X					X
				5	X		X							X	X	X	X					X
					6	X	X	X						X	X	X	X					X
						7	X	X						X	X	X	X					X
							8	X	X	X				X	X	X	X					X
								9	X	X				X	X	X	X					X
									10	X				X	X	X	X					X
										11				X	X	X	X					X
X	X	X		X	X	X	X				12			X	X	X	X	X	X	X	X	X
												13		X	X	X	X	X	X	X	X	X
													14	X	X	X	X	X	X	X	X	X
														15	X	X	X	X	X	X	X	X
														X	16	X	X	X	X	X	X	X
														X	X	17	X					X
														X	X	X	18					X
X											X	X	X	X	X	X	X	19	X			X
X											X	X	X	X	X	X	X	X	20			X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X			21		X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	22	X
											X	X	X	X	X	X	X	X				23

Cross reference between the numbers on the diagonal to the process name

1. stakeholder requirements definition
2. requirements analysis
3. architectural design
4. implementation
5. integration
6. verification
7. transition
8. validation
9. operation
10. maintenance
11. disposal
12. project planning
13. project assessment
14. project control
15. decision-making
16. risk management
17. configuration management
18. information management
19. enterprise management
20. investment management
21. system life cycle processes management
22. resource management
23. quality management

How to read this N-squared chart:

The outputs from lower-numbered processes that are input to higher-numbered processes are indicated by an x in the top diagonal. For an example, the shaded x at the intersection (1,8) reflects the passing of the validation criteria for stakeholder requirements into the Validation Process.

The outputs from higher-numbered processes that are input to lower-numbered processes are indicated by an x in the lower diagonal. For example, the shaded x at the intersection of (21, 3) reflects that the Project processes and procedures identified by the SLC Processes Management Process influences the Architectural Design Process.

Absence of an x in an intersection does not preclude tailoring to create a relationship between any two processes.

Appendix B: Acronym List

AHP	Analytical Hierarchy Process
AIAA	American Institute of Aeronautics and Astronautics [USA]
ANSI	American National Standards Institute
AP	Application Protocol
CCB	Configuration Control Board
CE	Concurrent Engineering
CER	Cost Estimating Relationship
CI	Configuration Item
cm	centimeter
CM	Configuration Management
CMMI®	Capability Maturity Model® Integration
CMP	Configuration Management Plan
ConOps	Concept of Operations
COTS	Commercial Off-The-Shelf
CSCI	Computer Software Configuration Item
CSEP	Certified Systems Engineering Professional
DOD	Department of Defense
DODAF	Department of Defense Architecture Framework [USA]
DoE	Department of Energy
DOE	Design of Experiments
DTC	Design to Cost
ECP	Engineering Change Proposal
EIA	Electronics Industries Alliance
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
EMI/EMC	ElectroMagnetic Interference/EM Compatibility
EN	Engineering Notice
FFBD	Functional Flow Block Diagram
FMECA	Failure Modes, Effects and Criticality Analysis
G&A	General and Administrative (cost)
h	hour
HCDE	Human Centered Design Environment

HSI	Human Systems Integration
HWCI	Hardware Configuration Item
ICD	Interface Control Document
IDEF	Integration DEFinition for Functional Modeling
IEC	International Engineering Consortium
IEEE	Institute of Electrical and Electronics Engineers
IFWG	Interface Working Group
IID	Incremental & Iterative Development
ILS	Integrated Logistics Support
IM	Information Management
IMP	Information Management Plan
INCOSE	International Council On System Engineering
IPAL	INCOSE Process Asset Library
IPDT	Integrated Product Development Team
IPPD	Integrated Product & Process Development
IPT	Integrated Product Team
IS	Interface Specification
ISO	International Organization for Standardization
IT	Information Technology
IV&V	Integration, Verification and Validation
km	kilometer
KTA	Kepner-Tregoe Analysis
LCC	Life Cycle Cost
LSA	Logistic Support Analysis
m	meter
MIT	Massachusetts Institute of Technology
MN	Mega Newton
MODAF	Ministry of Defence Architecture Framework [UK]
MOE	Measure of Effectiveness
MOP	Measures of Performance
MPE	Mass Properties Engineering
NASA	National Aeronautics & Space Administration [USA]
NDI	Non-Development Item
OCD	Operational Concept Document
OMG	Object Management Group

OSI	Open Systems Interconnection (for communication protocols)
PBS	Product Breakdown Structure
PCR	Process Compliance Review
PDT	Product Development Team
PHS&T	Packaging, Handling, Storage & Transportation
PIT	Product Integration Team
PLCS	Product Life Cycle Support
PUID	Project Unique Identifier
QA	Quality Assurance
QFD	Quality Function Deployment
QMP	Quality Management Plan
R&D	Research and Development
R&M	Reliability and Maintainability
RCM	Reliability Centered Maintenance
RFP	Request For Proposal
RMP	Risk Management Plan
ROI	Return on Investment
RTM	Requirements Traceability Matrix
RVTM	Requirements Verification and Traceability Matrix
SE	Systems Engineering; Systems Engineer
SEI	Software Engineering Institute
SEIT	Systems Engineering & Integration Team
SEMP	System Engineering Management Plan, see SEP
SEMS	System Engineering Master Schedule
SEP	Systems Engineering Plan
SI	System Integration
SLC	System Life Cycle
SoW	Statement of Work
SOP	Standard Operating Procedures
SRD	Systems Requirements Document
SRR	System Requirements Review
STEP	STandard for the Exchange of Product model data
SWOT	Strength-Weakness-Opportunity-Threat
SysML	Systems Modeling Language
SYSPG	Systems Engineering Process Group

TBD	To Be Determined
TBR	To Be Resolved
TBS	To Be Supplied
TP	Technical Product
TPM	Technical Performance Measure
TQM	Total Quality Management
UML	Unified Modeling Language
URL	Uniform Resource Locator
USA	United States of America
USB	Universal Serial Bus
V&V	Verification and Validation
VCRM	Verification Cross Reference Matrices
WBS	Work Breakdown Structure

Appendix C: Terms and definitions

The term and definitions *in italic font style* are from ISO/IEC 15288: 2002(E) – *Systems engineering – System life cycle processes.*

Words not included in this glossary carry meanings consistent with dictionary definitions.

acquirer	*the stakeholder that acquires or procures a product or service from a supplier*
Acquisition Logistics	Technical and management activities conducted to ensure supportability implications are considered early and throughout the acquisition process to minimize support costs and to provide the user with the resources to sustain the system in the field.
activity	*a set of actions that consume time and resources and whose performance is necessary to achieve, or contribute to, the realization of one or more outcomes*
Agile	Project execution methods can be described on a continuum from “adaptive” to “predictive.” Agile methods exist on the “adaptive” side of this continuum, which is not the same as saying that agile methods are “unplanned” or “undisciplined.”
agreement	*the mutual acknowledgement of terms and conditions under which a working relationship is conducted*
Baseline	The gate-controlled step-by-step elaboration of business, budget, functional, performance, and physical characteristics, mutually agreed to by buyer and seller, and under formal change control. Baselines can be modified between formal decision gates by mutual consent through the change control process.
Capability	An expression of a system, product, function or process’ ability to achieve a specific objective under stated conditions.
Commercial Off-The-Shelf (COTS)	Commercial items that require no unique acquirer modifications or maintenance over the life cycle of the product to meet the needs of the procuring agency
Configuration	A characteristic of a system element, or project artifact, describing their maturity or perfomance.

Configuration Item (CI)	A hardware, software, or composite item at any level in the system hierarchy designated for configuration management. (The system and each of its elements are individual CIs.) CIs have four common characteristics: 1. Defined functionality, 2. Replaceable as an entity, 3. Unique specification, 4. Formal control of form, fit, and function See Figure I-4 in Appendix I.
Context diagram	This version of the handbook provides a high level view of the process-of-interest. The diagram summarizes the process activities, and their inputs and outputs from/to external actors; some inputs are categorized as controls and enablers. A control governs the accomplishments of the process; an enabler is the means by which the process is performed.
Decision gate	A decision gate is an approval event (often associated with a review meeting). Entry and exit criteria are established for each decision gate; continuation beyond the decision gate is contingent on the agreement of decision-makers.
Derived Requirements	Detailed characteristics of the system-of interest that typically are identified during elicitation of stakeholder requirements, requirements analysis, trade studies or validation
Design Constraints	The boundary conditions, externally or internally imposed, for the system-of-interest within which the organization must remain when executing the processes during the concept and development stage
enabling system	*a system that complements a system-of-interest during its life cycle stages but does not necessarily contribute directly to its function during operation*
enterprise	*that part of an organization with responsibility to acquire and to supply products and/or services according to agreements*
Environment	The surroundings (natural or man-made) in which the system-of-interest is utilized and supported; or in which the system is being developed, produced or retired.
facility	*the physical means or equipment for facilitating the performance of an action, e.g. buildings, instruments, tools*

Failure	The event in which any part of an item does not perform as required by its specification. The failure may occur at a value in excess of the minimum required in the specification, i.e., past design limits or beyond the margin of safety.
Human Factors	The systematic application of relevant information about human abilities, characteristics, behavior, motivation, and performance. It includes principles and applications in the areas of human related engineering, anthropometrics, ergonomics, job performance skills and aids, and human performance evaluation.
Human Systems Integration	The interdisciplinary technical and management processes for integrating human considerations within and across all system elements; an essential enabler to systems engineering practice.
"-ilities"	The operational and support requirements a program must address (e.g., availability, maintainability, vulnerability, reliability, supportability, etc.).
Integration DEFinition for Functional Modeling (IDEF)	A multiple page (view) model of a system that depicts functions and information or product flow. Boxes illustrate functions and arrows illustrate information and product flow. Alphanumeric coding is used to denote the view. IDEF0 - functional modeling method IDEF1 - information modeling method IDEF1X - data modeling method IDEF3 - process description capture method IDEF4 - object oriented design method IDEF5 - ontology description capture method
Life Cycle Cost (LCC)	The total cost to the organization of acquisition and ownership of a system over its entire life. It includes all costs associated with the system and its use in the concept, development, production, utilization, support and retirement stages.
life cycle model	*a framework of processes and activities concerned with the life cycle, which also acts as a common reference for communication and understanding*
Measure of Effectiveness	A metric used to quantify the performance of a system, product or process in terms that describe a measure to what degree the real objective is achieved.

N-squared diagrams	This graphical representation can be used to define the internal operational relationships or external interfaces of the system-of-interest.
operator	*an individual who, or an organization that, contributes to the functionality of a system and draws on knowledge, skills and procedures to contribute the function*
organization	*a group of people and facilities with an arrangement of responsibilities, authorities and relationships [ISO 9000:2000]*
Performance	A quantitative measure characterizing a physical or functional attribute relating to the execution of a process, function, activity or task; Performance attributes include quantity (how many or how much), quality (how well), timeliness (how responsive, how frequent), and readiness (when, under which circumstances).
process	*set of interrelated or interacting activities which transforms inputs into outputs [ISO 9000:2000]*
project	*an endeavor with defined start and finish dates undertaken to create a product or service in accordance with specified resources and requirements*
Proof-of-concept	A naïve realization of an idea or technology to demonstrate its feasibility
Requirement	A statement that identifies a system, product or process' characteristic or constraint, which is unambiguous, clear, unique, consistent, stand-alone (not grouped), and verifiable, and is deemed necessary for stakeholder acceptability.
resource	*an asset that is utilized or consumed during the execution of a process*
Specialty engineering	Analysis of specific features of a system that requires special skills to identify requirements and assess their impact on the system life cycle
stage	*a period within the life cycle of a system that relates to the state of the system description or the system itself*
stakeholder	*a party having a right, share or claim in a system or in its possession of characteristics that meet that party's needs and expectations*
supplier	*an organization or an individual that enters into an agreement with the acquirer for the supply of a product or service*

system	*a combination of interacting elements organized to achieve one or more stated purposes*
system element	*a member of a set of elements that constitutes a system*
system life cycle	*the evolution with time of a system-of-interest from conception through to retirement*
system-of-interest	*the system whose life cycle is under consideration*
System of systems	System of systems applies to a system-of-interest whose system elements are themselves systems; typically these entail large scale inter-disciplinary problems with multiple, heterogeneous, distributed systems.
Systems Engineering	Systems Engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem. Systems Engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs. [INCOSE]
Systems Engineering Effort	Systems Engineering effort integrates multiple disciplines and specialty groups into a set of activities that proceed from concept to production to operation. Systems Engineering considers both the business and the technical needs of all stakeholders with the goal of providing a quality system that meets their needs.
Systems Engineering Plan	Structured information describing how the system engineering effort, in form of tailored processes and activities, for one or more life cycle stages, will be managed and conducted in the organization for the actual project.
Tailoring	The manner in which any selected issue is addressed in a particular project. The organization may seek to minimize the time and efforts it takes to satisfy an identified need consistent with common sense, sound business management practice, applicable laws and regulations, and the time sensitive nature of the requirement itself. Tailoring may be applied to various aspects of the project, including project documentation, processes and activities performed in each life cycle stage, the time and scope of reviews, analysis, and decision-making consistent with all applicable statutory requirements.

trade-off	*decision-making actions that select from various requirements and alternative solutions on the basis of net benefit to the stakeholders*
user	*individual who or group that benefits from a system during its utilization*
validation	*confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled [ISO 9000: 2000]*
verification	*confirmation, through the provision of objective evidence, that specified requirements have been fulfilled [ISO 9000: 2000]*

Appendix D: The Context for Appendices E to N

Version 3.0 of the Systems Engineering Handbook was released in June 2006. The document was a significant departure from its predecessors in that it is based on ISO/IEC 15288:2002 (E) *Systems engineering – system life cycle processes.* The handbook was constrained to 150 pages (excluding the front material and appendices). The intent in 2006 was that elaborating material for each major handbook subsection would be put into appendices and stored in electronic form in the INCOSE Process Asset Library (IPAL).

These appendices are the first step in providing elaboration of systems engineering practices. The intent is to expand on the practices defined in Version 3.0 of the Systems Engineering Handbook providing in-depth information, "how-to" guidance, and examples for systems engineering practitioners. It is envisioned that over time relevant INCOSE working groups will enhance the existing material and expand these appendices with additional SE practices. It is also envisioned that industry specific practices in areas such as aerospace, transportation, medical, automotive and other industries, as well as the US Department of Defense and the UK Ministry of Defence will provide supplemental material to this handbook, resulting in a comprehensive body of knowledge in systems engineering. Over time, the common practices identified will become world class practices from which all industries will benefit. The following are the appendices E to N included in Version 3.1 of the Systems Engineering Handbook:

Appendix E: The Hierarchy WITHIN A System

Appendix F: Acquisition and Supply – Defining Needs

Appendix G: Systems Engineering Technical Management

Appendix H: Integrated Product and Process Development

Appendix I: Requirements Definition Process

Appendix J: Functional Analysis and Allocation

Appendix K: System Architecture Synthesis

Appendix L: Systems Engineering Analyses Activities

Appendix M: Human Systems Integration

Appendix N: System Integration

The intended use for this INCOSE body of knowledge it to promote the use of systems engineering and provide guidance to all systems engineers. It is also the source of information for the certification examinations. Effective October 2007 SEHv3.1 fully supports the CSEP exam. The earlier handbook, SEHv2a, can be used as well until the end of June 2008. After that time exam questions will be added based on material in SEHv3.1 only.

Appendix E: The Hierarchy *Within* A System

One of the Systems Engineer's first jobs on a project is to establish nomenclature and terminology that support clear, unambiguous communication and definition of the system, its functions, components, operations, and associated processes. While there is no universally accepted nomenclature, this appendix presents one convention in widespread use.

It is essential to the advancement of the field of Systems Engineering that common definitions and understandings be established regarding general methods and terminology. As more Systems Engineers accept and use a common terminology, we will experience improvements in communications, understanding, and ultimately, productivity. Toward that end, the following definitions of the hierarchy within a system are useful for the discussions in the appendices of this handbook. The following definitions elaborate the hierarchy terms used in Section 1.5: Definitions….

System An integrated set of elements, subsystems, or assemblies that accomplish a defined objective. These elements include products (hardware, software, firmware), processes, people, information, techniques, facilities, services, and other support elements. An example would be an air transportation system.

Element A major product, service, or facility of the system, e.g. the aircraft element of an air transportation system (commonly used, but subsystems can be used instead of elements).

Subsystem An integrated set of assemblies, components, and parts which performs a cleanly and clearly separated function, involving similar technical skills, or a separate supplier. Examples are an aircraft on-board communications subsystem or an airport control tower as a subsystem of the air transportation system.

Assembly An integrated set of components and/or subassemblies that comprise a defined portion of a subsystem, e.g. the pilot's radar display console or the fuel injection assembly of the aircraft propulsion subsystem.

Subassembly An integrated set of components and/or parts that comprise a well-defined portion of an assembly, e.g. a video display with its related integrated circuitry or a pilot's radio headset.

Component Comprised of multiple parts; a cleanly identified item, e.g. a cathode ray tube or the ear-piece of the pilot's radio headset.

Part The lowest level of separately identifiable items, e.g. a bolt to hold a console in place.

An example of a common hierarchy is shown in Figure E-1.

The system hierarchy should be a balanced hierarchy with appropriate fan-out and span of control. Appropriate fan-out and span of control refers to the number of elements subordinate to each element in the hierarchy. Hierarchies are organizational representations of a partitioning relationship. The hierarchy represents a partitioning of the entity into smaller more manageable entities.

System hierarchies are analogous to organizational hierarchies. Both can suffer from improper balance; that is, too great a span of control or excessive layers in the hierarchy. A "rule of thumb" useful in evaluating this balance is that a system should have no more than 7 ± 2 elements reporting to it. In the same way an element should have no more than 7 ± 2 subsystems reporting to it, and so on. A design level with too many entities reporting suffers from too much complexity. The design and corresponding test activities run the risk of running out-of-control or acquiring an informal partitioning that guides the work without proper control or visibility. A level of design with too few entities likely does not have distinct design activity, and both design and testing activities contain redundancy.

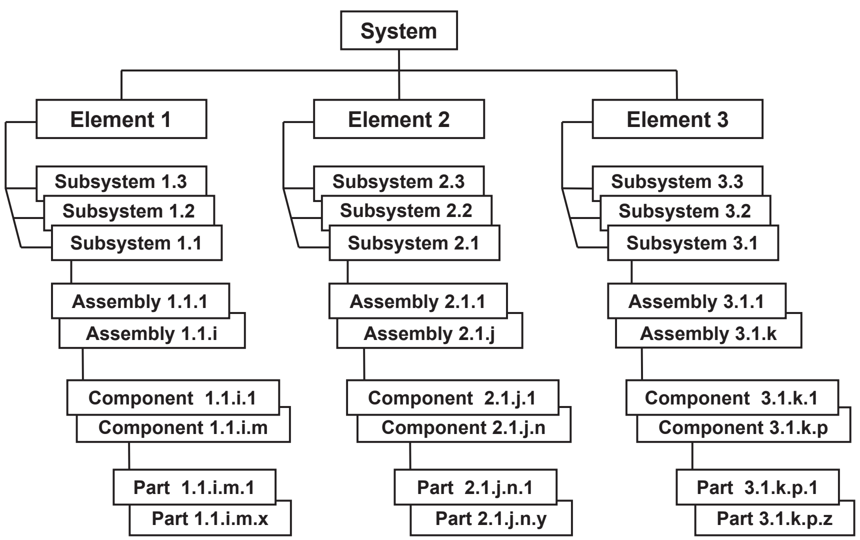


Figure E-1. Hierarchy *Within* a System

The depth of the hierarchy can be adjusted to fit the complexity of the system. For example, in the complex Apollo program, NASA added a "Module Level" in the hierarchy to breakout the Command Module, Lunar Module, etc. of the Space Vehicle

Element. Simple systems may have fewer levels in the hierarchy than complex systems. Some examples of the hierarchy of system terminology are shown in Figure E-2.

System	**Air Logostics**	**Information**	**Electric Car**
Elements	Aircraft Package Processing Support Equipment Air & Ground Crews Hub, Base, Facility	Computers Network Printers Data Storage Personnel	
Sub-systems		Data Processor Operating System Software	Power Train Body Chassis
Components		Input Devices Output Devices Processing Memory	Battery Motor(s) Generator Controller

Figure E-2. Examples of System Hierarchy

THIS PAGE
INTENTIONALLY
LEFT BLANK

Appendix F: Acquisition and Supply – Defining Needs

As noted in Section 6.7: Acquisition Process, Section 6.8: Supply Process, and Section 8.1: Acquisition and Supply, the initiation of a project begins with user need. Once a need is perceived and resources are committed to establish a project it is possible to define the parameters of an acquisition and supply relationship. Often our experience of the acquisition process is typified by the purchase of telephones or automobiles. To facilitate the purchase of more complex services and products is a primary Systems Engineering responsibility. The start of an acquisition/supply process begins with the determination of and agreement on user needs.

Users and Other Participants

Determining user need is normally designated the "Concept" or "Development" phase of the development life cycle, since the "big picture" is developed and authorization to fund the development and select the developers/suppliers are its primary functions. When Systems Engineering is employed as the implementation process, the key to this activity is to establish a decision database containing objectives, top-level quantified mission requirements, potential design and operational concepts, and a substantiation (verification) that the database is a valid interpretation of user needs. All parties involved in this process (users, developing agencies, builders, support organizations, etc.) should maintain and contribute to this database. Under some circumstances, however, if the product is a refrigerator, a telephone, an automobile or other consumer product, it may not be practical to elicit needs from the "user" but rather from the marketing organization or other surrogate.

In many cases the integration of user views, which may not necessarily be harmonious, is accomplished by "committee action". However, this can lead to confusion and unsatisfactory decision making. As the Systems Engineering process is applied, a common paradigm for examining and prioritizing available information and determining the value of added information can be created. Each of the user's views of the needed systems can be translated to a common description that is understood by all participants, and all decision making activities recorded for future examination. The top-level program and system descriptions can be established.

Recommended Activities

Systems Engineering should support acquisition program management in defining what must be done and gathering the information, personnel, and analysis tools to define the mission or program objectives. This includes gathering customer inputs on "needs" and "wants", system/project constraints (costs, technology limitations, and applicable specifications/legal requirements), and system/project "drivers" (such as capabilities of the competition, military threats, and critical environments). The set of recommended activities that follow are written for a complex project that meets a stated mission or goal, but the word "product" can be substituted to apply these steps to commercial products, for example.

1. Identify users and other stakeholders and understand their needs. Develop and document the new mission needs of all user organizations through user surveys.

2. Perform mission analysis to establish the operational environment, requirements, functionality, architecture, and verify capability (i.e. assess existing capability).

3. Document the inadequacies or cost of existing systems to perform new mission needs.

4. If mission success is technology driven, develop concepts and document the new capabilities that are made possible by the introduction of new or upgraded technology. Document the tradeoffs in mission performance vs. technology steps.

5. Prepare a justification for the need for this mission compared to alternative missions competing for the same resources.

6. Prepare the necessary documents to request funding for the first program phase.

7. If system procurement is involved, develop the information needed to release a request for proposal, establish the selection criteria and perform a source selection.

The inputs for acquisition & supply analysis depend on the market and the sellers. For products, where the first to market gains a larger market share and developers bear the burden for development costs, incremental or evolutionary development techniques are more common and the solicitation of user needs is conducted with stepwise refinements. For complex systems, the analysis of the need is ongoing.

The output of mission level activities should be sufficient definition of the operational need or concept of operations to gain authorization and funding for program initiation. The output should also generate a request for proposal if the system is to be acquired through a contract acquisition process, or to gain authorization to develop and market the system if market driven. These outputs can be documented in a mission statement, a system requirements document, a statement of work, and a request for proposal.

End Results

Contributing users have well defined completion criteria:

- User organizations have gained authorization for new system acquisition.
- Program development organizations have prepared a Statement of Work (SOW), Systems Requirements Document (SRD), and gained approval for new system acquisition. If they are going to use support from outside the company, they have issued a Request for Proposal (RFP), and selected a contractor.
- Potential contractors have influenced the acquisition needs, submitted a proposal, and have been selected to develop and deliver the system.
- If the system is market driven, the marketing group has learned what consumers want to buy. For expensive items (aircraft) they have obtained orders for the new systems.
- If the system is market and technology driven, the development team has obtained approval to develop the new system from the corporation

The metrics used to evaluate the acquisition & supply process are usually based on completion criteria, such as:

1. System Requirements Document, percent completion

2. Requirements stability and growth metrics, such as, number of requirements added, modified, or deleted during the preceding time interval (month, quarter, etc.).

3. Percent completion of each contract requirements document: SOW, RFP, Contract Data/Document Requirements List (CDRL), etc.

Methods / Techniques

There are many techniques that can be used for eliciting user requirements such as, marketing and technical questionnaires or surveys, focus groups, prototypes, and beta release of a product. Several aspects of requirements management are discussed further in Section 7.2: Requirements Management.

