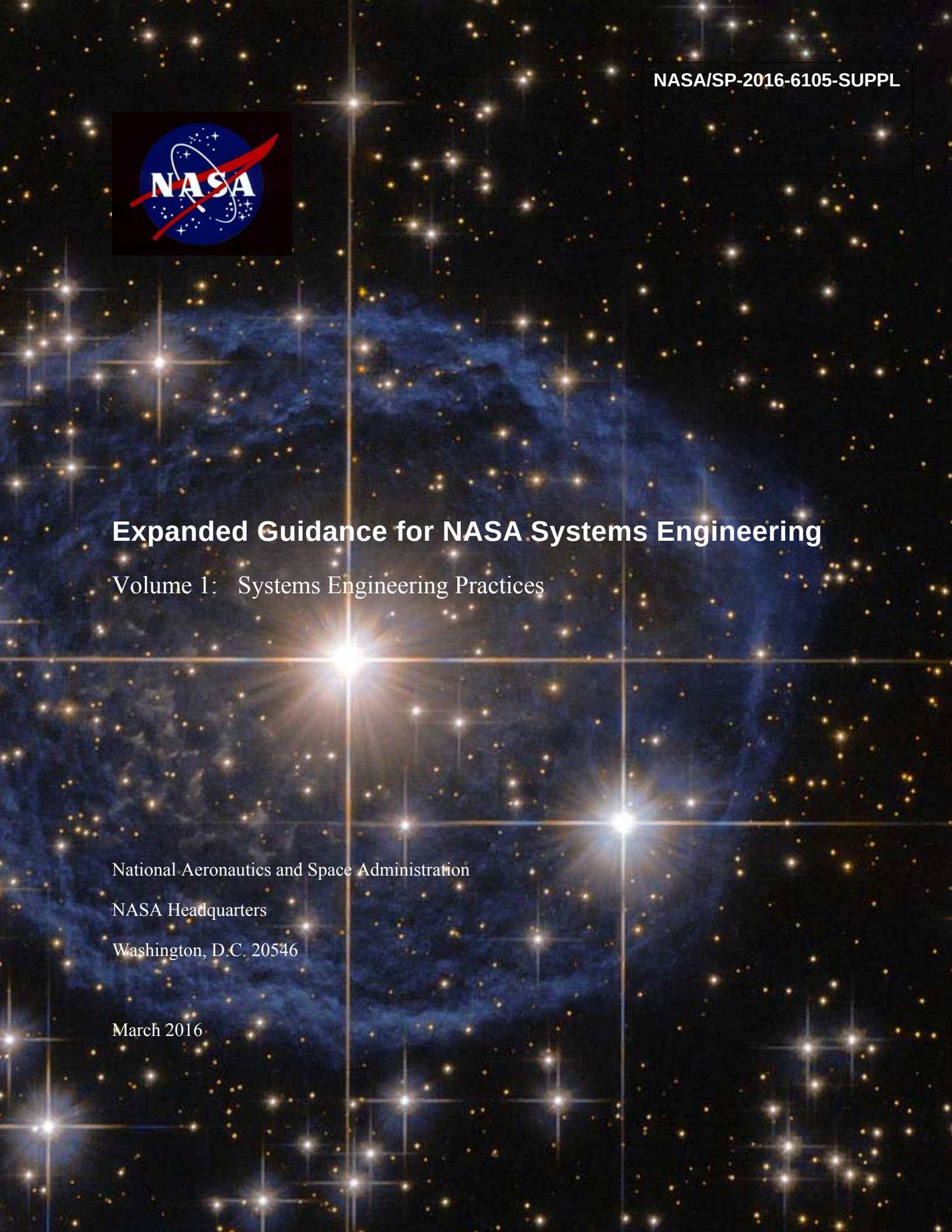


NASA/SP-2016-6105-SUPPL

Expanded Guidance for NASA Systems Engineering

Volume 1: Systems Engineering Practices

National Aeronautics and Space Administration

NASA Headquarters

Washington, D.C. 20546

March 2016

Part 1 Table of Contents

Part 1 Table of Figures

Part 1 Table of Tables

Part 1 Table of Blue Boxes

Preface

Since the initial writing of NASA/SP-6105 in 1995 and the following revision (Rev 1) in 2007, systems engineering as a discipline at the National Aeronautics and Space Administration (NASA) has undergone rapid and continued evolution. Changes include implementing standards in the International Organization for Standardization (ISO) 9000, using Model-Based Systems Engineering to improve development and delivery of products, and accommodating updates to NASA Procedural Requirements (NPR) 7123.1. Lessons learned on systems engineering were documented in reports such as those by the NASA Integrated Action Team (NIAT), the Columbia Accident Investigation Board (CAIB), and the follow-on Diaz Report. Other lessons learned were garnered from the robotic missions such as Genesis and the Mars Reconnaissance Orbiter as well as from mishaps from ground operations and the commercial spaceflight industry. Out of these reports came the NASA Office of the Chief Engineer (OCE) initiative to improve the overall Agency systems engineering infrastructure and capability for the efficient and effective engineering of NASA systems, to produce quality products, and to achieve mission success.

In 1995, SP-6105 was initially published to bring the fundamental concepts and techniques of systems engineering to NASA personnel in a way that recognized the nature of NASA systems and the NASA environment. In 2007, Rev 1 of the handbook was finalized and distributed. While updating the 2007 rev 1 version of the NASA Systems Engineering Handbook for a new Rev 2 version, authors from across the Agency submitted a wealth of information that not only expanded on the content of the earlier version but also added entire new sections. This body of knowledge has been captured in this document as “Expanded Guidance for NASA System Engineering,” presented in a two-volume set. The over 700 pages of information is considered a relevant reference to the larger NASA Systems Engineering community of practitioners.