Trade-off analysis and simulation tools can be used to evaluate mission operational alternatives and select the desired mission alternative.

Organizations with mature Systems Engineering practices will identify and manage risks associated with an acquisition program and establish the controls and actions to reduce such risks to an acceptable level as discussed in Sections 5.6: Risk and Opportunity Management Process and Section 7.3: Risk and Opportunity Management.

THIS PAGE
INTENTIONALLY
LEFT BLANK

Appendix G: Systems Engineering Technical Management

G.1 Systems Engineering Plan (SEP)

The Systems Engineering Plan (also called Systems Engineering Management Plan or SEMP) is the top-level plan for managing the Systems Engineering effort. The SEP defines how the project will be organized, structured, and conducted and how the total engineering process will be controlled to provide a product that satisfies customer requirements. A SEP should be prepared early in the project, submitted to the customer (or to management for in-house projects), and used in technical management for the study and development periods of the project, or the equivalent in commercial practice. The format of the SEP can be tailored to fit project, customer, or company standards.

Usually SE organizations maintain a focus on Systems Engineering processes. To maximize reuse of the SEP for multiple projects, project-specific appendices are often used to capture detailed and dynamic information such as the decision database, and deliverables schedule.

Participants in the creation of the SEP should include senior Systems Engineers, representative subject matter experts, the project management, and often the customer.

Recommended activities

Creation of the SEP involves defining the SE processes, functional analysis approaches, schedule and organizational roles and responsibilities, to name a few of the more important parts of the plan. As described in Section 8.6, the Systems Engineering Master Schedule (SEMS) is an essential part of the SEP, because it identifies the critical path of technical activities in the project.

Appendix H of this handbook discusses integrated project team organizations in some detail. The SEP should report the results of the effort undertaken to form a project team.

A technical objectives document should be developed. This may be one of the source documents for the decision database described below. The document may be part of a Concept of Operations for the system (see Section 7.2.3).

The SEP must identify the source material to be used in developing deliverables such as the Systems Specification and Technical Requirements Document. The SEP defines the scope of the project by identifying the applicable source documents (customer specifications, statement of work, industry standards, ect.) that must be observed in the performance of the project.

The approach and methods used to define the performance and functional requirements for the following areas of SE and Specialty Engineering should also be documented:

a) Definition of the organization of the project, and how SE interfaces with the other parts of the organization. How are communications at these interfaces handled? How are questions/problems elevated up the organization and resolved? Define responsibilities and authority of the key positions.
b) Crisp definition of the system boundaries and scope of the project.
c) Statement of project assumptions and constraints.
d) Key technical objectives.
e) Validation planning (not just verification planning).
f) Configuration Management planning.
g) Quality Assurance planning.
h) Resource (facilities, tools, IT, personnel, etc.)
i) Reliability and availability
j) Maintainability, supportability, and Integrated Logistics Support (ILS)
k) Survivability including nuclear, biological, and chemical
l) Electromagnetic compatibility, radio frequency management, and electrostatic discharge
m) Human Engineering and Human Systems Integration
n) Safety, health hazards, and environmental impact
o) System security
p) Producibility
q) Test and evaluation
r) Testability and integrated diagnostics
s) Computer resources
t) Transportability
u) Infrastructure support
v) Other engineering specialties bearing on the determination of performance and functional requirements

The SEP should indicate what trade studies will be included in the project.

Technical reviews are essential to insure that the system being developed will meet requirements, and that the requirements are understood by the development team. Formal reviews are essential determine readiness to proceed to the next stage of the system life-cycle. The SEP should list what technical reviews will be conducted and the methodology to be used in solving problems uncovered in reviews.

The life cycles in Figure 3-2 (Section 3.3) illustrate the appropriate time for decision gates. They may or may not be right for your project. You may need more or fewer reviews. Remember that formal, documented decision gates, with the customer in attendance can have a significant cost, so also use more-frequent informal, in-house reviews to resolve most issues; and strive to exit the decision gates with no major customer-imposed Action Items, i.e. be prepared.

Transitioning critical technologies should be done as a part of the risk management. A discussion of risk management is contained in section 7 of this handbook. It is called out separately here for special emphasis. Identify what technologies are critical and follow the steps outlined for risk management. Reference the work done (or to be done) explicitly in the SEP.

The system being proposed may be complex enough that the customer will require training in order to use it. During the project it may be necessary to train those who will develop, manufacture, verify, deploy, operate, support, do training, or dispose of the system. A plan for this is required in the SEP. Include in the training section:

a. Analysis of performance
b. Behavior deficiencies or shortfalls
c. Required training to remedy the above
d. Schedules to achieve required proficiencies

Verification planning is usually done following a verification matrix which lists all the requirements. The possible methods of verification include inspection, analysis, demonstration, and test. The SEP should state that a verification plan will be written to define the items to be verified and which methods will be used to verify performance. Detailed procedures are usually not written for inspection, analysis, and demonstration methods. Simulations may be used for testing when quantifiable results are needed or demonstration when qualititaive results are satisfactory.

The plan should define, at least in preliminary general terms, which performance items will be verified by which of the above methods. The plan should also define who is to perform and witness the verification of each item. This should also relate to the SEMS for time phasing of the verification process.

End Results

A well-written SEP provides guidance to a project and helps the organization avoid unnecessary discussions about how to perform Systems Engineering. In addition, a schedule and organization are defined that help the project procure the personnel needed throughout the development lifecycle and assess progress. The SEP outlines the major deliverables of the project including a decision database, specifications and baselines.

The Process Inputs paragraph of the SEP identifies the source material to be used for these deliverables, such as the Statement of Work, the technical requirements document, and the specification from the request for proposal. It also may include previously developed specifications for similar systems and company procedures affecting performance specifications.

The Project (or Contract) Work Breakdown Structure (WBS) document defines and outlines the project/program task hierarchy. Work authorization is the process by which the project is baselined and financially controlled. A description of your company's procedures for starting work on the detailed parts of the WBS should be defined in the SEP.

The SEP should also address Design to Cost and Value Engineering which would provide insight into system/cost effectiveness. For example, "Can the project be engineered to have significantly more value with minimal additional cost?" If so, does the customer have the resources for even the modest cost increase for the improvement? The intent is to assure the customer that no obvious cost effective alternatives have been overlooked.

Technical Performance Measure (TPM) is a tool for project control and the extent to which TPM will be employed should be defined in the SEP. The use of TPM is described in Section 7.2.5.

G.2 Systems Engineering Processes and Practices

G.2.1 Standard SE Processes

Stakeholders/Participants

An organization engaged in Systems Engineering should identify the standard process and the project tailored process. It must provide the requirements for establishing, maintaining, and improving the standard SE process. It must define a process for tailoring the standard SE process for use on projects, and define improvements to the tailored project SE processes. It is applicable to every engineering capability maturity focus area or process area in the Capability Maturity Model® Integration (CMMI®).

The organization should establish standard policies; SE processes, SE practices, and supporting functional processes (see Figure G-1). Organizational management must review and approve the standard SE process and changes to it. Organizations should establish a Systems Engineering Process Group (SYSPG) to oversee SE process definition and implementation.

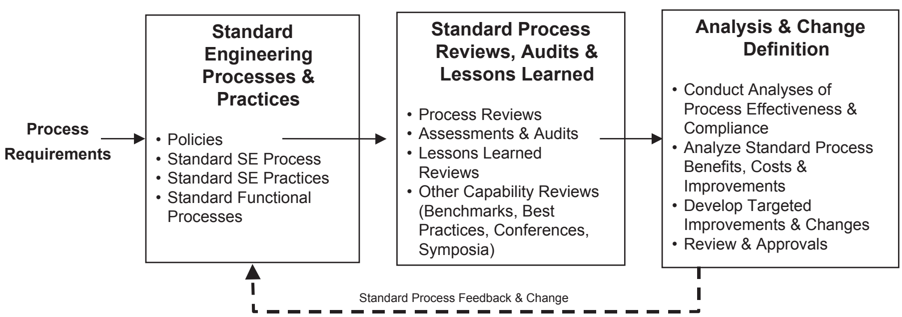


Figure G-1. Standard SE Process Flow

Recommended Activities

An organization establishes a standard SE process using a reference SE process model, which is tailored by projects to meet specific customer and stakeholder needs. The reference model should tailor industry, government or other agency "best practices" based on multiple government, industry and organization reference SE process documents. The reference SE model must include a SE improvement process. Projects are expected to follow this process, as tailored to meet project-specific SE process needs. The standard process must be tailorable, extensible, and scalable to meet a diverse range of projects, from small study contracts to large projects requiring thousands of participants.

The standard SE process model is established by selection of specific processes and practices from this handbook, industry SE process references (such as ANSI/EIA-632 and ISO 15288), and government SE process references as appropriate.

A high performing organization also must conduct reviews of the process (as well as work products), conduct assessments and audits (such as CMMI® assessments and ISO audits of SE), retain its corporate memory through the understanding of lessons learned, and establish how benchmarked processes and practices of related organizations can affect the organization. Successful organizations should analyze their process performance, its effectiveness, compliance to organizational and higher directed standards, benefits and costs, and develop targeted improvements.

End Results

The basic requirements for standard and tailored project SE process control, based on CMMI®, are:

a. SE processes shall be identified for use on projects.
b. Implementation and maintenance of SE processes shall be documented.
c. Inputs and outputs shall be defined for SE subprocesses.
d. Entrance and exit criteria shall be defined for SE process major activity.
e. Projects shall use a defined set of standard methods or techniques in SE process.
f. Tailoring guidelines shall be used to permit the standard process to meet project-specific needs.
g. Project management shall identify what parts of the standard SE process have been tailored to meet project-specific needs.
h. Strengths and weaknesses in the SE process shall be assessed.
i. The SE process shall be periodically assessed.
j. The SE process shall be compared to benchmark processes used by other organizations.

In addition, basic requirements specifically for SE process improvement control from these standards are:

a. Organization best practices shall be identified and communicated to projects.
b. The standard SE process shall identify areas for future improvement.
c. SE process users shall be able to identify proposed improvements.
d. Compliance with improvement processes, plans and practices shall be verified.
e. The project tailored SE improvement process shall include a means for evaluating its effectiveness.
f. The project tailored SE improvement process shall include a means for making needed improvements.
g. The standard SE process work products shall be reviewed and results used to improve this process.
h. The standard SE process compliance shall be reviewed and results used to improve this process.

G.2.2 Reviews, Audits, Lessons Learned, Analysis for Change

The standard SE process must meet requirements for review, assessment, and audit; and for establishment of lessons learned and best practices.

Process Compliance Reviews (PCRs)

The standard SE process must include a periodic Process Compliance Review (PCR) for assessing key process element implementation effectiveness. PCRs must be conducted on a recurring basis determined by the SE organization with management involvement. If the organization conducts other assessments or audits on a recurring basis (such as for self assessment or ISO 9000), they can be combined into one assessment review to reduce the perceived burden. A standard SE process checklist should be used as the basis for this PCR. It may be augmented by additional issues and questions. Each review may address a subset of the standard SE checklist. The questions asked and results from this review should be recorded and stored. The review should address defects in the SE process in use at the time of the review. The review should address the improvement process, tailoring of the SE process, and tailoring of the improvement process (if applicable).

The PCR must be organized by a PCR Coordinator who will notify responsible personnel of the specific dates, formats and requirements for the reviews, define the lists of required attendees and invitees, and set the agenda. The data presented in these reviews should be archived. Key results from PCRs must be provided for management consideration.

The PCR should cover at least the following:

- Identify strengths and weaknesses in the SE process and its improvement process.
- Identify key process elements which need to be followed in large and/or small projects

- Identify areas for future improvement
- Address the effectiveness of the tailored improvement process
- Address the conduct of, defects in, and improvements to the SE improvement process
- Review SE work products to identify potential trends indicating possible systemic issues
- Review the results of PCRs to identify potential trends indicating possible systemic issues
- Review a sampling of in-process reviews to identify potential trends indicating possible systemic issues
- Review the definition and use of metrics in SE process measurement.

Assessments and Audits

Assessments and audits should be conducted, which include internal and external assessments of capability maturity, and internal and external audits of key SE processes and those personnel which implement them.

Internal assessments of capability maturity should be conducted to improve the organization's SE process, and to prepare for external assessments. The assessment team should consist of at least one external, qualified lead assessor. The standard for use in capability assessments will be an external, industry formal standard such as CMMI® or equivalent.

External assessments of capability maturity should be conducted, when justified by the business needs of the organization. They should be led by an external, qualified lead assessor, with a significant part of the assessment team consisting of external, qualified assessors. The standard for use in capability assessments should be the external, industry formal standard required by organization or customer, such as the CMMI® or equivalent.

Internal audits of organizations using SE processes should be conducted to prepare for an external audit of the organization. A standard SE process activity checklist should be used as the basis for this audit. It may be augmented by additional issues and questions. Each audit may address a subset of the standard checklist. The questions asked and results from this audit must be recorded and stored. The audit should investigate defects (i.e. process errors) in the SE process in use at the time of the audit to understand why the defects or errors occurred.

Lessons Learned

Lessons learned are needed to recognize an organization's good and best practices, understand the lessons of the organization's history and avoid repeating the mistakes of the past. They must address technical SE management and engineering, specialty management and engineering, and any other project or organization activities affecting SE processes.

The SE organization must plan, and follow through, to collect lessons learned at pre-defined milestones in the system life cycle. The SE organization should periodically review lessons learned to analyze and improve SE processes and practices. It should establish best practices and capture them in an easy-to-retrieve form.

G 2.3 Analysis and Change Definition

The standard SE process must meet requirements for analysis and change of the standard SE process.

Analysis of Processes

The System Engineering Process Group (SYSPG) should sample and monitor project implementation of tailored SE processes to identify potential systemic problems in the standard SE process. Feedback, minutes, and reports from project assessments, audits, formal reviews, in-process reviews, and Process Compliance Reviews should be sampled and analyzed. Results of training evaluations and action item compliance reports should be analyzed. Reports of lessons learned and best practices should be analyzed. These analyses should identify and define potential process strengths, weaknesses, deficiencies, and problem areas.

The SYSPG should assess activities providing insight into project SE process effectiveness and compliance. The assessments should address project SE process implementation to understand project issues and concerns with the SE process, and to identify potential systemic problems in the standard SE process. Assessments should identify potential strengths, weaknesses, deficiencies or problem areas in the standard SE process that are revealed in the project process assessments. These assessments will not evaluate or judge project performance; they will focus on internal standard SE process improvement.

Assessments should address at least the following issues:

1. Is the SE process effective and useful (e.g. are we getting what we need from it)?
2. Can the SE process be improved (e.g. (1) are there process elements which were a “waste of time” or (2) are there process elements which should have been done or could have been done better)?
3. What can we change in the SE process to make it better (e.g. what could we do to eliminate the recorded action items or defects)?
4. What is the productivity of the standard major SE process elements?
5. Are the SE support tools and facilities effective and useful?
6. Is information being collected on the effectiveness and usefulness of the SE process?

7. Is information being used to improve the effectiveness and usefulness of the SE process?

These analyses and assessments should establish for the standard SE process, its:

- Effectiveness
- Utility
- Utility of support tools and facilities
- Issues and concerns
- Compliance in the organization
- Understanding of implementation impacts
- Potential systemic problems
- Potential for improvement

Rationales for and results from decisions should be recorded and stored.

G.3 Continuous Process Improvement

Defining Improvements Needed

The analyses and assessments should determine whether changes are needed in the standard SE process and its documentation. Improvements needed in tailoring guidance to better meet project-specific needs should be identified. The System Engineering Process Group (SYSPG) should document and store process compliance in process compliance and/or exception reports.

The organization should improve the SE process based in large part on usage, experience, and feedback from the programs as noted above. The standard SE process improvement should be managed and improved with the participation and support of key stakeholders in the process activities.

The SYSPG must identify strengths of and areas for improvement in the standard SE process. Assessments must consider trends in process technology such as changes in capability maturity assessment practices wanted by organizational stakeholders. Areas of improvement should be developed from the results of project work product reviews, management reviews, and SE process compliance reviews. Areas of improvement should be prioritized and alternatives assessed. The requested or recommended areas for improvement and the impact of consequential changes should be identified. If the improvements or changes involve errors or problems with the existing process, these are identified to determine the actions needed to prevent future occurrences.

The SYSPG should identify and refine the criteria being used in analyses and assessments to improve their focus on essential organization business and process needs. Criteria should be recorded and stored.

SE Process Changes

The SYSPG should prioritize the requested or recommended areas for improvement for the standard SE process. Management should approve the prioritized areas for improvement. Management should decide on what changes will be made, and adjusts budgets and labor estimates as needed to enable the changes to be accomplished. Changes may be required, requested, or recommended based on prioritized areas for improvement, process compliance requirements and/or exception reports. The SYSPG should study the priority areas for improvement, identify the specific changes needed, and recommend adjustments. The SYSPG should determine which changes can be made in the standard SE process to implement the priority improvements within budget and schedule.

The SYSPG should also assess the areas for improvement and related analyses to determine if additional tailoring guidelines are needed. If so, they should identify the tailoring changes needed, fit them into the overall improvement priority scheme, and recommend which changes should be made. A SE Process Improvement Plan should be developed and updated at least annually based in part on targeted improvements and results from reviews. After the SYSPG have prepared standard SE process process changes, they will be submitted to SE and project management for approval, with the coordination of the project managers.

G.4 Configuration Management

Configuration Management (CM) establishes and maintains control over requirements, specifications, configuration definition documentation, and design changes. Configuration Identification, Change Control, Status Accounting and the audit of the functional and physical configuration are the primary activities of configuration management.

Implementation

There will always be a need to make changes; however, Systems Engineering must ensure that the change is (1) necessary, and (2) that the most cost-effective solution has been proposed. Configuration Management must:

- Identify and document the functional and physical characteristics of individual configuration items making them unique and accessible in some form;
- Establish controls to changes in those characteristics; concur in product release, and ensure consistent products via the creation of baseline products;
- Record, track, and report change processing and implementation status and collect metrics pertaining to change requests or problems with the product baseline.

The initial planning efforts for CM should be defined in a Configuration Management Plan. The configuration management program is implemented at the onset of the project. The CM Plan identifies the resources required, defines the tasks to be performed, describes organizational roles and responsibilities, identifies CM tools and processes, as well as methodology, standards and procedures to be used.

The CM process consists of four subprocesses: configuration identification, control, status accounting, and audits (validation and distribution). The CM program is implemented at the onset of the project.

Configuration Identification

The configuration identification process uniquely identifies the elements within a baseline configuration. This unique identification promotes the ability to create and maintain master inventory lists of baselines. As part of the Systems Engineering effort the system will be decomposed into Configuration Items, which will serve as the critical elements subjected to rigorous formal control. The compilation of all the configuration items is called the Configuration Item list. This list may reflect items that are developed, vendor produced, or provided by the customer for integration into the final system. These items may be deliverable items under the contract or used to produce the deliverable items.

Configuration Control

Managing the collection of the items to be baselined is another aspect of configuration management. Configuration control maintains the integrity of the configuration items identified by facilitating approved changes (e.g. via engineering change requests) and preventing the incorporation of unapproved changes into the baseline. Change control should be in effect beginning at project initiation.

Change Classification

Effective configuration control requires that the extent of analysis and approval action for a proposed engineering change be in concert with the nature of the change. The problem statement includes a description of the proposed change, the reason for the proposed change, the impacts on cost and schedule, and identifies all affected documentation. Change classification is a primary basis of configuration control. All changes to baselined documents are classified as outside of the scope of the requirements or within the scope of the requirements. A change outside the scope of project requirements is a change to a project baseline document that affects the form, fit, specification, function, reliability, or safety. The coordinating review board determines if this proposed change requires a change notice for review and approval.

Changes are sometimes categorized into two main classes: Class I and Class II. A Class I change is a major or significant change that affects cost, schedule, or technical performance. Normally Class I changes require customer approval prior to being implemented. A Class II change is a minor change that often affects documentation errors or internal design details. Generally, Class II changes do not require customer approval.

Configuration Control Board

An overall configuration control review board is implemented at the time of project initiation and is established to provide a central point to coordinate, review, evaluate, and approve all proposed changes to baselined documentation and proposed changes to baselined configurations including hardware, software, and firmware. The review board is composed of members from the functional disciplines including Systems Engineering, software and hardware engineering, project management, product assurance, and configuration management. The chairperson is delegated the necessary authority to act on behalf of the project manager in all matters falling within the review board responsibilities. The CM organization is delegated responsibility for maintaining status of all proposed changes. *Satellite* or *subordinate* boards may be established for reviewing software or hardware proposed changes below the configuration item (CI) level. If those changes require a higher approval review they are forwarded to the overall review board for adjudication.

Changes that fall within the review board jurisdiction should be evaluated for technical necessity, compliance with project requirements, compatibility with associated documents, and project impact.

As changes are written while the hardware and/or software products are in various stages of manufacture or test, the review board should require specific instructions for identifying the effectivity or impact of the proposed software or hardware change and disposition of the in-process or completed hardware and/or software product. The types of impacts the review board should assess typically include that:

- All parts, materials, and processes are specifically approved for use on the project;
- The design depicted can be fabricated using the methods indicated;
- Project quality and reliability assurance requirements are met; and
- The design is consistent with interfacing designs

Problem Reports

Problem Reports are written to identify the occurrence of a problem. The problem should be documented in either electronic or hardcopy. The problem report will identify time, date, location of the occurrence, and is reviewed by the review board. The problem statement should provide accurate and clear information of the problem. The review board validates the problem statement, assigns a responsible engineer to implement the change. When implementation of the change has been made, feedback of the resolution is provided to CM and the review board members.

Methods/Techniques

Change control forms provide a standard method of reporting problems and enhancements that lead to changes in formal baselines and internally controlled items. The following forms provide an organized approach to changing hardware, software or documentation:

- Software Problem/Change Reports can be used for documenting problems and recommending enhancements to software or its complementary documentation. These forms can be used to identify problems during software design, code, integration, and test.
- Specification Change Notice is used to propose, transmit, and record changes to baselined specifications.
- Engineering Change Proposals are used to propose Class I changes to the customer. These proposals describe the advantages of the proposed change, available alternatives, and identify funding needed to proceed.
- Engineering Change Requests are used to propose Class II changes.
- Request for Deviation/Waiver is used to request and document temporary deviations from configuration identification requirements when permanent changes to provide conformity to an established baseline are not acceptable.

Configuration Status Accounting

Status accounting is performed by CM to record and report information to management. CM maintains a status of approved documentation that identifies and defines the functional and physical characteristics, status of proposed changes, and status of approved changes. This subprocess synthesizes the output of the identification & control subprocesses. All changes authorized by the configuration review boards (overall and subordinate) culminate in a comprehensive traceability of all transactions.

Such activities as check-in and check-out of source code, builds of configuration items, deviations of manufactured items, waiver status are part of the status tracking.

By statusing and tracking project changes, a gradual change from the *build-to* to the *as-built* configuration is captured.

Metrics

Suggested metrics for consideration are: number of changes processed, adopted, rejected, and open; status of open change requests; classification of change requests summary; number of deviations or waivers by Configuration Item; number of problem reports open, closed, and in-process; complexity of problem reports and root cause; labor associated with problem resolution, and test phase when problem was identified; processing times and effort for: deviations, waivers, engineering change proposals (ECPs), specification change notices (SCNs), Engineering Change Requests (ECRs), and Problem Reports; activities causing a significant number of Change Requests; and rate of baseline changes.

Configuration Audits

Configuration audits are performed independently by CM and product assurance to evaluate the evolution of a product to ensure compliance to specifications, policies, and contractual agreements. Formal audits are performed at the completion of a product development cycle. They are the Functional and Physical Configuration Audits.

The Functional Configuration Audit (FCA) is intended to validate that the development of a configuration item has been completed and it has achieved the performance and functional characteristics specified in the System Specification (functional baseline).

The Physical Configuration Audit (PCA) is a technical review of the configuration item to verify the as-built maps to the technical documentation.

Finally, CM performs periodic in-process audits to ensure that the configuration management process is followed.

Appendix H: Integrated Product and Process Development

Objective:

The objectives of using integrated product & process development are to

- Reduce time to market
- Improve product quality
- Reduce waste
- Save costs through the complete integration of systems engineering life cycle processes

In addition to these tangible outcomes, an integrated product & process development improves team communications through integrated product teams, implements a proactive risk process, make decisions based on timely input from the integrated product team, and improves customer involvement.

Key Terms and Definitions:

Integrated Product & Process Development (IPPD): The process of using an Integrated Product Teams (IPT) to simultaneously develop the design for a product or system and the methods for manufacturing the product or system. The process verification may consist of review of a process description by an integrated product team. It may also include a demonstration to an IPT of a process.

Integrated Product Development Team (IPDT): Is a mulitidisciplinary group of people who are collectively responsible for delivering a defined product or process

Introduction:

Integrated Product Development evolved from recognizing the need to consider all elements of the product life cycle, from conception through disposal, starting at the beginning of the life cycle. Important items to be considered include quality, cost, schedule, user requirements, manufacturing, and support. Integrated product development implies the continuous integration of the entire product team, including engineering, manufacturing, test, and support, throughout the product life cycle. Later, as the importance of *process* was recognized, the terminology was modified to *Integrated Product and Process Development,* or IPPD.

Historically, traditional development took place in series with one activity starting as the preceding one was completed. This is a very lengthy process and the product could become obsolete before it is completed. With good interface definition and control, integrated product development, involving the entire team, can speed up the development process.

Integrated development can introduce *more* risk into a development program, because downstream activities are initiated on the *assumption* that upstream activities will meet their design and interface requirements. However, the introduction of a hierarchy of cross-functional product teams, each developing and delivering a product has been found to actually reduce risks and provide better products faster—as will be discussed.

***Concurrent Engineering (CE)*:** is a management/operational approach which aims to improve product design, production, operation, and maintenance by developing environments in which personnel from all disciplines (design, marketing, production engineering, process planning, and support) work together and share data throughout all phases of the product life cycle.

H.1 Overview of Integrated Product & Process Development

This section will introduce the IPPD concepts and explain why to use them.

H.1.1 Integrated Product Development Team (IPDT) is essential to the IPPD.

An IPDT is a process-oriented, integrated set of cross-functional teams (i.e. an overall team comprised of many smaller teams) given the appropriate resources and charged with the responsibility and authority to define, develop, produce, and support a product or process (and/or service). Process orientation means they are staffed with all the skills necessary to complete their assigned processes—which may include all or some of the development and production steps.

Although the teams are organized on a process basis, the organizational structure of the team of teams may approach a hierarchical structure for the product, depending upon the way the product is assembled and integrated.

Different members of a cross-functional team may have primary, secondary, or minor support roles during different phases of the project cycle. For example, the manufacturing and test representatives may have minor, part-time advisory roles during the early product definition phase, but will have primary roles later, during manufacture and test. The idea is to have them participate to the degree necessary from the outset to insure their needs and requirements are reflected in overall project requirements and planning to avoid costly changes later.

The team <u>must</u> be given both responsibility and authority to get the job done. If no one is in charge, things do not get done. The team should be empowered with authority to do the job. It should not be looking to higher management for its key decisions. It should, however, be required to justify its actions to others, including interfacing teams, the system integration team, and project management.

H.1.2 Why employ IPDTs?

To reduce the risks inherent in concurrent development, industry has learned that IPDTs, using best practices and continuous improvement, have been achieving significant ***process*** improvements, resulting in:

- Seamless interfaces within the teams
- Reduced engineering design time
- Fewer problems in transition from engineering to manufacturing
- Reduced development time and cost

In the early 1990s, companies began to discover that they really could be more productive if they moved away from the hierarchical management structure and organized into product development teams. These teams each mimic a small, independent project to produce its product. Some of the greatest productivity gains have come in three areas:

- Unleashing the team's ingenuity through decentralized processes
- Avoidance of previous problems through new, creative approaches
- Better integration between engineering and manufacturing

The above have led to improved product features, performance, quality, and customer satisfaction.

H.2 Integrated Product Development Team (IPDT) Process

A basic principle of IPDT is to get all disciplines involved at the beginning of the development process to ensure that requirements are completely stated and understood for the full life cycle of the product. This up-front activity is considered part of the Systems Engineering process. Historically, the initial development of requirements has been led by Systems Engineers. In an IPDT, the Systems Engineers still lead the requirements development process, but now more (all) disciplines participate in it.

Requirements are developed initially at the system level, then successively at lower levels as the requirements are flowed down. Teams, led by Systems Engineers, perform the up-front Systems Engineering functions at each level. *This is different* from the previous, classical development approach where Systems Engineers did the up-front work and passed the requirements along to development engineers who passed their designs on to manufacturing, thence to test, without the continuous involvement of the initial engineers. This resulted in a loss of understanding caused by asynchronous communications.

The general approach is to form cross-functional product/process teams for all products and services, plus a Systems Engineering & Integration Team (SEIT) to cover systems issues, balance requirements between product teams, and help integrate the teams. Each of the teams may have members representing the different areas indicated on the left side of the chart.

These team members' participation will vary throughout the product cycle, as the effort transitions from requirements development to conceptual design, through preliminary design and detail design, to manufacturing, assembly and test, to delivery, operational support, and finally retirement (and possibly replacement). It is good for at least some of the team to remain throughout the product cycle in order to retain the team's "project memory."

The product teams do their own internal integration. A SEIT representative belongs to each product team (perhaps several); with both internal and external team responsibilities. There is extensive iteration between the product teams and the SEIT to converge on requirements and design concepts. This effort should slow down appreciably after the preliminary design review, as the design firms up.

There are typically three types of IPDT. These are:

1. Systems Engineering & Integration Team (SEIT)
2. Product Integration Team (PIT)
3. Product Development Team (PDT)

The focus areas for the three types of IPDT teams and their general responsibilities are summarized in Figure H-1. This arrangement is often applicable to large, multi-element, multiple subsystem programs. It must obviously be adapted to the specific project. For example, on smaller programs, the number of PIT teams can be reduced or eliminated. In service-oriented projects, the system hierarchy, focus, and responsibilities of the teams must be adapted to the appropriate services.

Note that the teams are **process** oriented, focusing on components or their integration into more-complex subsystems and elements. The SEIT is used to focus on the integrated system, system processes, external and system issues, which, by their nature, the other teams would possibly relegate to a lower priority.

System Hierarchy	Team Type + Focus Responsibilities
External Interface & System	**System Engineering & Integration Team (SEIT)** • Integrated System and Processes • External & Program Issues • System Issues & Integrity • Integration & Audits of Teams
Element & Subsystem	**Product Integration Teams (PITs)** • Integrated H/W and S/W • Deliverable Item Issues & Integrity • Support to Other Teams (SE & IT and PDTs)
Components, Assemblies, & Parts	**Product Development Teams (PDTs)** • Hardware and Software • Product Issues & Integrity • Primary Participants (Design and Mfg.) • Support to Other Teams (SE & IT and PITs)

These Multi-functional Teams Have Life Cycle (Concept-to-Disposal) Responsibility for Their Products and The System

Figure H-1. Types of IPDTs, their Focus and Responsibilities

Systems engineers participate heavily in the SEIT and PIT and to a much lesser extent in the Product Development Team (PDT). The Systems Engineering processes described in this handbook are just as applicable to all teams in the Integrated Product and Process Development (IPPD) environment as they were in previous styles of organization. The iterative Systems Engineering process is still used. In fact, it is easier to apply the process throughout the program because of the day-to-day presence of Systems Engineers on all teams.

All product teams have many roles. Their integration roles overlap, based on the type of product team and the integration level. Some examples are shown in Figure H-2 for various program processes and system functions. In this figure, Program Processes covers just about anything required on the program. The three bars on the left side show the roles of the three types of product teams at different levels of the system. Note for example that the SEIT **leads** and **audits** in external integration and in system integration activities, as indicated by the shaded bar. But, for those program processes involving parts, components, or subassemblies, the appropriate PDT are the active participants (lead and audit), **supported** by the SEIT and the PIT.

Basic system functions include system requirements derivation, system functional analysis, requirements allocation and flowdown, system trade-off analysis, system synthesis, system integration, technical performance measurement, and system verification. The bars for functions 1, 2, and 3 in the chart show that the SEIT **leads** and **audits** activities on different system functions while the component and subsystem teams participate. The lower level part and subassembly teams support, if requested.

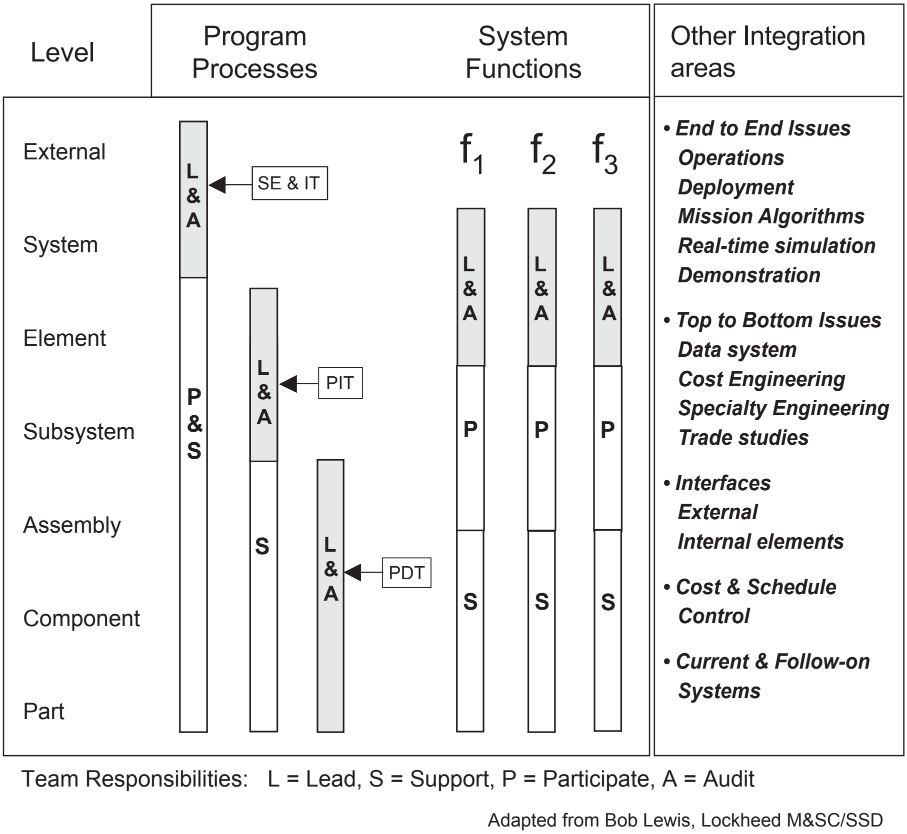


Figure H-2. Examples of Complementary Integration Activities of PDTs

The column at the right side of Figure H-2 shows other integration areas where all teams will have some involvement. The roles of the various teams must also be coordinated for these activities, but they should be similar to the example.

H.3 Steps in Organizing and Running an Integrated Product Development Team (IPDT)

The basic steps necessary to organize and run an IPDT on a project are listed in Table H-1. Each step is discussed in turn, with a summary of the key activities that should take place during the step.

Table H-1. Steps in Organizing and Running an IPDT

Step	Description	Details
1.	Define the IPDT teams for the project	Develop IPDT teams that cover all project areas.
2.	Delegation of responsibility and authority to the IPDT teams	Select experienced team leaders early in the development process and avoid frequent budget changes through-out the life cycle.
3.	Staff the IPDT	Balance – competency and full time commitment and availability for core team. Plan when competencies are needed and not needed. Identify issues where specialist are needed Candidates must: work well in a team environment, communicate well and meet their commitments.
4.	Understand the team's operating environment	Recognize how the team influences directly or indirectly other teams and the project as a whole.
5.	Plan and conduct the "Kick-off meeting"	Recommend two kick off meetings, one for the project as a whole and one for the individual Product Development Teams. Well planned kick off meetings will set the project off on the right foot.
6.	Team training	Training for the project is a critical element. The following are recommended topics that should be covered: • The tailored SE process for the project • Project description, customer stakeholders, purpose, mission organization, schedule budget • Terminology and nomenclature • Access to project products • Communications skills • Team of Teams concept • Project IPDT procedures, metrics, reporting Develop self learning guides that new team members can use to come up to speed on the project when staff turnover occurs.
7.	Define the team vision and objectives	In the initial individual IPDT meetings, do collaborative brainstorming to define the team's vision and objectives, such that each member has an ownership. It most likely will be necessary to bring in other IPDT members, management and customer to flesh out the vision and objectives of the team.
8.	Each team expands the definition of its job	Once the higher level project plan has been reviewed each team must start to identify the tasks, roles and responsibility of the team and each of the members. Clearly define the tasks, milestones, roles for each member. Members need to understand how their tasks fit into the higher level project/ program tasks.
9.	Process assessment and continuous improvement	Each team must document the process they are using and the key metrics to monitor. The teams must have the mindset of continuous improvement and then monitor their own activities and continually make course corrections along the way.

10.	Monitor team progress via metrics and reports	Each team will have a set of metrics and reports to monitor its own performance. These reports and metrics must be reviewed by the Systems Engineering Integration Team that coordinates among the other IPDTs. These metrics may include an Earned Value Report, and technical metrics such as Defect Rate report. The selected metrics are dependent on the team's function on the project.
11.	Sustain and evolve the team throughout the project	Personnel assignments to a team will probably vary over the project cycle. Each team will most likely grow, shrink, and change skill mix over the life cycle phases. As issues occur, technical specialist may need to join the team to help address these specific issues. Functions such as marketing, program controls, procurement, finance, legal, and human resources generally support the team at a steady, low level of effort, or as required
12.	Documentation of team products	The team's products should be well defined and do not usually change significantly. Because of the IPDT structure the overhead of cross-organizational communications varies. Ideally this would be reduced or eliminated. When multiple documents are required, different team members, with identified back ups, should be assigned as the responsible author with contributions from other team members. The IPDT should maintain a log of activities in addition to the mission, vision, objectives, deliverable, meeting minutes, decisions, tailored process, agreements, teams project information, and contact information.
13.	Project closure and follow-on activities	In conjunction with step 12, the IPDT should maintain records as though the project may be re-engineered at some future time and all close out products must be accessible. All IPDT logs should be organized the same way when possible such that they can be easily integrated into an overall project report. The close out should include lessons learned, recommended changes, and a summary of metrics for the team.

H.4 Potential Integrated Product Development Team (IPDT) Pitfalls versus High Performance

There are ample opportunities to go astray, before team members and leaders go through several project cycles in the IPDT framework and gain the experience of working together. There are some things teams should watch out for. Table H-2 describes eight pitfalls. Table H-3 lists ten techniques for high performance in an IPDT.

Table H-2. Pitfalls of using IPDT

	IPDT Pitfalls	What to do
1.	Spending too much time defining the vision and objectives	Converge and move on
2.	Insufficient authority – IPDT members must frequently check with management for approval	Give team leader adequate responsibility, or put the manager on the team
3.	IPDT members are insensitive to management issues and over commit or overspend	Team leader must remain aware of overall project objectives and communicate to team members
4.	Teams are functionally-oriented rather than cross-functional, process-oriented	Review step 1 of Table H-1
5.	Insufficient continuity of team members throughout the project	Management should review staffing requirements
6.	Transition to the next phase team specialists occurs too early or too late in the schedule	Review staffing requirements
7.	Overlapping assignments for support personnel compromises their effectiveness	Reduce the number of teams
8.	Inadequate project infrastructure	Management involvement to resolve

Obviously, some things do require checking with higher authority. Encourage team members to anticipate these from the outset. Functional managers/supervisors, if any, must remain aware of major team issues and coach/guide/train participants until they gain the requisite experience.

Project managers should review team staffing plans to ensure proper composition and strive for continuity of assignments. It has been observed that the advantages of a full time contributor outweigh the work of many part-time team members. The loss of a key team member who knows how and why things are done can leave the team floundering.

On product teams it is important to have people who can work well together and communicate. However, team results may be condemned to mediocrity by avoiding those outstanding technical/specialist professionals who can really make a difference.

Table H-3. Ten Techniques for High Performance in Integrated Product Development Teams (IPDTs)

	Recommended technique
1.	Careful selection of staff – excellent people do excellent work
2.	Establish and maintain positive team interaction dynamics; all should know what is expected of the team and each individual, all should strive to meet commitments, interactions should be informal but efficient, and a “no blame” environment where problems are fixed and the team moves on
3.	Generate team commitment and buy-in to the vision, objectives, tasks and schedules
4.	Breakdown the job into manageable activities that can be accurately scheduled, assigned and followed-up on weekly
5.	Delegate and spread out routine administrative tasks among the team; frees the leader to participate in technical activities, give every team member some administrative/managerial experience.
6.	Create a “world class” analysis and simulation capability for requirements and performance to be better than the competition
7.	Schedule frequent team meetings with mandatory attendance for quick information exchanges; everyone is current; assign action items with assignee and due date
8.	Maintain a Team Leader’s Notebook
9.	Anticipate and surface potential problems quickly (internally and externally)
10.	Acknowledge and reward good work

Appendix I: *Requirements Definition Process*

The purpose of this appendix is to elaborate and provide "How-to" information on requirements identification, capture, analysis, and management. This appendix is a complement to Sections 4.2: Stakeholder Requirements Definition Process, Section 4.3: Requirements Analysis Process, and Section 7.2: Requirements Management. Other key information on requirements can be found in the EIA 632 Standard – Processes for Engineering a System (Requirements 14, 15, and 16, and Annex C3.1 a, b, and c).

Requirements are the foundation of the project. They form the basis for design, manufacture, test, and operations. Each requirement carries a cost. It is therefore essential that a complete, but minimum set of requirements be established early. Changes in requirements later in the development cycle can have a significant cost impact on the project, possibly resulting in cancellation.

The objective of requirements analysis is to identify and express verifiable requirements that state user needs in appropriate terms to guide system concept development. Requirements analysis, like the total Systems Engineering process, is an iterative activity in which new requirements are identified and constantly refined as the concept develops and additional details become known. These are analyzed, and deficiencies and cost drivers are identified and reviewed with the customer to establish a requirements baseline for the project.

A second objective of the requirements analysis is to provide an understanding of the interactions between the various functions and to obtain a balanced set of requirements based on user objectives. Requirements are not developed in a vacuum. An essential part of the requirements development process is the concept of operations, the implicit design concept that accompanies it, and associated demands of relevant technology. Requirements come from a variety of sources. Some requirements come from the customer/user, some come from regulations/codes, and some come from the corporate entity. Figure I-1 illustrates this environment.

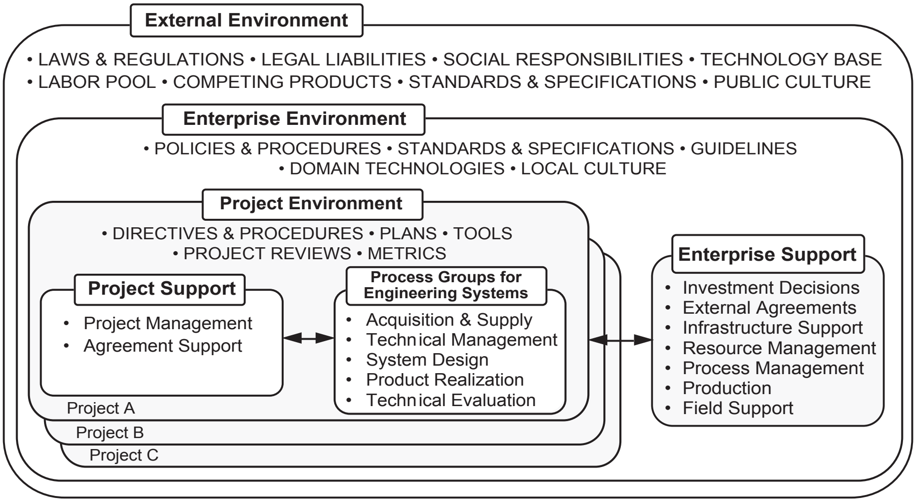


Figure I-1. Sources of Requirements

Requirements definition is a complex process that employs performance analysis, trade studies, constraint evaluation and cost/benefit analysis. System requirements cannot be established without determining their impact (achievability) on lower level elements. Therefore, requirements definition is an iteration and balancing process that works both "top-down" (called allocation and flowdown) and "bottom-up." Once the top-level set of system requirements has been established, it is necessary to allocate and flow them down to successively lower levels. As the allocation and flowdown process is repeated, it is essential that traceability be maintained to assure that all system level requirements are satisfied in the resulting design. The resulting requirements database usually contains many attributes for each requirement, and is also used in verification

To describe the requirements analysis process in more detail, this section is broken down into six subsections: capturing source requirements, Concept of Operations definition, refinement, requirements allocation and traceability, development of the resulting Specifications, and iterating between requirements design loops in the systems engineering process.

I.1 Capturing Source Requirements

This section discusses methods for developing requirements from user objectives and the customer's preliminary requirements. Eliciting and capturing requirements is one of the first steps in the requirements process. Requirements elicitation is an iterative activity and benefits from continuous communication and validation with stakeholders.

The Systems Engineering team leaders extract, clarify, and prioritize all of the written directives embodied in the source documents relevant to the particular phase of procurement activity. Examples of typical inputs include (but are not limited to):

a. New or updated customer needs, requirements, and objectives in terms of missions, measures of effectiveness, technical performance, utilization environments, and constraints
b. Technology base data including identification of key technologies, performance, maturity, cost, and risks
c. The outputs from the preceding acquisition phase (reference Section 6.7: Acquisition Process and Appendix F: Acquisition and Supply – Defining Needs)
d. Requirements from contractually cited documents for the system and its configuration items
e. Technical objectives
f. Records of meetings and conversations with the customer

The source requirements gained by carrying out this function are only a portion of the total system requirements. They will be expanded by a number of activities as follows:

- The key activities described below to break down the broad requirements statements will reveal the need for additional clarification which will lead to either revision of the written source material or additional source documents such as clarification meeting minutes.
- The Concept of Operations definition function, covered below in Section I.2 may reveal the need for additional clarification.

The primary objective is to establish a database of baseline system requirements traceable to the source needs and requirements, to serve as a foundation for later refinement and/or revision by subsequent functions in the Systems Engineering process. As a minimum, this foundation must include the following:

a. Project requirements
b. Mission requirements
c. Customer specified constraints
d. Interface, environmental, and non-functional requirements
e. Unclear issues discovered in the requirements analysis process
f. An audit trail of the resolution of the issues raised
g. Verification and validation methods required by the customer.
h. Traceability to source documentation.

Prerequisites for the successful performance of this function are:

a. Empower a systems analysis team with the authority and mission to carry out the function. (See Appendix H: Integrated Product and Process Development)
b. Assign experienced Systems Engineer(s) to lead the team.

c. Assign experienced team members from relevant engineering, test, manufacturing, and operations (including logistics) disciplines to be available on call to the team.
d. Establish the form of a decision mechanism (such as a design decision database) and any supporting tools; select and obtain necessary SE tools for the function.
e. Complete the relevant training of team members in the use of tools selected for the function.
f. Define the formats of the output deliverables from this function (to permit the definition of any database schema tailoring which may be needed).

I.1.1 Initializing the Requirements Database

The requirements database must first be populated with the source documents that provide the basis for the total set of system requirements that will govern its design. Source documents used as inputs will include statements of user objectives, customer requirements documents, marketing surveys, systems analysis, concept analyses and others. These source or originating requirements should be entered in the database and disseminated to all team members assigned to the requirements analysis team. The information should also be accessible for rapid reference by other project personnel.

Recommended Activities

1. Take the highest priority source document identified and ensure that it is recorded in the database in a manner such that each paragraph in the source document is recorded as a separate requirements object. Record information to trace each such requirements object back to the identity of:
   - The source document identity
   - The paragraph title
   - The sentence number

   (One reason for selecting a paragraph as the parent requirement object is to evaluate later change impact analyses. Most changes to source documents are flagged by a change bar against paragraphs which have been modified or deleted.)

2. Analyze the content of each parent requirement object produced in the previous step. Based on its engineering content determine the following:
   - Does the parent object contain any information on requirements or systems objectives? If it does, it should meet the following critera:
     - Unambiguous,
     - Non-conflicting with other requirements
     - Uniquely assignable to a single system function, architectural component, performance measurement index, or system constraint

If it meets the previous criteria, bypass the mini-steps below and move to the next parent requirement object.

If it does not meet these criteria, based on the engineering content, determine a strategy for decomposing the parent requirement object into separate but related pieces with the objective of meeting these criteria.

- Record information in the database to provide vertical traceability from the parent requirement object to the child requirement object using the Project Unique Identifier (PUID) discussed in Section I.4.
- Repeat the procedure with child objects as necessary, creating and recording traceability to grandchild, great-grandchild to the lowest level requirement. Stop fragmentation at the level when the decomposition objective has been achieved. This is called a leaf node requirement. Each branch of the tree may be decomposed down to a different level, depending on the complexity and the system acquisition approach (make or code everything or use suppliers or subcontractors for one or more branches). As the requirements are flowed down, this will eventually end at a Configuration Item (CI). A CI is a hardware, software, or composite item at any level in the system hierarchy designated for configuration management. CIs have four common characteristics:
    1. Defined functionality,
    2. Replaceable as an entity,
    3. Unique specification,
    4. Formal control of form, fit, and function

3. Repeat steps 1 and 2 for lower priority source documents.

<u>Methods/Techniques</u>

Techniques for requirements elicitation include interviews, focus groups, and the Delphi technique.