The official second revision of the NASA Systems Engineering Handbook filters some of this information that is ancillary to implementing NPR 7123.1 for the purpose of condensing the information into a more manageable version useable as a handbook. The official revised NASA Systems Engineering Handbook is a more focused “core” version of the information in this expanded guidance document.

This expanded guidance continues the methodology of the SE Handbook: a top-down compatibility with higher level Agency policy and a bottom-up infusion of guidance from the NASA practitioners in the field. This approach provides the opportunity to obtain best practices from across NASA and bridge the information to the established NASA systems engineering processes and to communicate principles of good practice as well as alternative approaches rather than specify a particular way to accomplish a task. The result embodied in this handbook is a top-level implementation approach on the practice of systems engineering unique to NASA. Material used for updating this handbook has been drawn from many sources, including NPRs, Center systems engineering handbooks and processes, other Agency best practices, and external systems engineering textbooks and guides.

This expanded guidance consists of eight chapters: (1) an introduction, (2) a systems engineering fundamentals discussion, (3) the NASA program/project life cycles, (4) systems engineering processes to get from a concept to a design, (5) systems engineering processes to get from a design to a final product, (6) crosscutting management processes in systems engineering, (7)

crosscutting topics, and (8) special topics related to systems engineering that are not yet considered established best practices within the Agency but are included as reference and source material for practitioners. The chapters are supplemented by appendices that provide outlines, examples, and further information to illustrate topics in the chapters. This expanded guidance makes extensive use of boxes and figures to define, refine, illustrate, and extend concepts in the chapters.

Finally, it should be noted that this document provides top-level guidance for good systems engineering practices; it is not intended in any way to be a directive.

Acknowledgments

The following individuals are recognized as contributing practitioners to the content of this expanded guidance:

Alexander, Michael, NASA/Langley Research Center

Allen, Martha, NASA, Marshall Space Flight Center

Baumann, Ethan, NASA/Armstrong Flight Research Center

Bixby, CJ, NASA/Armstrong Flight Research Center

Boland, Brian, NASA/Langley Research Center

Brady, Timothy, NASA/NASA Engineering and Safety Center

Bromley, Linda, NASA/Headquarters/Bromley SE Consulting

Brown, Mark, NASA/Jet Propulsion Laboratory

Brumfield, Mark, NASA/Goddard Space Flight Center

Campbell, Paul, NASA/Johnson Space Center

Carek, David, NASA/Glenn Research Center

Cox, Renee, NASA/Marshall Space Flight Center

Crable, Vicki, NASA/Glenn Research Center

Crocker, Alan, NASA/Ames Research Center

DeLoof, Richard, NASA/Glenn Research Center

Demo, Andrew/ Ames Research Center

Dezfuli, Homayoon, NASA/HQ

Diehl, Roger, NASA/Jet Propulsion Laboratory

DiPietro, David, NASA/Goddard Space Flight Center

Doehne, Thomas, NASA/Glenn Research Center

Duarte, Alberto, NASA/Marshall Space Flight Center

Durham, David, NASA/Jet Propulsion Laboratory

Epps, Amy, NASA/Marshall Space Flight Center

Fashimpaur, Karen, Vantage Partners

Feikema, Douglas, NASA/Glenn Research Center

Fitts, David, NASA/Johnson Space Flight Center

Foster, Michele, NASA/Marshall Space Flight Center

Fuller, David, NASA/Glenn Research Center

Gati, Frank, NASA/ Glenn Research Center

Gefert, Leon, NASA/Glenn Research Center

Ghassemieh, Shakib, NASA/Ames Research Center

Grantier, Julie, NASA/ Glenn Research Center

Hack, Kurt, NASA/Glenn Research Center

Hall, Kelly, NASA/Glenn Research Center

Hamaker, Franci, NASA/Kennedy Space Center

Hange, Craig, NASA/Ames Research Center

Henry, Thad, NASA/Marshall Space Flight Center

Hill, Nancy, NASA/Marshall Space Flight Center

Hirshorn, Steven, NASA/Headquarters

Holladay, Jon, NASA/NASA Engineering and Safety Center

Hyatt, Mark, NASA/Glenn Research Center

Killebrew, Jana, NASA/Ames Research Center

Jannette, Tony, NASA/Glenn Research Center

Jenks, Kenneth, NASA/Johnson Space Center

Jones, Melissa, NASA/Jet Propulsion Laboratory

Jones, Ross, NASA/Jet Propulsion Laboratory

Killebrew, Jana, NASA/Ames Research Center

Leitner, Jesse, NASA/Goddard Space Flight Center

Lin, Chi, NASA/Jet Propulsion Laboratory

Mascia, Anne Marie, Graphic Artist

McKay, Terri, NASA/Marshall Space Flight Center

McNelis, Nancy, NASA/Glenn Research Center

Mendoza, Donald, NASA/Ames Research Center

Miller, Scott, NASA/Ames Research Center

Montgomery, Patty, NASA/Marshall Space Flight

Mosier, Gary, NASA/Goddard Space Flight Center

Noble, Lee, NASA/Langley Research Center

Oleson, Steven, NASA/Glenn Research Center

Parrott, Edith, NASA/Glenn Research Center

Powell, Christine, NASA/Stennis Spaceflight Center

Powell, Joseph, NASA/Glenn Research Center

Price, James, NASA/Langley Research Center

Rawlin, Adam, NASA/Johnson Space Center

Rochlis-Zumbado, Jennifer, NASA/Johnson Space Center

Rohn, Dennis, NASA/Glenn Research Center

Rosenbaum, Nancy, NASA/Goddard Space Flight Center

Ryan, Victoria, NASA/Jet Propulsion Laboratory

Sadler, Gerald, NASA/Glenn Research Center

Salazar, George, NASA/Johnson Space Center

Sanchez, Hugo, NASA/Ames Research Center

Schuyler, Joseph, NASA/Stennis Space Center

Sheehe, Charles, NASA/Glenn Research Center

Shepherd, Christena, NASA/Marshall Space Flight Center

Shull, Thomas, NASA/Langley Research Center

Singer, Bart, NASA/Langley Research Center

Slywczak, Richard, NASA/Glenn Research Center

Smith, Scott, NASA/Goddard Space Flight Center

Smith, Joseph, NASA/Headquarters

Sprague, George, NASA/Jet Propulsion Laboratory

Trase, Kathryn, NASA/Glenn Research Center

Trenkle, Timothy, NASA/Goddard Space Flight Center

Vipavetz, Kevin, NASA/Langley Research Center

Voss, Linda, Dell Services

Walters, James Britton, NASA/Johnson Space Center

Watson, Michael, NASA/Marshall Space Flight Center

Weiland, Karen, NASA/Glenn Research Center

Wiedeman, Grace, Dell Services

Wiedenmannott, Ulrich, NASA/Glenn Research Center

Witt, Elton, NASA/Johnson Space Center

Woytach, Jeffrey, NASA/Glenn Research Center

Wright, Michael, NASA/Marshall Space Flight Center

Yu, Henry, NASA/Kennedy Space Center

1.0 Introduction

1.1 Purpose

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

This expanded guidance should be used as a companion for implementing NPR 7123.1, Systems Engineering Processes and Requirements, the Rev 2 version of SP-6105, and the Center-specific handbooks and directives developed for implementing systems engineering at NASA. It provides a companion reference book for the various systems engineering-related training being offered under NASA's auspices.

1.2 Scope and Depth

The coverage in this guide is limited to general concepts and generic descriptions of processes, tools, and techniques. It provides information on systems engineering best practices and pitfalls to avoid. There are many Center-specific handbooks and directives as well as textbooks that can be consulted for in-depth tutorials. References to "documents" is intended to include not only paper or digital files, but also models, graphics, drawings, or other appropriate means to capture the intended information.

This guide describes systems engineering best practices that should be incorporated in the development and implementation of large and small NASA programs and projects. The engineering of NASA systems requires a systematic and disciplined set of processes that are applied recursively and iteratively for the design, development, operation, maintenance, and closeout of systems throughout the life cycle of the programs and projects.

The scope of this guide includes systems engineering functions regardless of whether they are performed by a manager or an engineer, in-house or by a contractor. This guide is applicable to NASA space flight projects of all sizes and to research and development programs and projects. While all 17 processes are applicable to all projects, the amount of formality, depth of documentation, and timescales are varied as appropriate for the type, size, and complexity of the project. References to "documents" are intended to include not only paper or digital files but also models, graphics, drawings, or other appropriate forms that capture the intended information.

There are many Center-specific handbooks and directives as well as textbooks that can be consulted for in-depth tutorials. For guidance on systems engineering for information technology projects, refer to Office of Chief Information Officer *Information Technology Systems Engineering Handbook Version 2.0*. For guidance on entrance and exit criteria for milestone reviews of software projects, refer *to NASA-HDBK-2203, NASA Software Engineering Handbook*.

2.0 Fundamentals of Systems Engineering

At NASA, "systems engineering" is defined as a methodical, multi-disciplinary approach for the design, realization, technical management, operations, and retirement of a system. A "system" is the combination of elements that function together to produce the capability required to meet a need. The elements include all hardware, software, equipment, facilities, personnel, processes, and procedures needed for this purpose; that is, all things required to produce system-level results. The results include system-level qualities, properties, characteristics, functions, behavior, and performance. The value added by the system as a whole, beyond that contributed independently by the parts, is primarily created by the relationship among the parts; that is, how they are interconnected.[1] It is a way of looking at the "big picture" when making technical decisions. It is a way of achieving stakeholder functional, physical, and operational performance requirements in the intended use environment over the planned life of the system within cost, schedule, and other constraints. It is a methodology that supports the containment of the life cycle cost of a system. In other words, systems engineering is a logical way of thinking.

Systems engineering is the art and science of developing an operable system capable of meeting requirements within often opposed constraints. Systems engineering is a holistic, integrative discipline, wherein the contributions of structural engineers, electrical engineers, mechanism designers, power engineers, human factors engineers, and many more disciplines are evaluated and balanced, one against another, to produce a coherent whole that is not dominated by the perspective of a single discipline.[2]

Systems engineering seeks a safe and balanced design in the face of opposing interests and multiple, sometimes conflicting constraints. The systems engineer should develop the skill for identifying and focusing efforts on assessments to optimize the overall design and not favor one system/subsystem at the expense of another while constantly validating that the goals of the operational system will be met. The art is in knowing when and where to probe. Personnel with these skills are usually tagged as "systems engineers." They may have other titles—lead systems engineer, technical manager, chief engineer— but for this document, the term systems engineer is used.