I.1.2 Generation of the System Requirements Document (SRD)

The output of this function will be a baseline set of complete, accurate, non-ambiguous system requirements, recorded in the requirements database, accessible to all parties.

To be non-ambiguous, requirements must be broken down into constituent parts in a traceable hierarchy such that each individual requirement statement is:

- Clear, unique, consistent, stand-alone (not grouped), and verifiable
- Traceable to an identified source requirement
- Not redundant, nor in conflict with, any other known requirement
- Not biased by any particular implementation.

Note that these objectives may not be achievable using source requirements. Often requirements analysis is required to resolve potential conflicts and redundancies, and to further decompose requirements so that each applies only to a single system function.

Recommended Activities

1. Periodically during the analysis process, it is desirable to be capable of generating a "snapshot" report of clarified system requirements. To aid this process it may be desirable to create a set of clarified requirement objects in the database with information providing traceability from their corresponding originating requirement. Clarified requirements may be grouped as functional, performance, constraining, and non-functional.

2. Generate a draft System Requirements Document (SRD) if one does not already exist. Use of an automated database will greatly facilitate this effort, but is not explicitly required. This is the highest level document to be created by the project to represent the customer/user requirements. If a SRD already exists, review it internally and with your customer to ensure that it is valid, meets the customer needs, and that you understand it.

I.2 Concept Of Operations

Early in the requirements definition process Systems Engineering produces a Concept of Operations (ConOps) document, sometimes also called the Operational Concept Document (OCD), to describe what the system will do (not how it will do it) and why (rationale). It should also define any critical, top-level performance requirements or objectives (stated either qualitatively or quantitatively) and system rationale. The ConOps should contain a preliminary functional block diagram of the system with only the top-level functional "threads" specified. No attempt is made at this stage to define a complete operational concept or to allocate functions to hardware or software elements (this comes later). This concept of operations is essentially a functional concept definition and rationale *from the user and customer perspective.*

The concept of operations defines the way the system will be used and must involve the input from stakeholders such as operations, maintenance, and management personnel. The concept of operations document should contain the roles and responsibilities and the set of skills needed for operations and maintenance of the system.

As discussed in Section 4.2: Stakeholder Requirements Definition Process, other concept documents should also be considered: Concept of Production, Concept of Deployment, Concept of Support, Concept of Disposal.

Objective

The primary objective is to communicate with the end user of the system during the early specification stages to ensure that operational needs are clearly understood and incorporated into the decision mechanism for later inclusion in the system and lower level specifications.

Recommended Activities

The concept of operations consists of describing system behavior, starting with outputs generated by external systems (modified as appropriate by passing through the natural system environment) which act as stimuli to the system, causing it to take specified actions and produce outputs which are absorbed by external systems. These single threads of behavior are traced from source document statements and cover every aspect of operational performance, including logistical modes of operation, operations under designated conditions, and behavior required when experiencing mutual interference with multi-object systems.

Aggregation of these single threads of behavior represents a dynamic statement of what the system is required to do. In some cases, the word "scenario" is used to describe a single thread of behavior, and in other cases it describes a superset of many single threads operating concurrently.

1. Start with the source operational requirements; deduce a set of statements describing the higher-level, mission-oriented system objectives and record them.
2. Review the system objectives with end users and operational personnel and record the conflicts.
3. Define and model the operational boundaries.
4. For each model, generate a context diagram to represent the model boundary.
5. Identify all of the possible types of observable input and output events that can occur between the system and its interacting external systems.
6. If the inputs/outputs are expected to be significantly affected by the environment between the system and the external systems, add concurrent functions to the context diagram to represent these transformations and add input and output events to the database to account for the differences in event timing between when an output is emitted to when an input is received.
7. Record the existence of a system interface between the system and the environment or external system.
8. For each class of interaction between a part of the system and an external system, create a functional flow diagram to model the sequence of interactions as triggered by the stimuli events generated by the external systems.

9. Add information to trace the function timing from performance requirements and simulate the timing of the functional flow diagrams to confirm operational correctness or to expose dynamic inconsistencies. Review results with users and operational personnel.

10. Develop timelines, approved by end users, to supplement the source requirements.

Draft Concept of Operations (ConOps) are prepared in early project phases, such as concept definition studies or pre-proposal studies. Usually concepts will evolve, and prior drafts should be updated for the next project phase.

Input

The following typical source documents serve as inputs for the Concept of Operations:

- System business case
- Statement of User Need
- Technical operational requirements
- System operational requirements documents
- Statement of operational objectives
- Statement of Work (SoW)
- Customer Standard Operating Procedures (SOPs)

Output

A ConOps comprising:

- Operations, production, deployment, support, and disposal
- A top-level operational concept definition containing approved operational behavior models for each system operational mode (which can be documented as functional flow diagrams), supporting time lines, and event transcripts, which are fully traceable from source requirements
- Context diagrams
- Mission Analyses

End Result

Understanding of operational needs will typically produce:

- A source of specific and derived requirements that meet the customer and user needs and objectives.
- Invaluable insight for Integrated Product Development Team (IPDT) members as they design, develop, deliver, verify and validate the system
- Diminished risk of latent system defects in the delivered operational systems.

This activity is generally concluded when the Concept of Operations Document (ConOps) is released and approved by the System Requirements Review.

Metrics

1. Functional Flow Diagrams required and completed;
2. Number of system external interfaces;
3. Number of scenarios defined;
4. Number of unresolved source requirement statements;
5. Missing source documents;
6. Number of significant dynamic inconsistencies discovered in the source requirements.

Methods/Techniques

Interviews with operators of current/similar systems and potential users, Interface Working Group meetings, Context Diagrams, Functional Flow Block Diagrams (FFBD), time-line charts, N^2 charts.

I.3 Define/Derive/Refine Functional/Performance Requirements

At the beginning of the project, Systems Engineering is concerned primarily with user requirements analysis – leading to the translation of user needs into basic functions and a quantifiable set of performance requirements that can be translated into design requirements.

Functional/performance requirements definition/derivation/refinement covers the total system over its life cycle, including its support requirements. These need to be formally documented requirements that define the functions and interfaces and characterize the system by performance requirements that can be flowed down to hardware and software designers.

Participation/Stakeholders

All Systems Engineering groups will be involved in this activity. In the early phases (up through System Requirements Review (SRR)), this is the primary Systems Engineering activity, with significant support from the design engineering organizations. The customer is also a key stakeholder and validates the work as it progresses.

Recommended Activities

Establishing a total set of system requirements is a complex, time consuming task involving nearly all project areas in an interactive effort. It must be done early, since it forms the basis for all design, manufacturing, test, operations, maintenance, and disposal efforts, and therefore determines the cost and schedule of the project. The process is iterative for each phase, with continuous feedback as the level of design detail increases. The overall technical process is shown in Figure I-2.

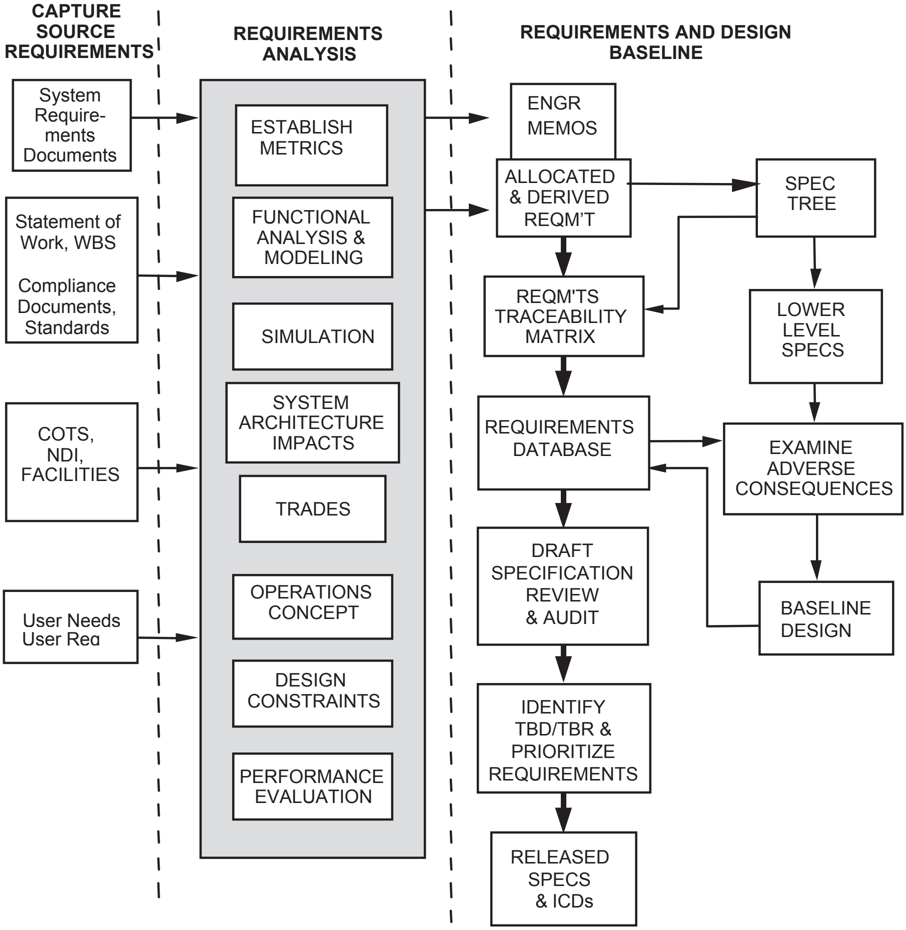


Figure I-2. Requirements Derivation, Allocation, and Flowdown Process

The following paragraphs describe the process steps; however, some steps are concurrent and others are not always done in the order shown.

1. The starting point is the set of source requirements developed as described in Section I.1. Establish constraints on the system including:

- Cost
- Schedule
- Use of Commercial Off-The-Shelf (COTS) equipment
- Use of Non-Developmental Items (NDI)
- Use of Existing Facilities
- Operational Interfaces with other systems or organizations
- Operational environment

As a result of this activity, a number of functional and performance requirements are identified.

2. The mission should then be examined and characterized in measurable requirement categories such as: Quantity, Quality, Coverage, Timeliness, and Availability. Actual systems have many measurables under each attribute, and additional attributes, for example communications, command and control, and security.

3. Use detailed functional analysis to extract new functional requirements, particularly those required to support the mission (the ConOps is a rich source for the analysis). This includes items such as power, propulsion, communications, data processing, attitude control or pointing, commanding, and human interaction and intervention. This will eventually result in the conversion from mission parameters (customers supported per node) into parameters that the hardware and software designers can relate to, for example Effective Radiated Power (ERP), and Received Signal Strength Intensity (RSSI). Functional decomposition tools such as functional block diagrams, functional flow diagrams, time lines, and control/data flow diagrams are useful in developing requirements. As requirements are derived, the analysis that leads to their definition must be documented and placed into the requirements database.

Quality Function Deployment (QFD) is a useful technique, particularly where the "voice of the customer" is not clear (see Figure I-3). It provides a fast way to translate customer requirements into specifications and systematically flowdown the requirements to lower levels of design, parts, manufacturing, and production.

- A Planning Tool for Translating Customer Requirements into Specifications
- Deploys the "Voice of the Customer"
- A Fast, Systematic Way to Derive & Flow Down Requirements to Design, Parts, Manufacturing, and Production

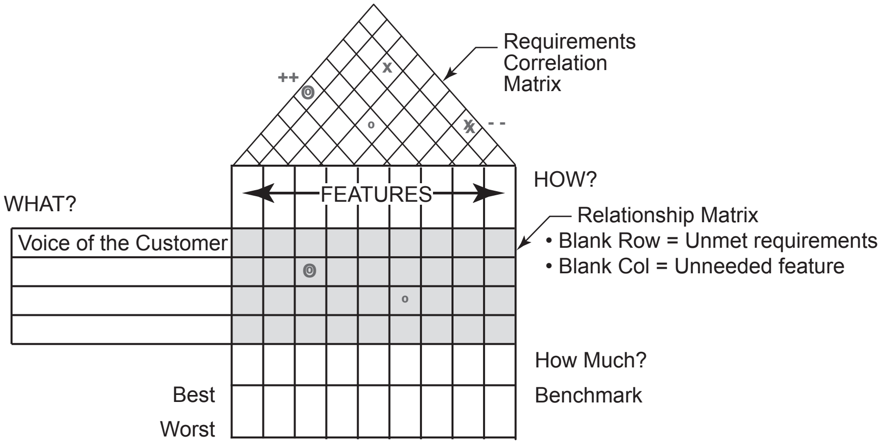


Figure I-3. Quality Function Deployment (QFD); The House of Quality

The shaded Relationship Matrix shows the correlation between features and requirements. Two concentric circles (double circle) are used to indicate a strong correlation between the feature and the requirement. A modest contribution is indicated by a single circle. A blank column indicates an unnecessary feature relative to the listed requirements. Similarly, a blank row indicates an unaddressed requirement.

4. For larger systems, develop a high-level system simulation evolved from the system architecture. The simulation should contain sufficient functional elements that the interactions can be properly assessed. The purpose of the simulation is to establish measurable parameters for the functional requirements developed above, and convert them wherever possible from functional requirements to performance requirements. This provides the necessary guidance to the designers on the size and capability required of their equipment. In addition, these parameters will be used as an integral part of the verification process in establishing the capability of the equipment (and the system) to satisfy user needs. The simulation will be used to quickly examine a range of sizes and parameters, not just a "Point Design". This will ensure that the "best" solution is obtained – the system is the proper size throughout, with no choke points. Exercise the simulation using scenarios extracted from the ConOps with inputs based on system requirements. A number of scenarios should be run to exercise the system over the possible range of mission activities. Monte Carlo runs may be made to get averages and probability distributions. This will help establish (or verify) timeliness requirements. In addition to examining nominal conditions, non-nominal runs should also be made to establish system reactions or breakage when exposed to extraordinary (out-of-spec) conditions.

5. Examine any adverse consequences of incorporating requirements:
   - Is unnecessary risk being introduced?
   - Is the system cost within budget limitations and the budget profile?
   - Will the technology be ready for production?
   - Are sufficient resources available for production and operation?
   - Is the schedule realistic and achievable (be sure to consider downstream activities such as design and verification associated with the requirements)?

6. Where existing user requirements cannot be confirmed, trade studies should be performed to determine more appropriate requirements, and achieve the best-balanced performance at minimum cost. Where critical resources, for example Weight, Power, Memory, and Throughput. must be allocated, trade studies may be required to determine the proper allocation.

7. Incorporate revised and derived requirements and parameters into the requirements database and maintain traceability.

8. Prepare the specification documents (see Section I.5) and submit to all organizations for review. Then enter the documents into the formal release system, and maintain them under configuration management control. Any further changes will require Configuration Control Board (CCB) approval.

In defining the requirements in a specification, care should be exercised to assure the requirement is appropriately crafted. The following questions should be considered for every requirement:

1. **Is each requirement clear?** Requirements must convey what is to be done to the next level of development. Its key function is to communicate. Is the requirement clear, compatible and complete? Is it possible to interpret the requirement in multiple ways? Are the terms defined? Does the requirement conflict or contradict another requirement?

2. **Is each requirement a proper requirement?** A requirement's specification is a demand on the designer (or implementer) at the next level. Is this requirement at the proper level? Customer requirements may be imposed at any level they desire; however, when customer requirements specify design, it should be questioned. When generating requirements, the requirements should be targeted at the next lower level and no lower (except when carrying forward a legitimate customer design requirement or constraint). A proper requirement should deal with the entity being specified as a "black box" describing what transformation is to be performed by the "box". The requirement should specify "what" is to be done at that level, not "how" it is to be done at that level.

3. **Is the requirement necessary?** Every requirement generates extra effort in the form of processing, maintenance, and testing. Only necessary requirements

should be written. Unnecessary requirements are of two varieties: (1) unnecessary specification of design which should be left to the discretion of the designer, and (2) a redundant requirement covered in some other combination of requirements.

4. **Is each requirement consistent with product standards?** In many instances, there are applicable government, industry and product standards, specifications, and interfaces with which compliance is required. An example might be additional requirements placed on new software developments for possible reusability. Another might be standard test interface connectors for certain product classes.

5. **Is each requirement achievable?** It is imperative that the implementing designer participate in requirements definition. The designer should have the expertise to assess the achievability of the requirements. In the case of items to be subcontracted, it's important that the expertise of potential subcontractors be represented in the generation of the requirements. Additionally, participation by manufacturing and customers/users can help assure achievable requirements. IPDTs and requirements reviews provide mechanisms to achieve these perspectives.

6. **Do the requirements pass the traceability test?** Do all requirements trace to the higher level specification? Are there requirements at the higher level not allocated (or allocated, but not picked up)? Those with no allocation may be satisfied at that level of the specification. Requirements with either deficiency should be corrected. Refer to section I.4 for more information.

7. **Is each requirement verifiable?** Each requirement must be verified at some level by one of the four standard methods (test, demonstration, analysis, or inspection). A customer may specify, "The range shall be as long as possible." This is a valid but unverifiable requirement. This type of requirement is a signal that a trade study is needed to establish a verifiable maximum range requirement. Each test requirement should be verifiable by a single test. A requirement requiring multiple tests to verify should be broken into multiple requirements. There is no problem with one test verifying multiple requirements; however, it indicates a potential for consolidating requirements. When the system hierarchy is properly designed, each level of specification has a corresponding level of test during the test phase. If element specifications are required to appropriately specify the system, element verification should be performed.

Requirements must be written with extreme care. The language used must be clear, exact, and in sufficient detail to meet all reasonable interpretations. A glossary should be used to precisely define often-used terms or terms that could have multiple interpretations. In most writing, it is desirable to substitute words that are more or less synonymous in order to avoid the constant repetition of a word. However, because few words are exact synonyms, requirements should be written using the same wording with exact meaning established. Care must be taken in utilizing clear, unambiguous phraseology and punctuation. A misplaced comma can have dramatic ramifications.

Verb tense and mood in requirements specifications are very important. The following describes the common use of the forms of the verb “to be” as they apply to specifications:

- “Shall” – Requirement specifications are demands upon the designer or implementer and the resulting product, and the imperative form of the verb, “shall”, shall be used in identifying the requirement.
- “Will” – statement containing “will” identifies a future happening. It is used to convey an item of information, explicitly not to be interpreted as a requirement. “The operator will initialize the system by …” conveys an item of information, not a requirement on the designer of his product. However, some organizations have dropped the distinction between “shall” and “will” in specifications, and treat either word as a means of stating a requirement.
- “Must” – “Shall” is preferable to the word “must”. If both are used in a requirements specification, there is an implication of difference in degree of responsibility upon the implementer.
- Other forms – “To be”, “is to be”, “are to be”, “should” and “should be” are indefinite forms of the verb, and they should be minimized in requirement specifications.

The imperative mood may be used as well in specifying requirements. For example, “The database shall be dumped to magnetic tape every four hours.” Requirements done in table format, usually express the processing requirements in the imperative mood. Judicious use of the imperative mood can eliminate many words and enhance the readability of specifications.

There are words whose use should be avoided in requirements in that they covey uncertainty. These include:

- Pronouns. Pronouns should be avoided or used with care in that they can lead to ambiguity or confusion as to exact meaning. Words such as “he”, “she”, “this”, “they”, “their”, “who”, “it”, and “which”, should be used sparingly, if at all.
- Adjectives and adverbs. Adjectives and adverbs generally convey an indefinite degree. Words such as “timely”, “real-time”, “precisely”, “appropriately”, “approximately”, “various”, “multiple”, “many”, “few”, “limited”, and “accordingly” should be avoided in requirements.
- Other indefinites. Words or phrases such as “etc.” and “and so on” usually indicate an unbounded list. “To be determined” is generally an official flag of uncertainty. If used, “to be determined” along with “to be supplied” and “to be reviewed” should be logged and documented in a table at the end of the specification with an

assigned person for closure and a due date. The word "process" needs to be used with care. When used, that processing must be clearly defined.

Characteristics of ***the group*** of requirements as a whole, such as:

- Are all requirements mutually consistent?
- Have redundant requirements been avoided?
- Are there any requirements missing?
- Are the requirements structured in a useful and sensible way (e.g. according to requirement type, according to the problem space model, according to the standard template, etc.)?

Input

System Requirements Document, Statement of Work, Company Policies and Procedures, Concept of Operations Document (or Operations Concept Document), Design Concept, System Hierarchy, and Data Item Description (identifies expected content for specifications).

End Result

The result of performing this requirements analysis function should be a baseline set of complete, accurate, non-ambiguous system requirements, recorded in the requirements database, accessible to all parties, and documented in an approved, released System Specification.

Metrics

1. Number or percent of requirements defined, allocated, and traced;
2. Time to issue draft;
3. Number of meetings held;
4. Number and trends of To Be Determined (TBD), To Be Resolved (TBR), and To Be Supplied (TBS) requirements;
5. Number of requirement issues identified (e.g. requirements not stated in a verifiable way);
6. Number and frequency of changes (additions, modifications, and deletions).

Methods/Techniques

Functional decomposition using a system hierarchy, functional block diagrams, functional flow diagrams, time lines, control/data flow diagrams, trade studies, requirements allocation sheets, and Quality Function Deployment.

I.4 Requirements Allocation and Traceability

Traceability is not an end goal in and of it self, but rather, is a tool that can be used to:

1. improve the integrity and accuracy of all requirements, from the system level all the way down to the component level,
2. allow tracking of the requirements development and allocation and generating overall metrics, and
3. support easier maintenance and change implementation of the system in the future.

The essential point is that every requirement at every level should have a clear definition of its source and why it is needed.

Derived requirements (i.e. those requirements that are indirectly traced to higher level requirements; see definition in RTCA/DO-178B) must be documented and their justification given. Other requirements (i.e. the primary requirements) are justified by virtue of the fact that they trace directly to a higher level requirement.

Traceability should be maintained throughout all levels of documentation; bi-directional traceability is top-down and to verification and validation plans and procedures for specifications (CI and interface), and should include traceability to the test program (plans, procedures, test cases, and reports) to provide closed loop verification.

1. Allocate all system requirements to hardware, software, or manual operations, facilities, interfaces, services, or others as required;
2. Ensure that all functional and performance requirements or design constraints, either derived from or flowed down directly to a system architecture component, have been allocated to a system architecture component
3. Ensure that traceability of requirements from source documentation is maintained through the project's life until the test program is completed and the system is accepted by the customer; and
4. Ensure that the history of each requirement on the system is maintained and is retrievable.

Recommended Activities

1. While requirements can be traced manually on small projects, such an approach is generally not considered cost-effective, particularly with the proliferation of requirements management tools. A requirements traceability tool that augments the requirements database should be accessible to and usable by all technical personnel on the project. This includes subcontractors who are preparing specifications and verification data.

The tool should generate the following directly from the database:

a. Requirements Statements with Project Unique Identifiers (PUID)
b. Requirements Traceability Matrices (RTM) – list requirements and their traces
c. Verification Cross Reference Matrices (VCRM) – list requirements and their verification attributes
d. Lists of TBD, TBR, and TBS
e. Specifications
f. Requirements metrics (e.g. requirements stability)

The tool must have configuration management capability to provide traceability of requirements changes, and ensure that only properly authorized changes are made

2. Each requirement must be traceable using a Project Unique Identifier (PUID). The specification tree provides the framework for parent-child vertical traceability (tree-down or tree-up) used for specifications. For interface documents such as Interface Control Documents (ICDs) the traceability is – in some cases over several levels. Thus, the specification tree does not adequately portray interface traceability. However, the requirements tool must have capability for bi-directional traceability top-down and to verification plans and procedures. The PUID is an alphanumeric assigned to each requirement. The alphanumerics employed are similar to acronyms in order to provide an easily recognizable identification of the specification for the requirements statements. This is particularly useful when requirements statements are extracted from many specifications as part of the audit process. The numeric portion is assigned within individual documents.

3. Identify the functions and subfunctions for which each area is responsible, and the top level system requirements associated with those functions. The process for the identification of, and allocation to, subfunctions is described in Appendix J: Functional Analysis and Allocation. Assign a PUID to each of the functions (system actions) and subfunctions. For each system action, identify functional/performance requirements to be associated with it. Capture this association in the database. For each function and subfunction, identify which system component in the system architecture is responsible for it, and capture this information in the database.

4. The most difficult part of requirements flowdown can be the derivation of new requirements, which often involves a change in the parameters as appropriate to the level in the hierarchy (targets per sq. mi – a system parameter – has little meaning to the hardware designer). Repeat the process at each level until the appropriate level is reached. Each branch of the system tree may be decomposed down to a different level, depending on the complexity and the system acquisition approach (make or code everything or use suppliers or subcontractors for one or more branches). At the lowest Configuration Item (CI) level to be defined, the parameters specified must be relevant to that particular item (hardware, software, or composite), and provide adequate direction to the designer.