The exact role and responsibility of the systems engineer may change from project to project depending on the size and complexity of the project and from phase to phase of the life cycle. For large projects, there may be one or more systems engineers. For small projects, the project manager may sometimes perform these practices. But whoever assumes those responsibilities, the systems engineering functions should be performed. The actual assignment of the roles and responsibilities of the named systems engineer may also therefore vary. The lead systems engineer ensures that the system technically fulfills the defined needs and requirements and that a proper systems engineering approach is being followed. The systems engineer oversees the

---

[1] Rechtin, Systems Architecting of Organizations: Why Eagles Can't Swim.

[2] Comments on systems engineering throughout Chapter 2.0 are extracted from the speech "System Engineering and the Two Cultures of Engineering" by Michael D. Griffin, NASA Administrator.

project's systems engineering activities as performed by the technical team and directs, communicates, monitors, and coordinates tasks. The systems engineer reviews and evaluates the technical aspects of the project to ensure that the systems/subsystems engineering processes are functioning properly and evolves the system from concept to product. The entire technical team is involved in the systems engineering process.

The systems engineer usually plays the key role in leading the development of the concept of operations (ConOps) and resulting system architecture, defining boundaries, defining and allocating requirements, evaluating design tradeoffs, balancing technical risk between systems, defining and assessing interfaces, and providing oversight of verification and validation activities, as well as many other tasks. The systems engineer typically leads the technical planning effort and has the prime responsibility in documenting many of the technical plans, including the Systems Engineering Management Plan (SEMP), ConOps, Human Systems Integration (HSI) Plan, requirements and specification documents, verification and validation documents, certification packages, and other technical documentation.

In summary, the systems engineer is skilled in the art and science of balancing organizational, cost, and technical interactions in complex systems. The systems engineer and supporting organization are vital to supporting program and Project Planning and Control (PP&C) with accurate and timely cost and schedule information for the technical activities. Systems engineering is about tradeoffs and compromises; it uses a broad crosscutting view of the system rather than a single discipline view. Systems engineering is about looking at the "big picture" and not only ensuring that they get the design right (meet requirements) but that they also get the right design (enable operational goals and meet stakeholder expectations).

Systems engineering plays a key role in the project organization. Managing a project consists of three main objectives: managing the technical aspects of the project, managing the project team, and managing the cost and schedule. As shown in Figure 2.0-1, these three functions are interrelated. Systems engineering is tightly related to the technical aspects of program and project management. As discussed in NPR 7120.5, NASA Space Flight Program and Project Management Requirements, project management is the function of planning, overseeing, and directing the numerous activities required to achieve the requirements, goals, and objectives of the customer and other stakeholders within specified cost, quality, and schedule constraints. Similarly, NPR 7120.8, NASA Research and Technology Program and Project Management Requirements, states that the program or project lead (i.e., management) is responsible for the formulation and implementation of the R&T program or project and NPR 7120.7, NASA Information Technology and Institutional Infrastructure Program and Project Management Requirements, refers project managers to NPR 7123.1, NASA Systems Engineering Processes and Requirements, for systems engineering requirements. Systems engineering is focused on the technical characteristics of decisions including technical, cost, and schedule and on providing these to the project manager. The project manager is responsible for ensuring that the project delivers the system within cost and schedule. The overlap in these responsibilities is natural, with the systems engineer focused on the success of the engineering of the system (technical, cost, schedule) and the project manager providing constraints on engineering options to maintain a successful delivery of the system within cost and schedule. These areas are systems engineering and project control. Figure 2.0-1 is a notional graphic depicting this concept. Note that there are areas where the two cornerstones of project management, SE and PP&C, overlap. In these areas,

SE provides the technical aspects or inputs; whereas PP&C provides the programmatic, cost, and schedule inputs.

This document focuses on the SE side of the diagram. The practices/processes are taken from NPR 7123.1, NASA Systems Engineering Processes and Requirements. Each process is described in much greater detail in subsequent chapters of this document, but an overview is given in the following subsections of this chapter.

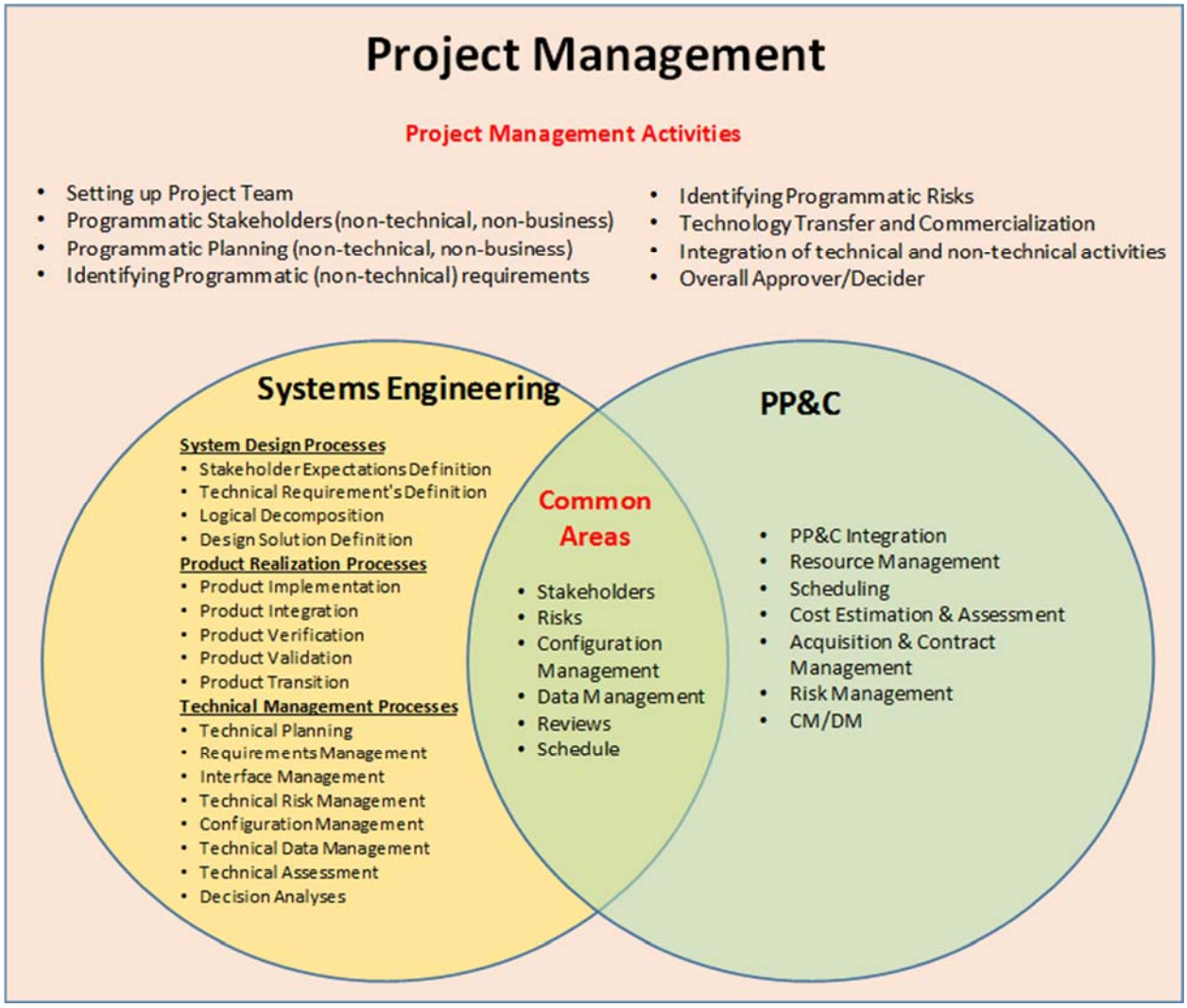


Figure 2.0-1 SE in Context of Overall Project Management

A NASA systems engineer can participate in the NASA Engineering Network (NEN) Systems Engineering Community of Practice, located at https://nen.nasa.gov/web/se. This Web site includes many resources useful to systems engineers, including document templates for many of the work products and milestone review presentations required by the NASA SE process.