5. Audit the specifications as they are produced to verify that the allocation process is correct and complete. Use the Requirements Database to generate audit reports that contain the flowdown of requirements statements. Identify proposed corrections and changes, and process them through the proper approval channels.

6. Generate Requirements Traceability Matrices (RTM) from the database.

Input

Specification Tree and Systems Requirement Document (SRD) or System Specification

The initial definition of system requirements from the source documents defined in this appendix, Section I.1, is completed using a combination of graphical functional analysis tools and simulations as described in Appendix J. As the requirements are developed, a design concept and a concept of operations Section I.2 are developed concurrently. The output of this effort is a set of requirements statements, which are placed in the System Specification as described in Section I.3. A specification tree (Section I.5) is developed first that identifies all requirements documents on the program and provides the hierarchy for requirements flowdown and traceability.

Output

- **Specifications** – The primary output of the Requirements Database is specifications. Draft specifications are generated by the database, and distributed to reviewers. The copies are returned with comments as appropriate, to the author. When all comments are resolved, the document is formally released. The Requirements Database tool should generate the specification directly from the database without manual intervention, thereby preserving the integrity of the database.

  Note that specifications at one level represent requirements to the levels below it.

- **Requirements Traceability Matrices** – The Requirements Traceability Matrices (RTMs) are generated directly from the database, and are also used as part of the audit process.

- **Status Reports** – As the system acquisition cycle proceeds, increasing effort will be directed toward verification that the demonstrated capability of the system meets its requirements as expressed in specifications. The database plays a major role in this by incorporating the verification data in its attribute files, either directly or by pointer to other databases where the data are located. Status reports on verification progress, progress in eliminating undefined requirements (To Be Determined (TBD)/ To Be Resolved (TBR)/ To Be Supplied (TBS)), and requirements changes can be obtained by sorting the appropriate attribute listings.

End Results

Traceability is achieved when all requirements at a particular level of the system hierarchy have been placed in the database and traced up and down, as appropriate. A complete set of allocated requirements should be found in specifications, with a Requirements Traceability Matrix (RTM).

Metrics

1. Number and trends of requirements in the database;
2. Number of TBD, TBR, and TBS requirements;
3. Number (or percent) of system requirements traceable to each lower level and number (percent) of lower level requirements traceable back to system requirements.

Methods/Techniques

A large variety of tools are available for requirements management and systems architecting. Since the information on these tools becomes outdated approximately every six months, INCOSE has elected to maintain a current database on SE tools available to anyone at its World Wide Web site.

I.5 Development of Specification Tree and Specifications

Create a Specification Tree for the system and specifications for each configuration item of the system under development. This activity represents the establishment of the documented baseline of a particular level of the system design. For complex systems there may be multiple iterations performed wherein the definition/design process is successively applied in a hierarchical manner down to the level of hardware and software configuration item definition. These baselines are captured in configuration item specifications. The road map and hierarchical representation of the specifications is the Specification Tree. The specifications document the set of configuration items for example, hardware, software, and operations, which will implement the system.

The objective is to create a specification baseline for each of the configuration items and place these specifications in a flowdown hierarchy. This will allow the further definition of each configuration item to proceed independently, in parallel with all the others, while maintaining requirements traceability and compatibility of all items that make up the system.

In practice, requirements engineering is not just a front-end to the system development process but a complex communication and negotiation process involving the parties that will use the system, i.e. the customers; the parties that will provide parts or all of the system, i.e. the developers and vendors; and the parties that will test the system, i.e. the test group(s). Systems Engineering acts as the translator in this communications process with the specifications being the key written embodiment of this communication. Some of the major challenges in performance of this requirements engineering task are:

- An envisioned system is seldom, if ever, designed to work totally independent of the other systems in the customer's environment. This means that the environment in which the system is to operate must be known and documented as thoroughly as the system itself.

- Off-the-shelf solutions or components play a major role in defining the system. While requirements are supposed to be independent of solution, being able to achieve an implementable solution within the resource constraints available is the primary requirement.

- Every aspect of an envisioned system's function and performance cannot practically be specified. Thus, a level of requirement specification must be established which represents a cost-effective balance between the cost of generating, implementing, and testing requirements versus the risk of not getting a system meeting customer's expectations. In each case, the cost of non-performance is a major driver.

The Systems Engineering process is a bridging process translating an identified need into a system solution composed of specified implementable hardware and software elements. The process is very much a communication process with all the potential flaws of any communication, plus the added uncertainty of the customer’s real desires and the risks associated with achieving an implementation.

Participation

This function is led by Systems Engineering, with support from design engineering and the supporting disciplines.

Systems Engineering creates the Specification Tree, the outlines for each of the specifications, crafts the requirements, and establishes traceability. Systems Engineering also ensures that the supporting disciplines are present when needed and actively participate, and ensures that their contributions are coordinated and integrated.

Design engineering provides technical definition data for derived requirements, and documents design decisions.

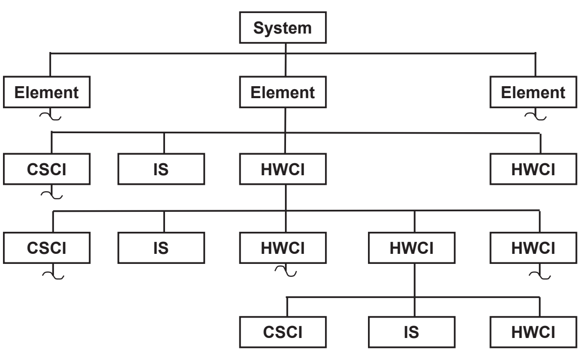
Supporting disciplines monitor implementation of requirements in each specialty area, identify requirements, and review the results of the requirement definition process.

Recommended Activities

1. Derive the Specification Tree from the system architecture configuration

As discussed in Appendix E: The Hierarchy WITHIN A System, the system hierarchy should be a balanced hierarchy with appropriate fan-out and span of control. A level of design with too few entities likely does not have distinct design activity, and both design and testing activities contain redundancy. Figure I-5 shows a typical specification tree. Note that hardware configuration items (HWCI) can have software configuration items subordinate to them. For example a display screen on a mobile phone is dominantly hardware, but it must have embedded software to function. Also the operating system in the mobile phone is dominantly a computer software configuration item (CSCI), but software defines subordinate hardware requirements in order to meet the higher level software requirements for example capacity and speed. In the figure, IRS refers to Interface Requirements Specification.

Developing the specification tree is one element of system design whereby the system is decomposed into its constituent parts. This process has major ramifications on the development of the system in that it essentially determines the items to be purchased versus those to be developed and establishes the framework for the integration and test program. The objective in the design is to achieve the most cost-effective solution to the customer's requirements with all factors considered. Generally, this is achieved by identifying existing or implementation units as early as possible in the tree development. At each element or node of the tree a specification is written, and later on in the project a corresponding individual verification will be performed. When identifying elements, it is useful to consider the element both from a design and a verification perspective. The element should be appropriate from both perspectives.



HWCI	Hardware Configuration Item
CSCI	Computer Software Configuration Item
IS	Interface Specification

Figure I-4. Example Project Specification Tree; also known as a Product Breakdown Structure (PBS)

2. For each specification in the Specification Tree, create an outline using a standard specification template and the definition of the configuration item.

Specification outlines or templates may be obtained from several sources. The most useful and commonly used are previous similar specifications prepared by your organization. These are often the source of useful material, parts of which can be used with minimal modification. In addition, there are Standard formats and IEEE formats recommended for system, hardware, and software specifications.

3. Craft requirements for each specification, fulfilling all flowdown and accommodating derived requirements emerging from the definitions of each configuration item.

A specification represents a design entity and a test entity. The specification should represent appropriate complexity from both the design and the test perspective. Many factors contribute to the appropriate selection of elements. However, as a measure of complexity, a requirements specification should not have too many or too few requirements. As a "rule of thumb" 50 to 250 functional/ performance requirements in a specification is appropriate. Requirements in the physical or environmental areas would be in addition to the functional/ performance variety.

Input

1. System requirements and functional architecture as defined by previous steps.
2. System configuration, with sufficient technical supporting data.

Output

1. Specification Tree.
2. Specifications for each configuration item.

End Result

The result is the Specification Tree and the set of specifications for all of the configuration items that implement the system.

Completion Criteria

1. All specifications identified and located on the Specification Tree.
2. Each specification adequate to proceed with the next stage of development or procurement.

Methods/Techniques

The design and development methods described in the earlier sections apply to this step. As for the actual generation of the Specification Tree and the Specifications, templates and previously completed specifications are useful starting points for document generation.

Metrics

For the Specification Tree:

1. Its completeness as measured by its inclusion of all items required in the system
2. Its balance as determined by its span of control and fan-out from each entity.

For the Specifications,

1. Number of TBDs and TBRs in specifications (goal is zero).
2. Number of requirements in the specification (50 to 250 functional/performance requirements is the ideal range).
3. Stability of the requirements as the development progresses.

Tools

There are a number of commercially available requirement generation and maintenance support tools available. Check the INCOSE website for current tool availability information.

I.6 Requirements and Design Loops

Design is the process of defining, selecting, and describing solutions to requirements in terms of products and processes. A design describes the solution (conceptual, preliminary or detailed) to the requirements of the system.

Synthesis is the translation of input requirements (including performance, function and interface) into possible solutions satisfying those inputs. Synthesis defines a physical architecture of people, product and process solutions for logical groupings of requirements and then designs those solutions. Refer to Appendix K: System Architecture Synthesis for more information on synthesis.

This section describes iterations and feedback (“loops”) between the requirements and design activities. The loop is also an integrated process to refine the requirements.

Stakeholders Participation

The key participants in carrying out the Requirements and Design feedback loops are Systems Engineering, with design, manufacturing, specialty engineering and materials and processes engineering. The function of Systems Engineering is to ensure that the proper inputs and feedback to hardware and software are occurring at the system and lower levels.

The result of the Systems Engineering process in these loops is system and lower level designs that are properly allocated to hardware and software and thoroughly audited to ensure that they meet requirements and are consistent with established manufacturing practices.

Recommended Activities

The following steps describe the Systems Engineering activities in achieving Requirements and Design feedback loops.

1. Determine how SE process is tailored for different levels of the project. This is a Systems Engineering task, and is performed in conjunction with project management. It determines the amount and detail of Systems Engineering to be performed at each level. This should be established early in the project and is covered in the SEP (see Appendix G: Systems Engineering Technical Management).

2. Audit the system requirements. Audits occur at various levels, from peer reviews against requirements in specifications, to design reviews, both informal and formal. The results of the audits serve as feedback to previous SE activities. These results may cause changes in requirements at any level, or may cause design changes.

3. Iterate between systems (hardware and software), design, and manufacturing functions. Systems Engineering should ensure that IPPD engineering is taking place. This may be by chairing a Product Development Team, or by being a member of one. In either case, it is the responsibility of Systems Engineering to ensure that all necessary disciplines in the project are participating in any phase. Systems Engineering consults on all phases of the project to provide the traceability and flow down of the customer's needs and requirements. As necessary, Systems Engineering will conduct producibility meetings (determine production methods and materials) and will conduct producibility trade studies.

4. Audit the design and manufacturing process. After the critical design review (CDR) or the build-to decision gate, Systems Engineering will perform audits on the design (hardware and software) and manufacturing processes to ensure compliance with requirements. Audits occur at various levels, from peer reviews to design reviews, both informal and formal. This provides feedback to the requirements and design functions.

5. Iterate with other parts of the Systems Engineering process. As indicated above, Systems Engineering will ensure that all the elements of the Systems Engineering process are executed.

6. Interface with specialty engineering groups and subcontractors to ensure common understanding across disciplines. This is part of the Systems Engineering role in ensuring that IPPD engineering is being performed on the project.

7. Update models as better data becomes available. Systems Engineering should always ensure that models are up to date.

Input

- Results of Requirements Analysis and Functional Analysis steps; i.e. requirements flowed down to lowest levels.
- Project baseline, proposed changes (initiated by customer or internally from requirements and design analyses, new technology or test results).
- Between project phases, an input to the new phase is the process output from the previous phase. The process outputs include an audit trail of requirements, designs, and decisions.

Output

- New project baseline, which consists of requirements, specifications and designs which comply with requirements.
- A design, which has been audited to ensure compliance with requirements, and which can be produced.

Criteria for Successful Completion

- Successfully establish project baseline
- Completion of requirements audits
- Completion of design audits

Methods/Techniques

Performance of standard configuration management processes will document a baseline that is consistent with the output of the project. Alternatively, create a baseline document, which contains drawings, specifications, published analyses, and deliverable documents that show the current baseline. Also, ensure that all internal and external interfaces and interactions are included.

Appendix J: Functional Analysis and Allocation

Introduction

A *function* is a characteristic task, action, or activity that must be performed to achieve a desired outcome. A function may be accomplished by one or more system elements comprised of equipment (hardware), software, firmware, facilities, personnel, and procedural data.

The scope of the *Functional Analysis/Allocation* activity can be defined by the following:

1) *Functional Analysis/Allocation* is an examination of a defined function to identify all the subfunctions necessary to the accomplishment of that function. The subfunctions are arrayed in a functional architecture to show their relationships and interfaces (internal and external). Upper-level performance requirements are flowed down and allocated to lower-level subfunctions.

2) This activity should be conducted to define and integrate a functional architecture for which system products and processes can be designed. Functional analysis/allocation must be conducted to the level of depth needed to support required synthesis efforts. Identified functional requirements must be analyzed to determine the lower-level functions required to accomplish the parent requirement. All usage modes must be included in the analysis. Functional requirements should be arranged so that lower-level functional requirements are recognized as part of higher-level requirements. Functions should be arranged in their logical sequence; have their input, output, and functional interface (internal and external) requirements defined; and be traceable from beginning to end conditions. Time critical requirements must also be analyzed.

3) The performance requirements should be successively established, from the highest to lowest level, for each functional requirement and interface. Timing requirements that are prerequisite for a function or set of functions must be determined and allocated. The resulting set of requirements should be defined in measurable terms and in sufficient detail for use as design criteria. Performance requirements should be traceable from the lowest level of the current functional architecture, through the analysis by which they were allocated, to the higher-level requirement they are intended to support.

4) Functional analysis/allocation should be conducted iteratively:

- To define successively lower-level functions required to satisfy higher-level functional requirements and to define alternative sets of functional requirements.

- With requirements analysis, to define mission and environment driven performance and to determine that higher-level requirements are satisfied.
- To flow down performance requirements and design constraints.
- With design synthesis, to refine the definition of product and process solutions.

J.1 Purpose of the Functional Analysis/Allocation Task

The objective of *Functional Analysis/Allocation* is to create a functional architecture that can provide the foundation for defining the system architecture through the allocation of functions and subfunctions to hardware/software, databases, facilities and operations (i.e. personnel). It does not describe either the hardware architecture or software architecture of the system. Those architectures are developed during the *System Synthesis* process of the Systems Engineering process (see Appendix K: System Architecture Synthesis).

Functional Analysis/Allocation describes what the system will do, not how it will do it. Every function that must be done by the system in order to meet the operational requirements needs to be identified and defined in terms of allocated functional, performance, and other limiting requirements. Then, each of these functions is decomposed into subfunctions, and the requirements allocated to the function are each decomposed with it. This process is iterated until the system has been completely decomposed into basic subfunctions, and each subfunction at the lowest level is completely, simply, and uniquely defined by its requirements. In the process, the interfaces between each of the functions and subfunctions are fully defined, as are the interfaces to the external world.

Ideally, Functional Analysis/Allocation should begin only after all of the system requirements have been fully identified. This means that the Requirements Analysis must be completed before this task starts. Often, of course, this will not be possible, and these tasks will have to be done iteratively, with the functional architecture being further defined as the system requirements evolve.

Input Criteria

Representative inputs from the user/customer or program management are:

- Functional requirements
- Performance requirements
- Architectural
- Program decision requirements (such as objectives to reuse certain hardware & software, or use Commercial Off The Shelf (COTS) items)
- Specifications and Standards requirements
- Concept of Operations
- Constraints

Output Criteria

There are various formats that the output products of the *Functional Analysis/ Allocation* task can take depending on the specific stage of the process and on the specific technique used to develop the functional architecture. The following are some key diagrams generated from the functional analysis activity:

a. Behavior Diagrams – Behavior Diagrams describe behavior that specifies system-level stimulus responses using constructs that specify time sequences, concurrencies, conditions, synchronization points, state information and performance.

b. Context Diagrams – Top-level diagram of a Data Flow Diagram that is related to a specific level of system decomposition. This diagram portrays all inputs and outputs of a system but shows no decomposition.

c. Control Flow Diagrams – A diagram that depicts the set of all possible sequences in which operations may be performed by a system or a software program. There are several types of Control Flow Diagrams which include Box diagrams, flowcharts, input-process-output (IPO) charts, state transition diagrams.

d. Data Flow Diagrams – they provide an interconnection of each of the behaviors that the system must perform. All inputs to the behavior designator and all outputs that must be generated are identified along with each of the data stores that each must access. Each of the Data Flow diagrams must be checked to verify consistency with the context Diagram or higher level Data Flow Diagram.

e. Data Dictionaries – Documentation that provides a standard set of definitions of data flows, data elements, files, etc. as an aid to communications across the development organizations.

f. Entity Relationship Diagrams – These depict a set of entities (functions or architecture elements) and the logical relationships between them.

g. Functional Flow Block Diagrams (FFBD) – These relate the inputs and outputs and provide some insight into flow between the system functions.

h. Models: Abstractions of relevant characteristics of a system which are used as a means to understand, communicate, design, and evaluate (including simulation) a system. They are used before the system is built and while it is being tested or in service. Section 9: Modeling, Simulation, and Prototyping (page 9.10) states, "Modeling, simulation, and prototyping used during architecture design can significantly reduce the risk of failure in the finished system. These techniques enable the development of complex and costly enabling systems. Systems engineers use modeling and simulation on large complex projects to manage the risk of failure to meet system mission and performance requirements."

i. Simulation Results – The output from a model of the system that behaves or operates like the system under interest when provided a set of controlled inputs.

j. Integrated Definition for Functional Modeling (IDEF) Diagrams – Process control diagrams that show the relationship between functions by sequential input and output flows. Process control enters the top of each represented function and lines entering the bottom show the supporting mechanism needed by the function.

These various output products characterize the functional architecture. There is no one preferred output tool that will support this analysis. In many cases, several of these are necessary to understand the functional architecture and the risks that may be inherent in the subsequent synthesis of system architecture. Using more than one of these formats allows for a "check and balance" of the analysis process and will also aid in the communication across the system design team.

J.2 Major Steps in the Functional Analysis/Allocation Process

Even within a single stage in the system life-cycle, the Functional Analysis/Allocation process is iterative. The functional architecture begins at the top level as a set of functions that are defined in the applicable requirements document or specification, each with functional, performance, and limiting requirements allocated to it (in the extreme, top-level case, the only function is the system, and all requirements are allocated to it). Then, as shown in Figure J-1, the next lower level of the functional architecture is developed and evaluated to determine whether further decomposition is required. If it is, then the process is repeated. If not, then the process is completed and System Synthesis can begin.

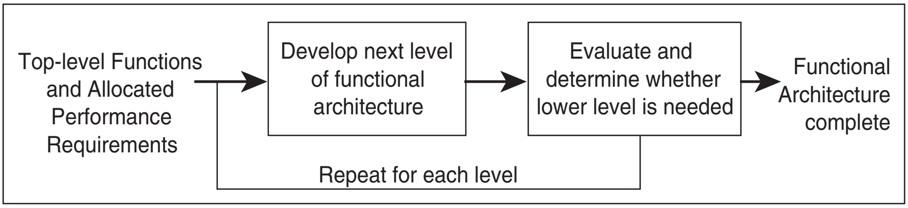


Figure J-1. Functional Analysis/Allocation Process

The Functional Analysis/Allocation process is iterated through a series of levels until a functional architecture is complete. At each level of the *Functional Analysis/ Allocation* process, alternative decompositions and allocations may be considered and evaluated for each function and a single version selected. After all of the functions have been identified, then all the internal and external interfaces to the decomposed subfunctions are established. These steps are shown in Figure J-2.

These steps are each described briefly in the following paragraphs. Note that while performance requirements may be decomposed and allocated at each level of the functional decomposition, it is sometimes necessary to proceed through multiple levels before allocating the performance requirements. Also, sometimes it is necessary to develop alternative candidate functional architectures, and conduct a trade study to determine a preferred one.

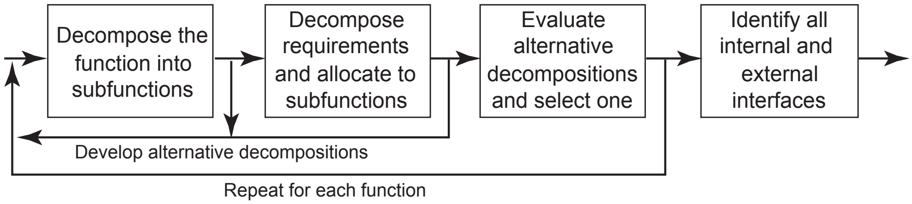


Figure J-2. Alternative Functional Decomposition Evaluation and Definition

With each iteration of Functional Analysis/Allocation, alternative decompositions are evaluated, and all interfaces are defined.

J.2.1 Decompose Each Function to Lower-Level Functions: Functional Flow Block Diagrams

The work of a function is accomplished by one of the system or system elements of equipment, software, facilities, or personnel. Functional identification and decomposition can be performed with respect to logical groupings, time ordering, data flow, control flow, state transitions, or some other criterion. The stepwise decomposition of a system can be viewed as a top-down approach to problem solving.

Objective

The objective of this process is to develop a hierarchy of Functional Flow Block Diagrams (FFBDs) that meet all the functional requirements of the system. Note, however, that this hierarchy is only a portion of the functional architecture. The architecture is not complete until all of the performance and limiting requirements have been appropriately decomposed and allocated to the elements of the hierarchy.

Recommended Activities

For the initial iteration of *Functional Analysis/Allocation*, the baseline requirements and operational concept have been identified during *Requirements Analysis*. First, determine the top-level system functions. This is accomplished by evaluating the total set of baseline requirements as they map to the system-level design, keeping in mind the desire to have highly cohesive, loosely coupled functions. The result is a set of top-level functions which, when grouped together appropriately, provide the required capabilities of each component in the system-level design. Each of the top-level functions is then further refined to lower-level functions based upon its associated requirements.

Decomposition of the function involves the creation of a network of lower-level "child" functions, each of which receives its allocated portion of the "parent's" functional requirements. In this process, each functional requirement is decomposed into lower-level requirements, and each of these is allocated to a lower-level function (i.e. a subfunction) in the next-level FFBD. Functional interfaces fall out of this process.

Develop a description for each function in the hierarchy. This description must include the following:

1. its place in a network (Functional Flow Block Diagram or IDEF0/1) characterizing its interrelationship with the other functions at its level
2. the set of functional requirements that have been allocated to it and which define what it does
3. its inputs and outputs, both internal and external

This process may use various graphical methods to capture the results of the analysis, including structured analysis, such as Data Flow Diagrams, IDEF0/1 diagrams, and Control Flow Diagrams, or other modern techniques. These are all forms of the Functional Descriptions.

"Stop" Criteria

In undertaking the *Functional Analysis/Allocation* process, it is important to establish criteria for completion of the functional decomposition. The usual criteria are to continue until the functional requirement is clear and realizable in hardware, software, and/or manual operations. In some cases, the engineer will continue the effort beyond what is necessary until funding for the activity has been exhausted. In establishing the stop criteria, recognize that the objective of pushing the decomposition to greater detail is to reduce the program risk. At some point, the incremental risk reduction becomes smaller than the cost in time or money of the effort to further decompose. Each program will be different, so it is impossible to set forth all-purpose stop criteria. The program manager and Systems Engineer who understand their specific program's risks need to establish their own stop criteria early in the process and ensure that the decomposition efforts are reviewed frequently.

J.2.2 Allocate Performance and Other Limiting Requirements to All Functional Levels

Requirements allocation is the further decomposition of system-level requirements until a level is reached at which a specific hardware item or software routine can fulfill the needed functional/performance requirements. It is the logical extension of the initial functional identification and an integral part of any functional analysis effort.

Objective

Functional requirements have been fully allocated to functions and subfunctions in the previous step. The objective of this step is to have every performance or limiting requirement allocated to a function or subfunction at the next level in the hierarchy of FFBDs. Some performance requirements will have been decomposed in order to do this. Additional requirements may have to be derived.

Recommended Activities

In this step, performance, and other limiting requirements are allocated to the functions in the next level FFBD. Some straightforward allocation of functional requirements can be made, but the procedure may involve the use of supporting analyses and simulations to allocate system-level requirements. An example of the need for additional analysis is the allocation of availability goals to configuration items. These goals can only be expressed as maintainability and reliability requirements. Allocations are made by these parameters (maintainability and reliability), but only in conjunction with analytical and/or computer simulation to ascertain the impact of a given set of allocations on system availability.

If a requirement cannot be allocated as a single entity, then it must be decomposed and the derived requirements allocated. Often this step requires some anticipation of the results of the *System Architecture Synthesis* because decomposition of response-time or noise-level requirements is equivalent to developing timing or noise budgets. In some cases, it will be necessary to defer decomposition of performance and limiting requirements until multiple stages of functional hierarchy have been developed.

Design Constraint Requirements

Ensure that all constraints are identified to the designer prior to start of detailed design. This should prevent the need for redesign due to unidentified constraints. Include all Systems Engineering groups, but primarily the Engineering Specialties: Reliability, Maintainability, Producibility, Human Engineering, ElectroMagnetic Interference/ ElectroMagnetic Compatibility (EMI/EMC), System Safety, Survivability, Support, Security, Life Cycle Cost/Design-to-Cost.

Design constraints recognize inherent limitations on the sizing and capabilities of the system, its interfacing systems, and its operational and physical environment. These typically include power, weight, propellant, data throughput rates, memory, and other resources within the vehicle or which it processes. These resources must be properly managed to insure mission success.

Design constraints are of paramount importance in the development of derivative systems. A derivative system is a system that by mandate must retain major components of a prior system. For example, an aircraft may be modified to increase its range while retaining its fuselage or some other major components. The constraints must be firmly established: Which components *must* remain unmodified? What can be added? What can be modified? The key principle to be invoked in the development of derivative systems is that the requirements for the system as a whole must be achieved while conforming to the imposed constraints.

Engineering Specialty Constraints

Care must be exercised that the myriad of engineering specialty requirements and constraints are incorporated into appropriate specifications. Incorporation of engineering specialties personnel into the Systems Engineering and Integration Team (SEIT) of an Integrated Product and Process Development (IPPD) organization or into all appropriate Product Development Teams (PDTs), are ways of ensuring that their requirements are incorporated into specifications.

Recommended Activities

1. Identify from the Statement of Work (SOW) all design constraints placed on the program. This particularly includes those from compliance documents.
2. Identify the groups defining constraints and incorporate them into the Systems Engineering effort.
3. Analyze the appropriate standards and lessons learned to derive requirements to be placed on the hardware and software Configuration Item (CI) design.
4. Tailor the compliance documents to fit overall program needs.
5. Identify the cost goals allocated to the design.
6. Define system interfaces and identify or resolve any constraints that they impose.
7. Identify any Commercial Off The Shelf (COTS) or Non-Developmental Item (NDI) CIs that must be used, and the constraints that they may impose.
8. Document all derived requirements in specifications and ensure that they are flowed down to the lowest CI level.
9. Ensure that all related documents (operating procedures, etc.) observe the appropriate constraints.
10. Review the design as it evolves to ensure compliance with documented constraints.

J.2.3 Evaluate Alternative Decompositions and Select One

Not all functional decompositions are of equal merit. It is necessary to consider alternative decompositions at each level, and select the most promising. Because of the reality of system design constraints or target COTS or NDI components, it is often desirable to produce multiple alternative functional architectures that can then be compared in a trade study to pick the one most effective in meeting the objectives.

Objective

Eventually, each sub-function in the lowest levels of the functional architecture is going to be allocated to hardware, software, or manual operations. In addition, each of these functions will have to be tested. The objective here is to select those decompositions that lend themselves to straightforward implementation and testing. Also, we may be able to come up with decompositions that allow a single function to be used at several places within the hierarchy, thereby simplifying development.

Recommended Activities

This is a task that requires sound engineering judgment. There are various ad hoc figures of merit that can be applied to evaluate alternative decompositions. The degree of interconnectivity among functions is one possible measure. There are several measures for software-intensive systems that can be applied, such as high cohesion and low coupling. The Systems Engineer needs to be aware of opportunities for use of NDI hardware and software. That means that a subfunction that has already been implemented in a compatible form on another system may be preferred to one that has not.

J.2.4 Define/Refine Functional Interfaces (internal and external)

All of the internal and external interfaces must be completely defined.

Objective

Each function requires inputs in order to operate. The product of a function is an output. The objective of this step is to identify and document where within the FFBD each function (or subfunction) will obtain its required inputs and where it will send its outputs. The nature of the flows through each interface must be identified.

Recommended Activities

N^2 diagrams can be used to develop interfaces. These apply to systems interfaces, equipment (hardware) interfaces, or software interfaces. Alternatively, or in addition, Data/Control Flow Diagrams can be used to characterize the flow of information among functions and between functions and the outside world. As the system architecture is decomposed to lower and lower levels, it is important to make sure that the interface definitions keep pace, and that interfaces are not defined that ignore lower-level decompositions.

J.2.5 Define/Refine/Integrate Functional Architecture

It may be necessary to make some final modifications to the functional definitions, FFBDs, and interfaces in order to arrive at a viable allocation. The product of this activity is a final FFBD hierarchy with each function (or subfunction) at the lowest possible level uniquely described. The functional flow diagrams, interface definitions, and allocation of requirements to functions and subfunctions constitute the functional architecture.

J.3 Tools Used to Support Functional Analysis/ Allocation

Tools that can be used to perform the four steps in *Functional Analysis/Allocation* include:

- Analysis tools
- Modeling tools
- Prototyping tools
- Simulation tools
- Requirements traceability tools

See the INCOSE web page for a current listing of applicable tools.

J.4 Metrics Used in Functional Analysis/Allocation

This paragraph lists some metrics that can be used to measure the overall process and products of *Functional Analysis/Allocation*. Candidate metrics include the following:

1. Number of allocation-related trade studies completed as a percent of the number identified
2. Percent of analyses completed
3. Number of functions without a requirements allocation
4. Number of functions not decomposed
5. Number of alternative decompositions
6. Number of internal and external interfaces not completely defined
7. Depth of the functional hierarchy as a percentage versus the target depth
8. Percent of performance requirements that have been allocated at the lowest level of the functional hierarchy

Appendix K: System Architecture Synthesis

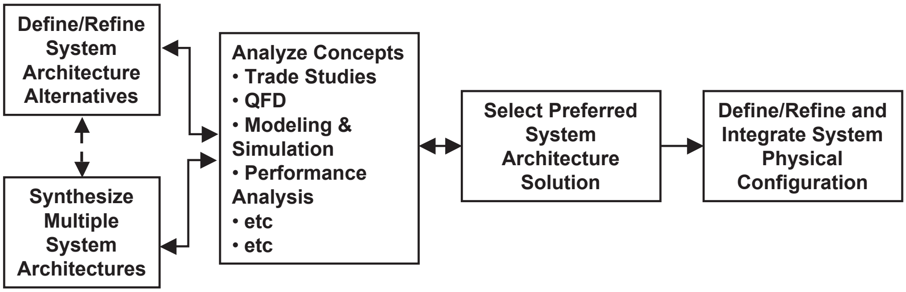
This appendix is complementary to and expands upon material in Section 4.4 Architectural Design Process, and Section 8.2 Architectural Design. This material is also closely tied to Appendix L: Systems Engineering Analyses.

The overall objective of System Definition is to create a System Architecture (defined as the selection of the types of system elements, their characteristics, and their arrangement) that meets the following criteria:

1. Satisfies the requirements (including external interfaces).
2. Implements the functional architecture.
3. Is acceptably close to the true optimum within the constraints of time, budget, available knowledge and skills, and other resources.
4. Is consistent with the technical maturity and acceptable risks of available elements.

System Architecture Synthesis – also known as “analysis of alternatives” – is part of the overall process of system design, which includes Requirements Analysis and Functional Analysis. This process is highly iterative. An initial set of functions is defined to carry out the system’s mission. Requirements quantify how well the functions must be performed, and impose constraints. An architecture is chosen to implement the functions and satisfy the requirements and constraints. The realities of a practical architecture may reveal need for additional functional and performance requirements, corresponding to architecture features necessary for wholeness of the design, but not invoked by the original set of functions. The initial functional and performance requirements may prove infeasible or too costly with any realizable architecture. Consequently, the process involves a mutual adjustment of functions, requirements, and architecture until a compatible set has been discovered.

The process of System Architecture Synthesis flows as shown in Figure K.1. The limitations of text force a sequential description of these functions, although in practice, the process usually proceeds in a highly-interactive, parallel manner with considerable iteration. In addition, for clarity and completeness, a rather formalized description is provided. A large project, with numerous participating organizations at separate locations, may require a high level of formality and discipline for coordination and concurrence, whereas a small unified team can function in a much more informal manner.



QFD = Quality Function Deployment

Figure K.1. System Architecture Synthesis Process Flow

The process of System Architecture Synthesis is essentially a tradeoff, performed at a grand scale, leading to a selected system architecture baseline as the final output. The objective is to select the best from among a set of System Architecture candidates, which have been constructed in a manner that assures (with reasonable certainty) that one of the candidates is acceptably close to the true (usually unknowable and unattainable) optimum.

The process works as follows. A bottom-up approach may be selected, starting with a menu of options for each element of the system (see Section K.1), from which a set of system architecture options is created (see Section K.2). Alternatively, in a top-down approach, a set of system architecture options is created, each providing a framework into which element options may be inserted. These two approaches work together.

K.1 Define/Refine System Element Alternatives

The system elements include the hardware, software, information, procedures, interfaces (including human-machine interactions), and people that make up the system.

The purpose of this activity is to identify element options, one level down from the top of the system hierarchy, which constitutes a set of building blocks from which System Architecture options will be assembled, as described in Section K.2.

The range of element options in the set may be defined by either or both of the following:

a. Expand a range of various types of elements, representing diverse approaches to implementing the system functions (e.g. if the function is communications, the types of elements might include microwave relay, satellite link, or fiber optics)
b. Consider variations of design parameters within any given element type (e.g. number and thrust level of thrust chambers to produce a required total thrust for a launch vehicle).

The objective is to create a set of element options that satisfy the following criteria:

- With reasonable certainty, spans the region of design space which contains the optimum
- Supports analysis which efficiently closes on the optimum
- Contains all relevant design features necessary to provide a firm baseline for the subsequent round of system definition at the next level of detail

A. Participation

This function is lead by Systems Engineering with major support from Hardware and Software Design and Operations. Other engineering support disciplines participate as appropriate to support the definition of system element options, in particular to anticipate issues which will become important later in the system definition process. Within the framework of concurrent engineering, all disciplines are kept informed as the process unfolds. Specific comments may be solicited, and any discipline is free to contribute at any time. The objective is to include all disciplines in identifying design drivers, defining selection criteria, and detecting show-stoppers. In addition, participation in these earlier stages of system definition lays the groundwork for knowledgeable contributions later in the process.

B. Recommended Activities

The following steps, although listed in sequence, are highly interactive and usually evolve in a parallel and iterative manner.

1. Create a list of the elements that will make up the system. These may be derived from the functional requirements or from a decomposition of the existing system architecture. The initial version of this list may be considered preliminary, and will mature as the process is iterated.

2. Identify a set of option descriptors for each element in the list. These are the definitive attributes (design features and parameters) that distinguish one element option from another. The descriptors are the minimal set of significant element characteristics which allows a unique identification for every element choice in the design space.

3. Define the envelope of design space (range of design features and parameter values) which is to be scanned.

4. Develop a process to generate a range of element options, providing both diversity of element types and range of design parameters within a given type. Demonstrate that the range of element options created is both exhaustive and lean:

   - Exhaustive – no good options have been left out and the optimum is somewhere within the envelope of options under consideration (the definition of optimum includes satisfaction of all requirements and factors such as acceptable design maturity, compatibility with the development schedule, minimum cost, acceptable risk, etc.)

- Lean – the number of options to be analyzed is small enough to support efficient selection and closure on the optimum

Borrow from similar existing systems or create new element options through application of the appropriate structured creativity methods (e.g. brainstorming, morphological analysis, synectics, etc. See Adams[1] and other references on structured creativity). Any element previously defined or inferred by the Requirements Analysis and Life Cycle Operations Concept, or Functional Analysis must be included.

5. Generate a set of element options which populates the design space envelope. In general, the options selected should satisfy all requirements, but it is useful to include some which may challenge the requirements in ways leading to a better system concept. This includes relaxing requirements of marginal utility, which are costly to implement, or extending requirements where added capability can be purchased cheaply.

6. Develop the attendant data describing each element option and its interfaces with other elements, as needed to support the selection process and subsequent system definition activity. These data should include estimates for cost, performance, development time, and risk descriptions for each option.

Input

1. Business case for the system, including cost and schedule goals or hard limits.
2. Requirements and Life Cycle Operational Concept, Functional Architecture
3. Technology (available and emerging), and technical constraints
4. Examples of existing systems or elements which perform similar functions.

Output

1. Set of descriptors that define the dimensions of the design space.
2. Set of element options, each characterized by a description of its salient features, parameter values, and interactions with other elements.
3. Documentation of the rationale which justifies the selection of the descriptors, the design space envelope, and the menu of element options.

C. End Result

The result of performing this function is a set of element options with descriptive and supporting documentation that provides:

a. For each option, a description of its salient features, parameter values, and interactions with other elements as necessary to characterize it for analysis and as a potential baseline component. This can take the form of diagrams, schematics, concept drawings, tabular data, and narrative.

b. Identification of design drivers (a limited set of top-level parameters which dominate definition of the design), definition of selection criteria related to elements which will be used in the evaluation process and detection of issues which contain possible show-stoppers.

c. Assurance with reasonable certainty, for the set of options as a whole, that a basis has been established for efficient selection of the optimum architecture, i.e. assuring that the options selected will meet the requirements, that the optimum is somewhere within the range of options to be analyzed, that the optimum can be found quickly and with reasonable certainty, and that the descriptive data (features and parameters) are adequate to support subsequent system definition work.

D. Methods/Techniques

Some useful methods include brainstorming, morphological analysis, synectics, (see Adams[1] and other references on structured creativity), literature search, surveys, inventory of existing concepts, and vendor inquiries.

Metrics

1. Technical performance, schedule spans, costs, and risk estimates for each alternative
2. Evidence that each alternative is consistent with the business case for the system

Tools

Quality Functional Deployment (QFD), Appendix I, Figure I-3, provides a framework to organize the data and test the completeness of the analysis.

K.2 Synthesize Multiple System Architectures

A System Architecture consists of a selection of the types of system elements, their characteristics, and their arrangement. This process uses the set of element options created by the process described above, and the design space of possible System Architecture arrangements of those elements.

The objective is to provide a set of candidate System Architecture options from which the final optimized and robust System Architecture will be selected or will evolve in an efficient manner.

A. Participation

This activity is lead by Systems Engineering with major support from teams responsible for hardware and software design, operations, human factors, producibility, logistics, safety, and others. Other engineering support disciplines participate as appropriate to support the definition of system element characteristics and creation of arrangement options, and in particular, to anticipate issues which will become important later in the system definition process.

B. Recommended Activities

1. Assemble candidate System Architectures.

   a. Examine the System Architecture of existing systems that perform similar functions and adopt, in existing or modified form, any which appear suitable.

   b. Generate system architecture options by combining elements from the set of element options. Utilize the products of Functional Analysis. For each top-level system function, identify a range of means by which it is implemented (choice of implementing element type or design). Build a set of integrated system concepts which incorporate all the element choices. The methodology should rule out absurd or obviously non-optimal combinations of elements, and seek particularly appealing new combinations of the elements. Apply structured creativity methods as appropriate.

2. Verify that the resulting System Architecture options meet the following criteria (see Section 8.10 Verification):

   a. Perform all the functions of the system

   b. Capable of meeting requirements

   c. Satisfies all constraints

   d. Resource usage is within acceptable limits

   e. Elements are compatible

   f. All interfaces are satisfied

3. Ensure in-process validation by involving the customer or user in this process (see Section 8.9 Validation)

4. Screen the set of System Architecture options generated so far, retaining only a reasonable number of the best. Modify the options as necessary to distribute them with reasonable separation throughout the most promising region of the design space. If some promising regions of design space are poorly represented, create more options to fill the void.

Input

1. Menu of elements and combinations.
2. Examples of existing systems which perform similar functions.
3. System Requirements and Functional Architecture.

Output

1. Set of System Architecture options.
2. Documentation of the rationale which justifies the selection of the set of options.

C. End Result

The result is a set of System Architecture options that spans the region of design space containing the optimum, with sufficient descriptive and supporting documentation that provides:

- For each System Architecture option, identification of the elements making up that option, their arrangement, the interactions among the elements, and a description of the salient features and parameter values, as necessary, to characterize the option for analysis and as a potential System Architecture baseline. This can take the form of diagrams, schematics, concept drawings, tabular data, and narrative. Also create the Concept of Operations for each candidate System Architecture (see Section 4.2.4).
- Assurance with reasonable certainty, for the set of candidate System Architecture options as a whole, that the set spans the region of design space which contains the optimum, that the selected set will support efficient selection and closure on the optimum, and that the descriptive data (features and parameters) are adequate to support subsequent system definition work.

D. Methods/Techniques

Some useful methods include brainstorming, morphological analysis, synectics, literature search, surveys, inventory of existing concepts, and vendor inquiries.

Metrics

1. Technical performance, schedule spans, costs, and risk estimates for each alternative
2. Evidence that each alternative is consistent with the business case for the system

Tools

Check the INCOSE website for current references of applicable tools. Quality Functional Deployment (QFD), described in Appendix I, provides a framework to organize the data and test the completeness of the analysis. Others include:

System Hierarchy (functional decomposition)

Functional Flow Block Diagram (FFBD) and Integrated Definition for Functional Modeling (IDEF) diagrams

System Schematic

N^2 Chart

Operational Scenario and System Concept of Operations documents

Kepner-Tregoe Analysis (KTA) and Analytic Hierarchy Process (AHP) models (e.g. Expert Choice)

Decision Trees

K.3 Select Preferred System Architecture/Element Solution

The objective of this process step is to select or evolve the preferred System Architecture from the set of System Architecture options developed in the previous processes steps. The selected baseline System Architecture should be acceptably close to the theoretical optimum in meeting requirements, with acceptable risk, within available resources, and robust, i.e. allows subsequent, more detailed system definition to proceed with minimum backtracking as additional information is uncovered.

A. Participation

This function is conducted by Systems Engineering with support from specialists as necessary to support the definition of selection criteria, and the modeling and analysis used to make the selection.

B. Recommended Activities

The selection of the preferred System Architecture is essentially a tradeoff among the options, using the tradeoff process with modeling. It includes the possibility of combining the best features of several options, and modifying top contenders to further improve their desirability.

1. Define selection criteria and their method of application. The selection criteria are the quantifiable consequences of system implementation and operation. They are derived from the requirements, operational concept, and functions, and from programmatic considerations such as available resources (financial and otherwise), acceptable risk, and political considerations. These selection criteria include:
   a. Measures of the system's ability to fulfill its mission as defined by the requirements
   b. Ability to operate within resource constraints

   c. Accommodation of interfaces
   d. Costs, economic and otherwise, of implementing and operating the system over its entire life cycle
   e. Side effects, both positive and adverse, associated with particular architecture options
   f. Measures of risk
   g. Measures of quality factors
   h. Measures of subjective factors which make the system more or less acceptable to customer, users, or clients, e. g., aesthetic characteristics

2. Create models which map each option's characteristics onto measures of success against the criteria. The models should be as objective and analytical as possible. However, the detail and precision of the models need be sufficient only to clearly distinguish between the options, i.e. the models are used only to produce a clear ranking of the options and not as a design tool.

3. Use the Trade Studies methods described in Appendix L to compare and rank the options.

   Frequently a simple weighted scoring approach, with subjective evaluation of options against the criteria, will be adequate. With this method, the criteria are of two types: go/no-go criteria, which must be met, and criteria used to evaluate the relative desirability of each option on a proportional scale.

   The go/no-go criteria are applied first as an initial screening. Any option that fails any of these criteria is ruled out of further consideration.

4. Modify options or combine the best features of several options to correct shortcomings and advance the capability of the leading contenders. Consider doing a Force Field Analysis and mitigation analysis to help in making decisions. Also, look for adverse consequences and potentially unacceptable risks associated with the top contenders. Either correct such conditions or eliminate options that cannot be corrected.

5. Perform sensitivity analysis to test the robustness of the final selection. Examine the effects of variation in the definitions and application of the criteria, the methods of analyzing and evaluating the options, and any assumptions inherent in the analysis. Look for plausible scenarios that could result in a different selection. If two or more of the options are closely ranked or the ranking can be changed by plausible means, then look for ways to arrive at a clear decision by strengthening the options or improving the selection method, perhaps by expanding the set of criteria.

6. Document the process, providing a clear description of how each step is implemented, justifying all choices made, and stating all assumptions.

Input

1. Requirements and Operational Concept, Functional Architecture, and System Architecture options from Sections K.1 and K.2.
2. Updated business case for the system.
3. Additional information: programmatic, political, economic, etc. needed to define the criteria.
4. Technical information needed to create models and enable evaluation of options.

Output

1. The selected System Architecture baseline
2. Documentation of the selection process to:
   - Justify the selection
   - Enable its review
   - Support subsequent system development, modification and growth throughout its life cycle

C. End Result

The result is a System Architecture baseline, with sufficient descriptive and supporting documentation that provides:

- Identification of the elements (type and principle design characteristics), their arrangement, the interactions among the elements, and a description of the system's salient features and parameter values, as necessary to characterize the System Architecture baseline. This can take the form of diagrams, schematics, concept drawings, operational and life cycle scenarios, tabular data, and narrative.
- Assurance with reasonable certainty, that the selected System Architecture baseline is adequately close to the theoretical optimum, that it is robust, and that the descriptive data (features and parameters) are adequate to support subsequent work.

Metrics

1. Completeness of the selection criteria mentioned in paragraph 1 of the recommended activities (section K.3.B).
2. Completeness of the documentation

D. Methods/Techniques

The general tradeoff methods as described in Appendix L are used. In this case, to economize on the effort expended, the depth of detail and fidelity of modeling is limited to that necessary to clearly separate the options.

Tools

1. Weighted scoring spreadsheet
2. Kepner-Tregoe Analysis (KTA) and Analytic Hierarchy Process (AHP) models, e.g. Expert Choice)
3. Software for Multi-Attribute Utility Analysis
4. Models for converting option parameters to scores against criteria

K.4 Define/Refine/Integrate System Physical Configuration

After the System Architecture has been selected, sufficient detail must be developed on the elements to (1) ensure that they will function as an integrated system within their intended environment, and (2) enable subsequent development or design activity as necessary to fully define each element. During this process step establish the physical, software, and operational implementations, at the next level of detail, for the elements in the selected architecture. Identify the defining interface parameters and, to the degree possible for the current stage of development, define the values of those parameters. The objective of this function is to allow the further definition of each configuration item to proceed on its own, in parallel with all the others.

A. Participation

This function can be lead either by Systems Engineering or by Design Engineering, depending on the technical maturity of the design at the time. In either case, the discipline not in the lead has a strong supporting role.

Systems Engineering provides the process for generating and selecting options for each configuration item, performs analyses and trades, identifies and coordinates interfaces, integrates the results, and ensures that all requirements are implemented.

Design Engineering creates design options for configuration items and their arrangement as a system, develops technical definition data, performs analyses and trades, and documents design decisions.

Supporting disciplines can propose options for configuration items or their features, monitor implementation of requirements in each specialty area, and review the results of the system definition process.

B. Recommended Activities

Note that this function often proceeds in parallel with Define/Derive/Refine Functional/Performance Requirements, as the system development proceeds into a more detailed level of definition.

1. Create a system-level description of system operation, using appropriate tools and notation, to enable a thorough analysis of the system's behavior at the interfaces among all of its elements. Prepare system interface diagrams.
2. Enter the available data about the elements into the system-level description. Obtain interface identification and definition from design engineering and supporting disciplines. Determine what additional data are needed in order to support analysis of system operation.
3. Perform design activity on the elements as needed to provide the additional data identified in Step 2 above.
4. Perform liaison with customer representatives regarding definition of interfaces with the system's operating environment throughout its life cycle.
5. Analyze system operation to verify its compliance with requirements. Modify elements and system architecture, and resolve interface issues as needed to bring the result into compliance.

Input

1. The selected system architecture baseline, configuration items, and interface identifications.
2. Customer's definition of external interfaces.
3. Technical data on interfacing items.

Output

1. Selected design concepts for configuration items to implement all of the system elements, and identification of their interfaces.
2. Documented definition of all interfaces.
3. Documented justification for the selected concepts.

C. End Result

The result is the definition of the set of configuration items (selected technology, configuration, design parameter values, and arrangement) and the definition of their interfaces, integrated as a system.

D. Methods/Techniques

The method is to establish a systematic framework for identifying interfaces and tracking descriptive data, acquiring updates as they occur, and displaying a consistent set of data in a uniform format to concerned parties.