2.1 The Common Technical Processes and the SE Engine

There are three sets of common technical processes in NPR 7123.1, NASA Systems Engineering Processes and Requirements: system design, product realization, and technical management. The processes in each set and their interactions and flows are illustrated by the NPR systems engineering "engine" shown in Figure 2.1-1. The processes of the SE engine are used to develop and realize the end products. This chapter provides the application context of the 17 common technical processes required in NPR7123.1. The system design processes, the product realization processes, and the technical management processes are discussed in more detail in Chapters 4.0, 5.0, and 6.0, respectively. Processes 1 through 9 indicated in Figure 2.1-1 represent the tasks in the execution of a project. Processes 10 through17 are crosscutting tools for carrying out the processes.

- **System Design Processes:** The four system design processes shown in Figure 2.1-1 are used to define and baseline stakeholder expectations, generate and baseline technical requirements, decompose the requirements into logical and behavioral models, and convert the technical requirements into a design solution that will satisfy the baselined stakeholder expectations. These processes are applied to each product of the system structure from the top of the structure to the bottom until the lowest products in any system structure branch are defined to the point where they can be built, bought, or reused. All other products in the system structure are realized by implementation or integration.

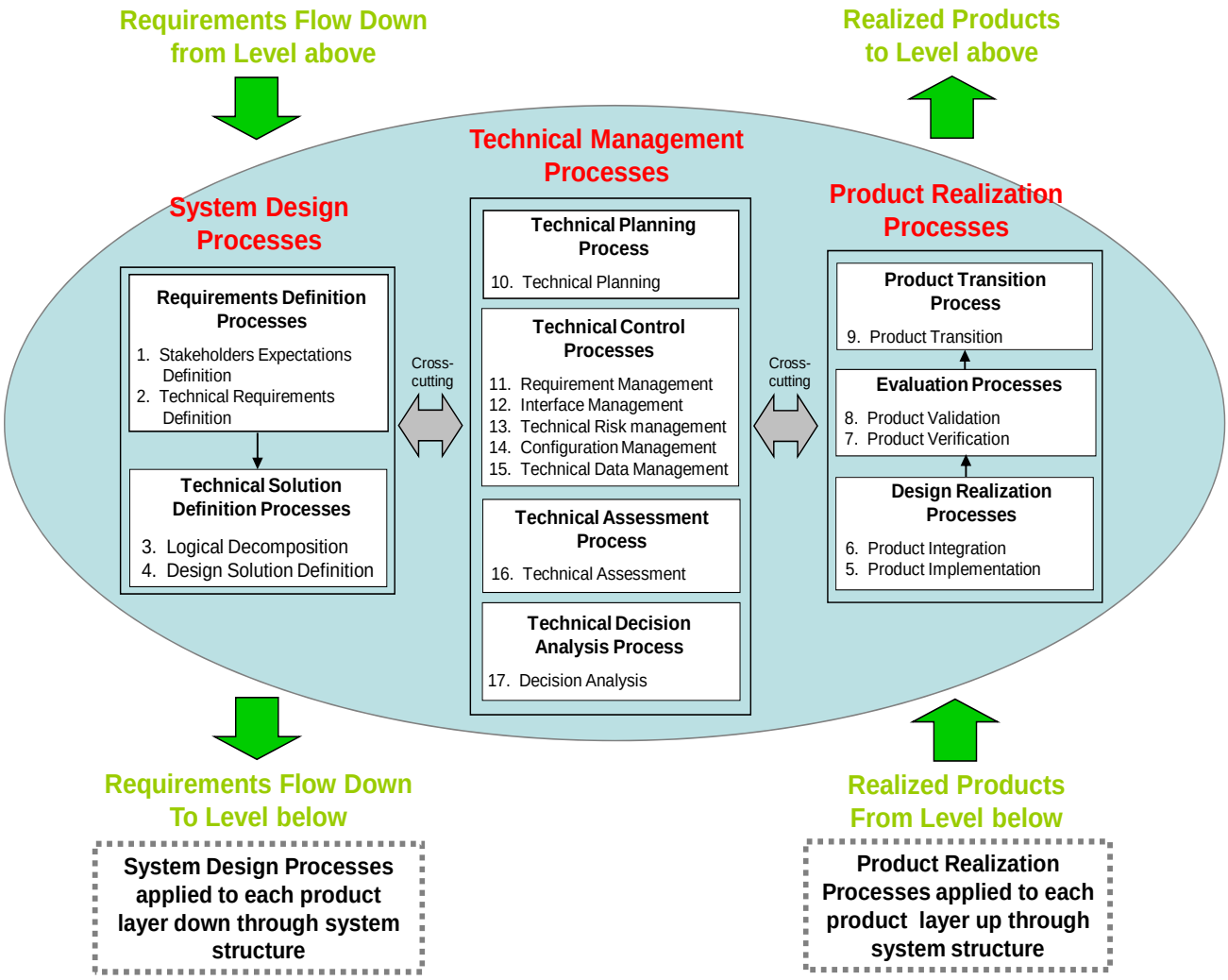


Figure 2.1-1 The Systems Engineering Engine (NPR 7123.1)

- **Product Realization Processes:** The product realization processes are applied to each operational/mission product in the system structure starting from the lowest level product and working up to higher level integrated products. These processes are used to create the design solution for each product (through buying, coding, building, or reusing) and to verify, validate, and transition up to the next hierarchical level those products that satisfy their design solutions and meet stakeholder expectations as a function of the applicable life-cycle phase.
- **Technical Management Processes:** The technical management processes are used to establish and evolve technical plans for the project, to manage communication across interfaces, to assess progress against the plans and requirements for the system products or services, to control technical execution of the project through to completion, and to aid in the decision-making process.

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

AS9100 is a widely adopted and standardized quality management system developed for the commercial aerospace industry. Some NASA Centers have chosen to certify to the AS9100 quality system and may require their contractors to follow NPR 7123.1. Table 2.1-1 shows how the 17 NASA SE processes align with AS9100.