Metrics

1. System requirements not met (if any) by selected concept.
2. Number or percent of system requirements verified by system operation analyses.
3. Number of To Be Determined (TBD) requirements/To Be Resolved (TBR) requirements in system architecture or design.
4. Number of interface issues not resolved.
5. Percent of identified system elements that have been defined.

Tools

- N^2 Chart
- System Schematic
- Interface diagrams
- Tables and drawings of detailed interface data

---

1 Adams, James L, (1990). *Conceptual Blockbusting, 3.ed.* San Francisco Book Company, Inc

THIS PAGE
INTENTIONALLY
LEFT BLANK

Appendix L: Systems Engineering Analyses Activities

The intent of this appendix is to elaborate on Systems Engineering Analysis activities that are only briefly covered or not covered at all in the Chapters 1 to 9 or other appendices.

Chapter 9: Specialty Engineering Activities and Appendix M: Human Systems Integration include the following SE Analysis topics:

- 9.1 Design for Acquisition Logistics and the "-ilities"
- 9.2 Electromagnetic Compatibility Analysis
- 9.3 Environmental Impact Analysis
- 9.4 Human Systems Integration
- 9.5 Mass Properties Engineering Analysis
- 9.6 Modeling, Simulation, and Prototyping
- 9.7 Safety and Health Hazard Analysis
- 9.8 Sustainment Engineering Analysis
- 9.9 Training Needs Analysis

This appendix will focus on supplementing the discussions in Chapters 7, 8, and 9 on the following SE Analysis topics:

- Life Cycle Cost Analysis (augmenting section 8.5.3: Life Cycle Cost Analysis)
- System modeling (augmenting Section 9.6: Modeling, Simulation, and Prototyping)
- Trade Studies (augmenting section 7.1.3: Trade Study and Sensitivity Analysis)

L.1 Life Cycle Cost Analysis

Life Cycle Cost (LCC) analyses are used in system cost/effectiveness assessments, as discussed in Section 8.5.3. The LCC is <u>not</u> necessarily the definitive cost proposal for a program. LCC estimates are often prepared early in a program's life cycle — during Concept Definition. At this stage, there is insufficient detail design information available to support preparation of a realistic, definitive cost analysis. These are much more detailed, and prepared perhaps several years later than the earliest LCC estimates. Later in the program, life cycle LCC estimates can be updated with actual costs from early program phases and should be more definitive and accurate due to hands-on experience with the system.

In addition to providing information for the LCC estimate, these studies also help to identify areas in which emphasis can be placed during the subsequent subphases to obtain the maximum cost reduction.

Adequate documentation requires three basic elements:

1. Data and sources of data on which the estimate is based
2. Estimating methods applied to that data
3. The results of the analysis

The following are recommended activities to perform life cycle cost analysis

1. Obtain a complete definition of the system, elements, and their subsystems.
2. Determine the total number of units of each element, including operational units, prototypes, spares, and test units to be procured. If it is desired to develop parametric cost data as a function of the number of operational units, define the minimum and maximum number of operational units and how, if any, the number of spares and test units will vary with operational unit size.
3. Obtain the life cycle program schedule, including spans for Research and Development (R&D), Production & Deployment, and Operations and Support (O&S) phases. The Production and Deployment phase length will vary with the number of operational units.
4. Obtain manpower estimates for each phase of the entire program and, if possible, for each element and subsystem. These are especially important for cost estimating during R&D and O&S.
5. Obtain approximate/actual overhead, General and Administrative (G&A) burden rates and fees that should be applied to hardware and manpower estimates. Usually this is only necessary for effort within your own company; suppliers will have already factored it into their cost estimates. These data are not required to the accuracy that your finance department would use in preparing a formal cost proposal.
6. Develop cost estimates for each subsystem of each system element for each phase of the program. This is, of course, the critical step. Generally, it should be done as accurately as time and resources allow. Sometimes the argument is heard that the LCC estimates are only to support internal program tradeoff decisions and therefore the estimates must only be accurate enough to support the tradeoffs (relative accuracy), and not necessarily realistic (absolute accuracy). This is usually a bad practice, and if done this is a risk element that should be tracked for some resolution of veracity. The analyst should always **attempt** to prepare accurate cost estimates. These estimates are often reviewed by upper management and customers. It enhances the credibility of results if reviewers sense the costs are “about right,” based on their past experience.
   a. Research and Development (R&D) and Operations and Support (O&S) costs can usually be estimated based on average manpower and schedule spans. Include overhead, General and Administrative (G&A) costs, and fees, as necessary.

b. Investment costs are usually prepared by estimating the cost of the **first** production unit, then applying **learning curve** formulae to determine the reduced costs of subsequent production units. For an item produced with a 90 percent learning curve, each time the production lot size doubles (2, 4, 8, 16, 32, ... etc.) the average cost of units in the lot is 90 percent of the average costs of units in the previous lot. A production cost specialist is usually required to estimate the appropriate learning curve factor(s).
c. R&D and Investment costs can sometimes be scaled by "complexity factors" or Cost Estimating Relationships (CERs) from accurate costs of existing items. This entails fact finding with experts familiar with the item.

The following are example Methods / Techniques for Life Cycle Cost analysts.

Note that some of the methods/techniques below are in addition to those listed in Section 8.5.3 (on page 8.10).

1. Expert Judgment – which is consultation with one or more experts (good for sanity check, but may not be consistent).
2. Analogy – which is reasoning by comparing the proposed project with one or more completed projects that are judged to be similar, with corrections added for known differences (may be acceptable for early estimations).
3. Parkinson Technique – defines work to fit the available resources
4. Price-To-Win – which focuses on providing an approach at or below the price judged necessary to win the contract.
5. Top-Down – which is based on developing costs from the overall characteristics of the project (from the top level of the architecture).
6. Bottom-Up – which identifies and estimates costs for each component and part separately and sums the contributions.
7. Algorithmic (parametric) – which uses mathematical algorithms to produce cost-estimates as a function of cost-driver variables, based on historical data. This technique is supported by commercial tools/models.
8. Design-To-Cost (DTC) or Cost-As-An-Independent-Variable (CAIV) – which works on a design solution that stays within a predetermined set of resources.
9. Wide-band Delphi techniques – estimations from multiple technical and domain experts (estimations only as good as the experts)
10. Taxonomy method – the hierarchical structure or classification scheme for the architecture

L.2 System Modeling

The function of modeling is quickly and economically to create data in the domain of the analyst or reviewer, not available from existing sources, to support decisions in the course of system development, production, testing, or operation. This section augments the discussion in Section 9.6.

A model is a mapping of the system of interest onto a simpler system which approximates the behavior of the system of interest in selected areas. Modeling may be used to represent:

- The system under development
- The environment in which the system operates

Objective

The objective of modeling is to obtain information about the system before significant resources are committed to its design, development, construction, testing, or operation. Consequently, development and operation of the model must consume time and resources not exceeding the value of the information obtained through its use.

Important areas for the use of models include the following:

- Requirements Analysis: determine and assess impacts of candidate requirements
- Architectural Design: evaluate candidate options against selection criteria
- Design & Development: obtain needed design data and adjust parameters for optimization
- Test and Verification: simulate the system's environment and evaluate test data (model uses observable data as inputs for computation of critical parameters that are not directly observable)
- Operations: simulate operations in advance of execution for planning and validation

Criteria for Completion

a. Validation of the model through an appropriate method to the satisfaction of responsible parties
b. Documentation of the model including background, development process, a complete description of the model itself and its validation, and a record of activities and data generated by its use, sufficient to support evaluation of model results and further use of the model
c. Output data delivered to users

B. Recommended Activities for System Modeling

The general steps in application of modeling are:

1. Select the appropriate type(s) of model (Quick, Economical, or Accurate)
2. Design the model
3. Validate the model
4. Obtain needed input data and operate the model to obtain desired output data
5. Evaluate the data to create a recommendation for the decision in question
6. Review the entire process, iterating as necessary to make corrections and improvements
7. Evolve the model as necessary

L.2.1 Selecting the Model Type(s)

Types of Models

As discussed in Section 9.6.2, models fall into one of two general categories – representations and simulations. Representations employ some logical or mathematical rule to convert a set of inputs to corresponding outputs with the same form of dependence as in the represented system, but do not mimic the structure of the system. Simulations, on the other hand, mimic the detailed structure of the simulated system.

The type of model selected depends on the particular characteristics of the system which are of interest. Generally, it focuses on some subset of the total system characteristics such as timing, process behavior, or various performance measures.

Representations and simulations may be made up of one or several of the following types:

- Physical (Wind Tunnel model, Mockups, Acoustic model, structural test model, engineering model, prototypes)
- Graphical (N^2 charts, Behavior diagrams, Program Evaluation Review Technique (PERT) charts, Logic Trees, blueprints)
- Mathematical (deterministic) (Eigenvalue calculations, Dynamic motion, Cost)
- Statistical (Monte Carlo, Process modeling, sequence estimation)

L.2.2 Design of the Model

Care is needed in the design of the model to ensure that the general criteria are met. Usually this requires some degree of fundamental analysis of the system:

1. Identify the relevant system characteristics which are to be evaluated through use of the model.
2. Determine the relevant measurable parameters which define those characteristics, and separate them from irrelevant parameters.
3. Define the scope and content of data needed to support the decision economically and accurately.

It is particularly important that the model be economical in use of time and resources, and that the output data be compact and readily understandable to support efficient decisions. The Taguchi Design of Experiments process (identifying the sensitivity of the results to variation of key parameters and adjusting the spacing of sampling so that the total range of results is spanned with the minimum number of test points) can be very effective in determining the bounds and the limits of the model. This data can be used to estimate the value of the information gained by producing the model.

The model itself can be considered as a system to which the Requirements Analysis, Functional Analysis, and System Synthesis steps of the Systems Engineering Process are applied to determine the requirements for the model and define the approach.

This analysis provides an overall description of the modeling approach. Following its review and approval, the detailed definition of the model can be created according to usual practice for the type of model selected.

L.2.3 Model Validation

It is crucial to prove that the model is trustworthy and suitably represents reality, particularly in cases where a feel for system behavior is absent, or when serious consequences can result from inaccuracy. Models can be validated by:

1. Experience with application of similar models in similar circumstances
2. Analysis showing that the elements of the model are of necessity correct and are correctly integrated
3. Comparison with test cases in the form of independent models of proven validity or actual test data
4. The modeling schema itself can be validated by using small scale models.

L.2.4 Model Application

Obtain needed input data to set the model's parameters to represent the actual system and its operating environment. In some situations, defining and acquiring the basis model data can be a very large effort, so care in design of the model is needed to minimize this problem. Perform as many runs as are needed to span the range of the system parameters and operating conditions to be studied, and in the case of statistical models, to develop the needed level of statistical validity.

L.2.5 Review

Review the entire process to ensure that it supports the conclusion reached. Explore the sensitivity of the result to changes in initial assumptions, data, and processes. If the result is an adequate level of confidence in an unambiguous decision, then the task is complete. Otherwise, look for corrections or improvements to the process and iterate.

L.2.6 Evolution of the Model into a Component of the System

In some cases, a model, created initially to support analysis of the system, evolves to become a deliverable portion of the system. This can occur in cases such as a model of system dynamics which becomes the core of the system control system, or an operations simulation model which evolves into a tool for system operations planning used in the operational phase. The potential for the model to evolve in this manner should be a factor in initial selection and design of the model; anticipation of future uses of the model should be included in its initial conception.

L.3 Trade Studies

Trade studies provide an objective foundation for the selection of one of two or more alternative approaches to solution of an engineering problem. The trade study may address any of a range of problems from the selection of high-level system architecture to the selection of a specific Commercial Off The Shelf (COTS) processor.

In developing a design, it is tempting to select a design solution without performing a formal trade study. The selection may seem obvious to the developer – the other possible alternatives appear unattractive, particularly to other team members (e.g. design, manufacturing, quality, and other "-ility" engineering disciplines). However, it will be far easier to justify the selected solution in a proposal or at a formal design review if we have followed certain procedures in making the selection. Use of a formal trade study procedure will provide discipline in our decision process, and may prevent some ill-advised decisions. It is important, also, to recognize when a formal trade study is not needed in order to reduce project costs.

Whenever a decision is made, a trade-off process is carried out, implicitly, if not explicitly. It is useful to consider trade studies in three levels of formality:

Formal. These trades use a standardized methodology, are formally documented, and reviewed with the customer or internally at a design review.

Informal. These trade studies follow the same kind of methodology, but are only recorded in the engineer's notebook and are not formally reviewed.

Mental. When a selection of any alternative is made, a mental trade study is implicitly performed. The trade study is performed with less rigor and formality than documented trades. These types of trade studies are made continuously in our everyday lives. These are appropriate when the consequences of the selection are not too important; when one alternative clearly outweighs all others; or when time does not permit a more extensive trade. However, if the rationale is not documented, it is soon forgotten and unavailable to those who may follow.

One chooses the level of trade study depending on the consequences to the project, the complexity of the issue, and the resources available. The resources to perform trades are allocated based on the overall life-cycle cost differences (with provision for risk coverage) in alternative selection for the potential trades. Those with the largest overall life-cycle cost deltas are performed first. Since more informal trades can be performed with fewer resources than formal trades, the number and selection of trades and their formality need to be decided with the customer and with the necessary team members who might find some design solutions favorably or unfavorably impacting manufacturability, producibility, reliability, testability, maintainability, etc. Remember, it takes minimal effort to document the rationale for informal and "mental" tradeoff conclusions.

Recommended Activities for Performing Trade Studies

There are multiple techniques for performing trade studies. These include Multi-Attribute Utility Analysis (MAUA), Decision Trees, and Maximum Expected Utility (MEU). There is no need to standardize on any one. One might be better for one trade study; another better in another situation.

The key components of a formal trade study are the following:

1. A list of viable alternative solutions to be evaluated.
2. A list of screening criteria. Any alternative that fails one of these criteria is ruled out of further consideration
3. A list of selection criteria, i.e. a set of factors that characterize what makes a specific alternative desirable. This should include cost, risk, and performance factors.
4. For each of the selection criteria, a metric characterizing how well the various solutions satisfy that criteria.
5. Weighting values assigned to each of the selection criteria, reflecting their relative importance in the selection process.

With these components, an objective measure of the suitability of each alternative as a solution to the problem is obtained. If this process is performed correctly and objectively, then the alternative with the best score is the best overall solution.

L.3.1 Identifying Alternatives

The first step in performing a trade study is the selection of a number of candidate alternative design solutions. In practice, there may be times when as few as only two alternatives need to be considered. However, in general, the trade study should consider between four and seven reasonable alternatives. This will tend to assure that the study will not overlook a viable alternative, while at the same time keeping the cost of the study within reasonable bounds.

It is important that the design solutions being considered be comparable in completeness, i.e. that one can be substituted in our system directly for the other. Where that is not possible, the selection criteria and weighting values need to take into account the disparity.

Do not include alternatives that cannot meet minimum specifications just to expand your trade study. If it cannot meet specification, do not include it. However, if you find that no solution is going to meet your specification, you had better inform higher levels of the problem. Then you might include all viable alternatives, and assign an appropriate metric and weighting value to how close each one comes to meeting the specification. Design alternatives should include those that meet the performance specification, but may be more easily produced, or more reliable, maintainable or supportable.

L.3.2 Determining Selection Criteria

In most cases, there should be no difficulty in determining the selection criteria. There are usually key characteristics that you are looking for in your solution. In almost every trade study, cost and risk are certain to be significant factors. Risk may be decomposed into cost risk, schedule risk, programmatic risk, and performance risk if it appears that these vary separately among the alternatives. If they are not independent, then keep them as a single criterion. Where possible select quantifiable selection criteria; these can be used in decision models.

Make sure that the performance criteria you select are essentially independent. For instance, CPU clock rate and Whetstone performance are closely coupled computer parameters—do not use both. Select only those performance criteria that accurately reflect the needs of your system.

Do not overlook life-cycle cost factors that may be significant to your customer. Manufacturability may be a key factor. Is the solution maintainable? Is it reliable? Will replacement parts be available in the future? Is the software portable to the platforms that will be available in future years? Also, physical parameters such as size, weight, and power consumption could be relevant criteria. Is the solution expandable or scalable? Are design elements or software reusable or already available off-the-shelf?

L.3.3 Assigning Metrics

Assigning metrics to each of the criteria can be very subjective. In order to standardize the interpretation, we will use a scale of one to ten.

One represents total dissatisfaction, while ten represents all we could ever want. The subjective component in assigning metric values arises in determining how to score (i.e. assign values to) various levels of performance. If one processor has a Whetstone rating half of what an ideal one has, do we give it a score of five? Probably not — more likely a one; unless of course, our modeling studies have indicated that half of the ideal is more than adequate for the task at hand, in which case it might even get a ten. The Systems Engineer will have to use their best engineering judgment in assigning scores. It is essential, however, that there be consistency in how the metrics are applied to the various solutions. Two processors with the same Whetstone rating better have the same score for that criterion.

L.3.4 Weighting Values

The weighting values for each criterion distinguish the degree of importance to our design decision. Values should be assigned in the range of one to ten, with ten applying to the most critical criteria for selection. It is important that all parties interested in the decision reach consensus in the assignment of weights. In order to achieve objectivity, this consensus should be reached before the alternative solutions have been scored.

Establishing weighting values can be a difficult task and can become very subjective. For important trades where weightings are particularly difficult to establish, consider using the Analytical Hierarchy Process[1] or Bayesian Team Support[2].

L.3.5 Determine Adverse Consequences

It is important to consider the adverse consequences that may be associated with the leading alternatives. These adverse consequences may have been reflected in the attributes selected; however, to assure that they are all considered, a separate step is appropriate. In many cases, where the risk is considerable, this step corresponds to a risk assessment and may be continually tracked as a risk. In any case, the methodology utilized in performing an adverse consequences analysis is the risk assessment methodology.

L.3.6 Sensitivity Analysis

For the final evaluation and selection, a sensitivity analysis should be performed. A sensitivity analysis is performed to determine if a relative small variation in scoring is affecting the outcome. If the decision is based primarily on scoring of an individual factor, that score needs to be given extra care since it essentially determines the selection. The sensitivity analysis should concentrate on the criteria most affecting winner selection.

L.3.7 Presenting the Results

The results of the formal trade study need to be both presented and explained in a report. A summary presentation would include the following:

- A summary description of each of the alternative solutions
- A summary of the evaluation factors—selection criteria and screening criteria—used
- A summary of the weighting values used, and an explanation of why or how the specific weighting values were selected
- A detailed description of each of the alternative solutions
- A summary description of why or how the specific scores were assigned to each of the alternatives for each of the criteria
- A copy of the spreadsheet (if one was used)
- A graphical display of the overall scores as illustrated above
- A graphical display of the weighted scores for each criterion for each of the alternatives, as shown in the example of Figure L-1.

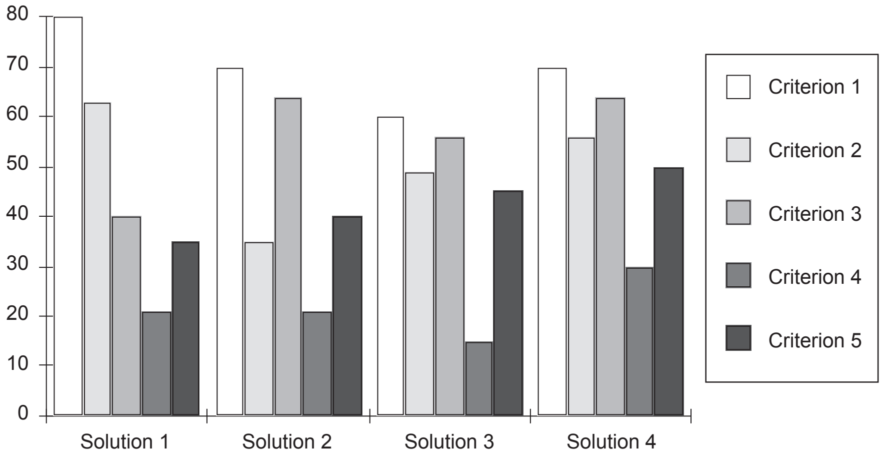


Figure L-1. Weighted Scores For Each Criterion For Each Alternative

1. Scope
2. Tradeoff Study Team Members
   A.
   B. (List Names and Specialties Represented)
   C.
3. Functional and Performance Design Requirements
   A.
   B.
   C.
4. Design Approaches Considered and Significant Characteristics
   A.
   B.
5. Comparison Matrix of the Design Approaches

Feature or Design Requirement	Alternative 1	Alternative 2	Alternative 3	Alternative 4
Requirements 1 (Weight)				
Requirements 2 (Weight)				
Requirements n (Weight)				

6. Design Approach Recommended
   A.
   B.
   C.

Figure L-2. Tradeoff Study Report Format

L.3.8 Preparation of Formal Trade Study Reports

Trade studies provide visibility into the Systems Engineering effort and the reasons for selection of one alternative over another. For the most important trades, a report is prepared and the trade result is presented at a customer design review. An example format for a tradeoff study report is shown in Figure L-2. What follows is a discussion of what information is to be included in each of the paragraphs listed in the figure.

a. Paragraph 1 — State the scope of the report.

b. Paragraph 3 — Identify and list the functional and performance design requirements which are germane to the tradeoff. In each subparagraph, state the functional requirement first and then identify the related performance design requirements. Immediately following each requirement (and in the same paragraph), a reference should be made which identifies the source of the requirement. This reference consists of the title, file name or number, date, page number, and paragraph number from which the requirement statement was extracted.

c. Paragraph 4 — List the possible design approaches and identify the significant characteristics and associated risks of each design approach. Only reasonably attainable design approaches should be discussed in detail, considering technical capabilities, time schedules, resource limitations, and requirement constraints.

Characteristics considered must relate to the attributes of the design approaches bearing most directly on stated requirements. These characteristics should reflect predicted impact on such factors as cost, effectiveness, supportability, personnel selection, training requirements, technical data, schedules, performance, survivability, vulnerability, growth potential, facilities, transportability, and producibility. List the less achievable alternatives with brief statements of why they were not pursued.

d. Paragraph 5 — Present a comparison matrix of design approaches. The purpose of the matrix is to compare the characteristics for each design approach to determine the degree to which the design approaches satisfy the functional and performance design requirements. The objective is to facilitate rapid comparison and evaluation of potential design approaches, and to allow preliminary screening out of those design approaches that are inconsistent with the functional and technical design requirements. Where applicable, include cost-effectiveness models and cost analysis data as enclosures.

e. Paragraph 6 — Recommend the most promising design approach and provide narrative to substantiate the recommendation. Include schematic drawings, outline drawings, interface details, functional diagrams, reliability data, maintainability data, safety data, statistical inference data, and any other documentation or data deemed necessary to support the recommendation. The narrative must cover the requirements that the recommended approach imposes on other areas of the system.

Because there may be a large number of tradeoff study reports prepared during a system development cycle, an index should be prepared which assigns a unique identification number to each tradeoff study report that has been completed.

1 Analytical Hierarchy Process is described in www.expertchoice.com

2 David Ullman, *Making Robust Decisions*, Trafford Publishing, 2007

THIS PAGE
INTENTIONALLY
LEFT BLANK

Appendix M: Human Systems Integration

The intent of this appendix is to elaborate on the Human Systems Integration activities that are described in Section 9.4: Human Systems Integration.

M.1 Definition

Human Systems Integration (HSI) is the interdisciplinary technical and management processes for integrating human considerations within and across all system elements; an essential enabler to systems engineering practice.

M.2 Enabler to Systems Engineering Practice

HSI focuses on the human, an integral element of every system, over the system life cycle. It is an essential enabler to systems engineering practice as it promotes a "total system" approach which includes humans, technology (hardware, software), the operational context and the necessary interfaces between and among the elements to make them all work in harmony. Failure to employ an HSI approach within the systems engineering process may result in failure to meet mission objectives, a poor design, unnecessary burdens on the workers, and in some cases negative impacts to the environment and public health and safety. Without this total system approach, the Enterprise which owns and operates the system may incur unnecessarily high total ownership costs.

An unforgettable example is the 1979 nuclear incident at the Three Mile Island power plant in the United States:

> *"The accident was caused by a combination of personnel error, design deficiencies, and component failures. The problems identified from careful analysis of the events during those days have led to permanent and sweeping changes in how NRC regulates its licensees – which, in turn, has reduced the risk to public health and safety." Further the article states: "it brought about sweeping changes involving emergency response planning, reactor operator training, human factors engineering, radiation protection, and many other areas of nuclear power plant operations." (U.S. Nuclear Regulatory Commission's (NRC) "Fact Sheet on Three Mile Island Accident,"* http://www.nrc.gov/reading-rm/doc-collections/fact-sheets/3mile-isle.html)

In this example, the failure to do HSI in conjunction with comprehensive systems engineering resulted in loss of confidence in nuclear power in the United States and delayed progress in the field for almost 30 years. The clean up costs, legal liability, and significant resources associated with responding to that near catastrophe trace directly to lack of attention to the human element of a highly complex system. Three

Mile Island also emphasizes that, while human performance includes raw efficiency in terms of tasks accomplished per unit time and accuracy versus number of errors made, human performance also involves additional considerations such as intrinsic motivation, the opportunity to learn and improve, and gaining confidence that the human-machine entity can cope with rare, surprising or non-routine events. These considerations can be positively or negatively impacted depending upon how well the hardware and software support the human in meeting system goals.

The "human" in HSI includes all personnel who interact with the system in any capacity:

- system owners
- users/customers
- operators
- maintainers
- support personnel
- trainers, etc.

While many systems engineers intuitively understand that the human operator and maintainer are part of the system under development, they often lack the expertise or information needed to incorporate human capabilities with the capabilities of the hardware and software. HSI assists systems engineers by bringing the various human-centered domains into the SE process and serving as the focal point to ensure that human considerations are integrated into system design, development, manufacturing, operation, sustainment, and disposal. This integration focuses on important customer/user issues:

- usability
- usefulness
- suitability
- effectiveness
- resilience
- understanding of the technological elements
- reliability
- availability
- maintainability
- supportability
- trainability
- ownership cost

The comprehensive application of HSI to a system is intended to optimize total system performance, while accommodating the characteristics of the population that will use, operate, maintain, and support the system; and also support efforts to reduce total ownership costs.

The primary objective of HSI is to integrate the human as a critical system element, regardless of whether humans in the system function as individuals, crews, teams, units or organizations. The technology elements of the system have inherent capabilities. Similarly, humans possess particular knowledge, skills, abilities, expertise, and cultural experiences. HSI seeks to treat humans equally with other system elements, such as hardware and software, in system design. Integration is essential to ensure efficient interfaces between technology elements and the system's intended users, operators, maintainers, and support personnel in operational environments. It is also important to acknowledge that humans outside the system may be affected by its operation. Accordingly, any prescription for SE must accommodate these people as well.

Key methods of integration are trade studies and analyses that address incorporation of human issues. Integrated Product Development Team (IPDT) analyses in which human issues and implications are explored with the users and other engineering disciplines often result in insights not otherwise realized. Trade studies that include human-related issues are critical to optimize total system performance and determine the design that is the most:

- Effective
- Efficient
- Suitable
- Usable
- Affordable

Organizations routinely focus on short term acquisition cost and schedule, while not paying sufficient attention to the more expensive total ownership costs, e.g.

- personnel costs
- repair and sustainment costs
- training costs
- cost of mishaps
- handling hazardous materials
- and disposal costs

Long-term system success and customer satisfaction rely heavily upon demonstrated effectiveness of the total system inclusive of its operators, maintainers, supported customers, sustainers, and the support network. In all systems, failure to address long-term, life cycle issues can result in lost customer confidence, lost market share, product liability, and little repeat business.

The foundation for program success is rooted in requirements development. Front-end analyses are extremely important in generating system requirements. This is no less true for HSI. Effective front-end analyses start with a thorough understanding of the mission of the new system, successes or problems with any predecessor systems, systems with which the proposed system must interface, and the knowledge, skills, abilities, and training associated with the people who are likely to interact with the proposed system. HSI modeling and simulation can also pay large dividends early in the development process, particularly before system hardware and software elements are developed and begin development/certification testing. Supported by this background information, HSI analyses are then used to allocate human-centered functions. Decisions about whether or not to automate certain functions can also be evaluated with modeling and simulation to identify and reduce risk, or at least scope the types and levels of risk involved.

HSI activities must begin with initial capabilities studies followed by continuing analyses (e.g. requirements and capability gap analyses, analyses of alternative solutions, task/function analyses, etc.). With a clearer understanding of the missions, functions, operational scenarios, and tasks that must be supported by the human and technological elements of the system, the systems engineering team can help refine the overall system requirements, inclusive of HSI considerations.

It is critical to include HSI early in concept development and continuously through the development process to realize substantial Life Cycle Cost (LCC) savings. Changes made early in concept refinement, technology development and product development can have major positive impacts compared to later stages of system development where these same changes would be much more difficult and costly with actual hardware and software. Fully utilizing HSI in IPPD helps ensure that systems will not require expensive "train-arounds" or late-phase fixes to address issues of ineffective usability and inefficiencies due to poor human interfaces.

The attributes of communication, coordination, and collaboration are essential to success of the development team. Program managers, chief engineers and systems engineers should ensure that HSI practitioners are actively participating in design reviews, working groups, and IPDTs. Consistent involvement and communications with customers, users, developers, scientists, testers, logisticians, engineers, and designers (human, hardware and software) are essential to successful HSI and good systems engineering practice. Integration takes serious effort, dedicated team participation, and willingness to make effective compromises to benefit the total system.

M.3 Technical and Management Processes

HSI must be addressed by all program level IPTs and at program, technical, design and decision reviews. HSI influences the design and acquisition of developmental systems, non-developmental systems, and system modifications, to include associated support requirements. HSI implementation makes explicit the role that the human plays in system performance and cost, and how these factors are shaped by design decisions.

M.3.1 HSI Domains

The HSI processes facilitate trade-offs among the human-centric domains without replacing individual domain activities, responsibilities, or reporting channels. The human-based domains typically cited by various organizations may differ in what they are called or in number, but the human considerations addressed are quite similar. The human-centered domains with recognized application to HSI include:

M.3.1.1 Manpower. Addresses the number and type of personnel in the various occupational specialties required and potentially available to train, operate, maintain, and support the deployed system. The Manpower community promotes pursuit of engineering designs that optimize the efficient and economic use of manpower, keeping human resource costs at affordable levels. Determination of required Manpower positions must recognize the evolving demands on humans (cognitive, physical, and physiological) and consider the impacts that technology can make on humans integrated into a system. Manpower in HSI is related to but not identical to Human Resources.

M.3.1.2 Personnel. Considers the type of human knowledge, skills, abilities, experience levels, and human aptitudes (i.e. cognitive, physical, and sensory capabilities) required to operate, maintain, and support a system; and the means to provide (recruit and retain) such people. Personnel recruitment, testing, qualification and selection are driven by system requirements. The Personnel community helps define the human performance characteristics of the user population and then determine target populations to select for occupational specialties, manage recruitment, and track retention trends. Personnel must manage occupational specialties to include career progression and assignments. Adequate numbers of workers in these specialties must be recruited, trained, and assigned to meet the entire career field need. Personnel population characteristics can impact manpower and training, as well as drive design requirements. Like Manpower, Personnel is related to Human Resources, but not identical to it.

M.3.1.3 Training. Encompasses the instruction and resources required to provide personnel with requisite knowledge, skills, and abilities to properly operate, maintain, and support systems. The Training community develops and delivers individual and collective qualification training programs, placing emphasis on options that:

- enhance user capabilities
- maintain skill proficiencies (through continuation training and retraining)
- expedite skill attainment
- optimize the use of training resources

Training systems, such as simulators and trainers, should be developed in conjunction with the emerging system. The overall training system may be required prior to fielding the system so that personnel can be adequately trained to operate, maintain, and support the system when it is fielded; therefore, it also is important to develop the training system concurrent with the operational system. If engineering changes are made to the operational system, associated training system changes must be planned and funded.

M.3.1.4 Human Factors Engineering (HFE). Involves understanding and comprehensive integration of human capabilities (cognitive, physical, sensory, and team dynamic) into system design beginning with conceptualization and continuing through system disposal. The primary concern for HFE is creating effective integration of human-system interfaces to achieve optimal total system performance (use, operation, maintenance, support, and sustainment). This "optimal performance" is the achievement of:

- Primary, secondary, backup and emergency tasks and functions
- Goals and objectives, and
- The avoidance of errors in all expected environments

HFE, through comprehensive task analyses (including cognitive), helps define system functions and then allocates those functions to meet system requirements. These efforts should recognize the increasing complexity of technology and the associated demands on people. The design should not demand unavailable or unachievable skills. HFE maximizes usability for the targeted range of users/customers; minimizes design characteristics that induce frequent or critical errors; and strives to eliminate the need for workers to design work-arounds. HFE works with the IPDTs to ensure that representative personnel are tested in situations to determine whether the human can operate, maintain, and support the system in adverse environments, while working under the full range of anticipated mission stress and endurance conditions.

M.3.1.5 Environment. Considers water, air, land, space, cyberspace, markets, organizations and the relationships which exist among them and with all living things and systems. Environmental considerations may affect the concept of operations and requirements to protect systems from the environment and to protect the environment from system design, manufacturing, operations, sustainment, and disposal activities. Refer to SEH Section 9.3 Environmental Impact Analysis for additional information on the Environment.

M.3.1.6 Safety. Promotes system design characteristics and procedures to minimize the potential for accidents or mishaps that: cause death or injury to operators, maintainers, and support personnel; threaten the operation of the system; or cause cascading failures in other systems. Using safety analyses and lessons learned from predecessor systems, the Safety community prompts design features to prevent safety hazards where possible and to manage safety hazards that cannot be avoided. The focus is on designs that have back-up systems, and, where an interface with humans exists, to alert them when problems arise and also help to avoid and recover from errors. Prevalent issues include:

- factors that threaten the safe operation of the system;

- walking and working surfaces;
- pressure extremes;
- control of hazardous energy releases such as
  - mechanical,
  - electrical,
  - fluids under pressure,
  - ionizing or non-ionizing radiation,
  - fire, and
  - explosions

Refer to SEH Section 9.7 Safety and Health Hazard Analysis for additional information on Safety.

M.3.1.7 Occupational Health. Promotes system design features and procedures that serve to minimize the risk of injury, acute or chronic illness, disability, and enhance job performance of personnel who operate, maintain, or support the system. The Occupational Health community prompts design features to prevent health hazards where possible, and recommends personal protective equipment, protective enclosures, or mitigation measures where health hazards cannot be avoided. Prevalent issues include:

- noise,
- chemical exposures,
- atmospheric hazards (e.g. confined space entry and oxygen deficiency),
- vibration,
- ionizing and non-ionizing radiation,
- human factors considerations that can result in chronic disease or discomfort such as repetitive motion injuries or other ergonomic-related problems

Refer to SEH Section 9.7 Safety and Health Hazard Analysis for additional information on Occupational Health.

M.3.1.8 Habitability. Involves characteristics of system living and working conditions such as:

- lighting
- ventilation
- adequate space
- vibration, noise, and temperature control
- availability of medical care, food and/or drink services
- suitable sleeping quarters, sanitation, and personal hygiene facilities

Such characteristics are necessary to sustain high levels of personnel morale, motivation, quality of life, safety, health, and comfort, contributing directly to personnel effectiveness and overall system performance. These habitability characteristics also directly impact personnel recruitment and retention. Some operational/organizational issues may preclude sufficient attention to habitability concerns, hence other HSI domains may need to be worked to mitigate the resulting effects on system personnel and performance.

M.3.1.9 Survivability. Addresses characteristics of a system (e.g. life support, body armor, helmets, plating, egress/ejection equipment, air bags, seat belts, electronic shielding, etc.) that reduce susceptibility of the total system to mission degradation or termination; injury or loss of life; and partial or complete loss of the system or any of its components. These issues must be considered in the context of the full spectrum of anticipated operations and operational environments and for all people who will interact with the system (e.g. users/customers, operators, maintainers, or other support personnel). Adequate protection and escape systems must provide for personnel and system survivability when they are threatened with harm. Refer to SEH Section 9.1.2 for additional information on survivability analysis.

These individual, but very interdependent, domains must be considered simultaneously because decisions made in one domain can have significant impacts on other domains. Each individual domain decision generates the need to concurrently assess HSI issues across all the domains and against mission performance, prior to making formal programmatic decisions. This approach mitigates the potential for unintended, adverse consequences, including increased technical risk and cost.

The Systems Engineer plays a critical role in engaging HSI expertise to support the integrated design teams, ensuring that human performance and interface considerations are properly identified and developed in the design and that sound functional allocations are made. A knowledgeable, interdisciplinary HSI team is generally required to address the full spectrum of human considerations.

M.3.2 Key HSI Tenets

HSI programs have distilled HSI activities into the following key actionable tenets:

M.3.2.1 Initiate HSI Early

- HSI should begin prior to program initiation with front-end analyses and requirements definition. HSI-related requirements include not just those of the individual domains, but also those that arise from interactions among the HSI domains.
- HSI requirements must be developed in consonance with other system requirements, considering any constraints. The human considerations identified in the requirements must address the capabilities of users, operators, maintainers, and other personnel as they interact with and within the system. HSI requirements must be reconsidered, refined, and revised, as program documents, system requirements, and specifications are updated. Doing this early, continuously and comprehensively as part of the systems engineering process provides the opportunity to identify risks and costs associated with program decisions.

- HSI must be included in the initial Acquisition Strategy, Systems Engineering Plan (SEP), and Life Cycle Cost (LCC) documents.

M.3.2.2 Identify Issues and Plan Analysis

- The project manager must have a comprehensive plan for HSI early in the acquisition process and summarize HSI planning in the Acquisition Strategy. Project managers should address HSI throughout the entire acquisition cycle as part of the Systems Engineering process.
- Planning/re-planning are the cornerstones of HSI efforts, ensuring human considerations are effectively integrated into system development and acquisition. HSI plans outline criteria for success and how to determine when a change may be required.
- HSI planning should be expressed in program documentation either as a stand-alone plan or integrated within the program's Systems Engineering Plan (SEP), or comparable planning document. If the HSI plan is a stand-alone document, then the SEP, or comparable document must contain a reference to it.
- HSI plan information should include the details associated with the analyses to support HSI tradeoffs, such as goals of the analyses and required timeline and resources.

M.3.2.3 Document/Crosswalk HSI Requirements

- Derive HSI requirements as needed at each level of the system hierarchy, in accordance with the processes described in Appendix I. Use HSI plans, analyses, and reports as sources (or rationale) for the derived requirements.
- Cross-reference HSI requirements with other documents, plans, and reports and document them in the requirements traceability documents.

M.3.2.4 Make HSI a Factor in Source Selection for Contracted Development Efforts

- HSI requirements need to be an explicit part of source selection planning and implementation with an appropriate level of importance assigned to them.
- HSI needs to be treated as a distinct major managerial and technical area during source selection.
- Solicitations must require offerors to respond to all pertinent HSI considerations in the Statement of Work (SOW).
- As a minimum, HSI requirements should be included in the SOW; descriptions of deliveries and performance requirements; Instructions to Offerors; and Evaluation Factors for award.
- The source selection evaluation board should include an HSI representative(s) capable of adequately representing the HSI domains in source selection scoring.

M.3.2.5 Execute Integrated Technical Processes

- Domain integration begins prior to program initiation with front-end analyses and requirements definition and continues through development, operations, sustainment, modification, and all the way to eventual system disposal.
- HSI activities/considerations must be included in each key project planning document (e.g. Acquisition Strategy, SEP, test plan, etc.), not just the HSI planning document.
- Systems Engineers and HSI personnel must be prepared to present accurate, integrated cost data to demonstrate reduced total ownership costs, thereby justifying trade-off decisions that may increase design and acquisition costs.
- Program tools/processes should facilitate exchange of technical data between HSI activities, engineering/design disciplines, and the overall program.
- Development of an HSI Integrated Architecture, which details the integrated technical HSI processes (e.g. mission task analyses, function allocation, hazards assessment), makes the HSI process and its products (e.g. Manpower Estimate, Training System Plan, HSI Plan, etc.) clear for engineering team members and project managers.

M.3.2.6 Conduct Proactive Tradeoffs

- Often tradeoff decisions are made rapidly, so HSI representatives must be available and ready with information to influence HSI-related decisions. HSI-related tools are available to support these tradeoffs (such as IMPRINT, available at http://www.manprint.army.mil/; and tools found in the Directory of Design Support Methods, http://www.dtic.mil/dticasd/ddsm).
- The development of HSI objectives and thresholds in the requirements documents are critical to subsequent HSI tradeoff analyses.
- In conducting tradeoff analyses both within HSI domains and for the system as a whole, the primary goal is to ensure human performance meets or exceeds the performance requirements for the total system without compromising survivability, environment, safety, occupational health, and habitability.

M.3.2.7 Conduct HSI Assessments

- The purpose of the HSI assessment process is to evaluate the application of HSI principles throughout the system life cycle. The process enables cross-corporate and cross-discipline HSI collaboration in acquisition program evaluations, and provides additional avenues for issue resolution.
- Programs should address HSI assessments during related IPDT meetings, Working Group meetings, System Engineering Reviews, Design Reviews, Logistical Assessments, verification, and validation.
- Responses to HSI assessments should be based on sound analysis and reported findings. Deficiencies should be documented and include a detailed description

of the deficiency, its operational impact, recommended corrective action, and current status.

M.4 Additional Information

Included below are some key web sites that will provide methods, tools, processes, referrals, and more detailed references to accomplish HSI in Systems Engineering:

http://www.connect.incose.org/tb/specialty/hsi/default.aspx – INCOSE Connect

https://www.afkm.wpafb.mil/ASPs/CoP/EntryCoP/asp?Filter=HP-HS – Joint HSI Community of Practice

http://hsiport.resourceconsultants.com/register.aspx – HSI Tools Port: Program Online Tools Support

http://www.dtic.mil/dticasd/ – DoD DTIC Information Center: Support Methods and Resources

M.5 Suggested References

Bias, R.G. and D.J. Mayhew, *Cost Justifying Usability*, Boston, MA. Academic Press, 1994.

Blanchard and Fabrycky, *Systems Engineering and Analysis*, Fourth Edition, N.J.: Pearson Prentice Hall, 2006.

Booher, H. R. (Ed.), *Handbook of Human Systems Integration*, NY: John Wiley & Sons, Inc., 2003.

Chapanis, A., *Human Factors in Systems Engineering*, NY: John Wiley & Sons, Inc., 1996.

Rouse, W.B., Design for Success: *A Human-Centered Approach to Designing Successful Products and Systems*, New York: John Wiley & Sons, Inc., 1991.

THIS PAGE
INTENTIONALLY
LEFT BLANK

Appendix N: System Integration

The System Integration (SI) function establishes system internal interfaces and interfaces between the system and larger program(s). The SI function includes the integration and assembly of the system with emphasis on risk management and continuing verification of all external and internal interfaces (physical, functional and logical).

At the top level, System Integration is performed on the system and its elements, and on the system and interfacing external systems. The objective is to ensure that elements are integrated into the system and that the system is fully integrated into the larger program. The System Integration function may also be involved in the integration of subsystems into their elements. At lower levels of the system hierarchy, Product Integration Teams and Product Development Teams perform integration.

A discussion of these activities is divided into the internal interfaces among the components comprising the system, entitled System Build; and the external interfaces between the system and other systems, entitled System Integration with External Systems.

N.1 System Build

This process addresses the System Integration internal to the system — i.e. the integration of all the components comprising the system. System build is bottom-up. That is, components at the bottom of the system hierarchy are integrated and tested first.

Recommended activities

1. Obtain the system hierarchy. The system hierarchy shows the relationship between the system segments and elements, which are structured functionally to form the system. The process begins with a good knowledge of this system structure. In addition to the system hierarchy, obtain the system and Configuration Item (CI) design specifications, functional block diagrams, N2 charts, and any other data which defines the system structure and its interfaces.

2. Determine the interfacing subsystems and components.

3. Ascertain the functional and physical interfaces of the system, elements, subsystems, assemblies, and components within the system. This will require a detailed assessment of the functions flowing in both directions across the interfaces, such as data, commands and power. It will also require a detailed assessment of the physical interfaces such as fluids, heat, mechanical attachments and footprints, connectors and loads.

4. Organize Interface Control Document(s) or drawing(s) to document the interfaces and to provide a basis for negotiation of the interfaces between/among the parties to the interfaces.
5. Work with producibility/manufacturing groups to verify functional and physical internal interfaces, and to ensure changes are incorporated into the specifications.
6. Conduct internal Interface Working Groups (IFWGs) as required. These would involve all the relevant engineering disciplines. There may be a series of subgroups by discipline, or one group, depending on the size and complexity of the system.
7. Review test procedures and plans which verify the interfaces.
8. Audit design interfaces.
9. Ensure that interface changes are incorporated into specifications.

N.2 System Integration with External Systems

This process addresses the System Integration external to the system – i.e. the integration of all the system under development with interfacing external systems.

Recommended activities

1. Obtain system hierarchy, and the systems and Configuration Item design specifications, functional block diagrams, N-squared charts and any other data which defines the system structure and its interfaces.
2. Determine the interfacing systems by reviewing the items in step 1 above.
3. Obtain interfacing programs' Interface Control Documents (ICDs), System Engineering Plans (SEPs) and relevant interface documents.
4. Ascertain the functional and physical interfaces of the external systems with the subject system. This will require a detailed assessment of the functions flowing in both directions across the interface, such as data, commands and power. It will also require a detailed assessment of the physical interfaces such as fluids, heat, mechanical attachments and footprints, connectors and loads.
5. Organize an Interface Control Document (ICD) to document the interfaces and to provide a basis for negotiation of the interfaces between/among the parties to the interfaces.
6. Conduct Interface Working Groups (IFWGs) among the parties to the interfaces. These can be one group covering all interfaces for a smaller program, or it can be broken into engineering disciplines addressing the interfaces for larger programs.

The ICD is developed over a series of meetings/teleconferences in which the representatives of each side of the interface directly present the performance or needs for their side of the interface. One party takes the lead to be the author of the ICD, and to ensure that copies are available to other parties before a meeting. All parties sign the ICD when agreement has been reached. After the document is signed it is released and comes under formal change control.

7. Review test procedures and plans which verify the interfaces.
8. Audit design interfaces.
9. Incorporate interface changes into specifications.

Metrics

Percentage of released interface drawings.
Percentage of completed ICDs.
Percentage of approved ICDs.
Number and type of interface issues resolved and unresolved.

Methods/Techniques

Performance of standard configuration management processes will document a concurrent baseline that is consistent with the output of the program. Alternatively, create a baseline document, which contains drawings, specifications, published analyses, and deliverable documents which show the current baseline.

Additionally, ensure that all internal and external interfaces and interactions are included. Interface Working Groups (IFWG) are established to review interface statements/drawings, and are a good means of ensuring direct interaction of all parties to the interface, as discussed above.

Tools

N^2 charts
Functional block diagrams
Interface Working Group (IFWG)
Traceability database

THIS PAGE
INTENTIONALLY
LEFT BLANK

Appendix O: Comment Form

Reviewed document:	Insert Document Title
Name of submitter (first name & last name):	John Doe III
Date Submitted:	21-Aug-2010
Contact info (email address):	john.doe@anywhere.com
Type of submission (individual/group):	group
Group name and number of contributors (if applicable)	INCOSE XYZ WG

(See **SAMPLE FORM** on following two pages ➡)

Submit comments to TBD Working Group chair. Current WG chair will be listed at:
http://www.incose.org/techcomm.html

If this fails, comments may be sent to:
info@incose.org (the INCOSE main office), which can relay to the appropriate WG, if so requested in the comment cover page.

					Please read examples carefully before providing your comments (and delete the examples provided).		
Com-menter's Name	Comment Sequence Number	Cate-gory (TH, TL, E, G)	Section Number (e.g. 3.4.2.1, no alpha)	Specific Reference (e.g., Para-graph, Line, Figure, Table	Issue, comment and rationale *(rationale must make comment clearly evident and supportable)*	Proposed Changed/New Text *– MANDATORY ENTRY – (must be substantial to increase the odds of acceptance)*	Importance Rating (R=Required, I = Important, T=Think About for future version
John Doe III	1	E	6.3.2	Paragraph three	Is the inclusion of the spiral model in the incremental life cycle stray text? The spiral model is more often associated with the evolutionary model (6.3.3)	(delete third paragraph)	
John Doe III	2	TH	A5.2.e	first line	Find a different term for reviewing requirements to assure goodness: this is not requirements validation. Call the activity review, or ?	Each technical requirement statement should be reviewed to ensure that it exhibits the following quality attributes:	R

					Please read examples carefully before providing your comments (and delete the examples provided).		
John Doe III	4	TH	A.5.5		This section wants validation to be completed before integration. Usually validation is completed after integration. If this is written as intended, then more amplification is needed to clarify why validation should precede integration. These sound like they are notes for A.5.8; the validation notes section, and belong in that section. This section should address some notes tied directly to integration. (See the SAE TBD WG or INCOSE's Jane Smith for some further thoughts on integration.)	This section wants validation to be completed before integration. Usually validation is completed after integration. If this is written as intended, then more amplification is needed to clarify why validation should precede integration. These sound like they are notes for A.5.8; the validation notes section, and belong in that section. This section should address some notes tied directly to integration. (See the SAE TBD WG or INCOSE's Jane Smith for some further thoughts on integration.)	R

THIS PAGE
INTENTIONALLY
LEFT BLANK