Table 2.1-1 Alignment of the 17 SE Processes to AS9100

SE Process	AS9100 Requirement
Stakeholder Expectations	Customer Requirements
Technical Requirements Definition	Planning of Product Realization
Logical Decomposition	Design and Development Input
Design Solution Definition	Design and Development Output
Product Implementation	Control of Production
Product Integration	Control of Production
Product Verification	Verification
Product Validation	Validation
Product Transition	Control of Work Transfers; Post Delivery Support, Preservation of Product
Technical Planning	Planning of Product Realization; Review of Requirements; Measurement, Analysis and Improvement
Requirements Management	Design and Development Planning; Purchasing
Interface Management	Configuration Management
Technical Risk Management	Risk Management
Configuration Management	Configuration Management; Identification and Traceability; Control of Nonconforming Product
Technical Data Management	Control of Documents; Control of Records; Control of Design and Development Changes
Technical Assessment	Design and Development Review
Decision Analysis	Measurement, Analysis and Improvement; Analysis of Data

2.2 An Overview of the SE Engine by Project Phase

Figure 2.2-1 conceptually illustrates how the SE engine is used during each phase of a project (Pre-Phase A through Phase F). Figure 2.2-1 is a *conceptual* diagram. For all of the details, refer to the poster version of this figure, which can be located at https://nen.nasa.gov/web/se/doc-repository.

The uppermost horizontal portion of this chart is used as a reference to project system maturity, as the project progresses from a feasible concept to an as-deployed system; phase activities; Key Decision Points (KDPs); and major project reviews. The next major horizontal band shows the technical development processes (steps 1 through 9) in each project phase. The SE engine cycles five times from Pre-Phase A through Phase D. Note that NASA's management has structured Phases C and D to "split" the technical development processes in half in Phases C and D to ensure closer management control. The engine is bound by a dashed line in Phases C and D. Once a project enters into its operational state (Phase E) and closes out (Phase F), the technical work shifts to activities commensurate with these last two project phases. The next major horizontal band shows the eight technical management processes (steps 10 through 17) in each project phase. The SE engine cycles the technical management processes seven times from Pre-Phase A through Phase F.

Each of the SE engine entries is given a 6105 paragraph label that is keyed to chapters 4.0, 5.0, and 6.0 in this guide. For example, in the technical development processes, "Get Stakeholder Expectations" discussions and details are in Section 4.1.

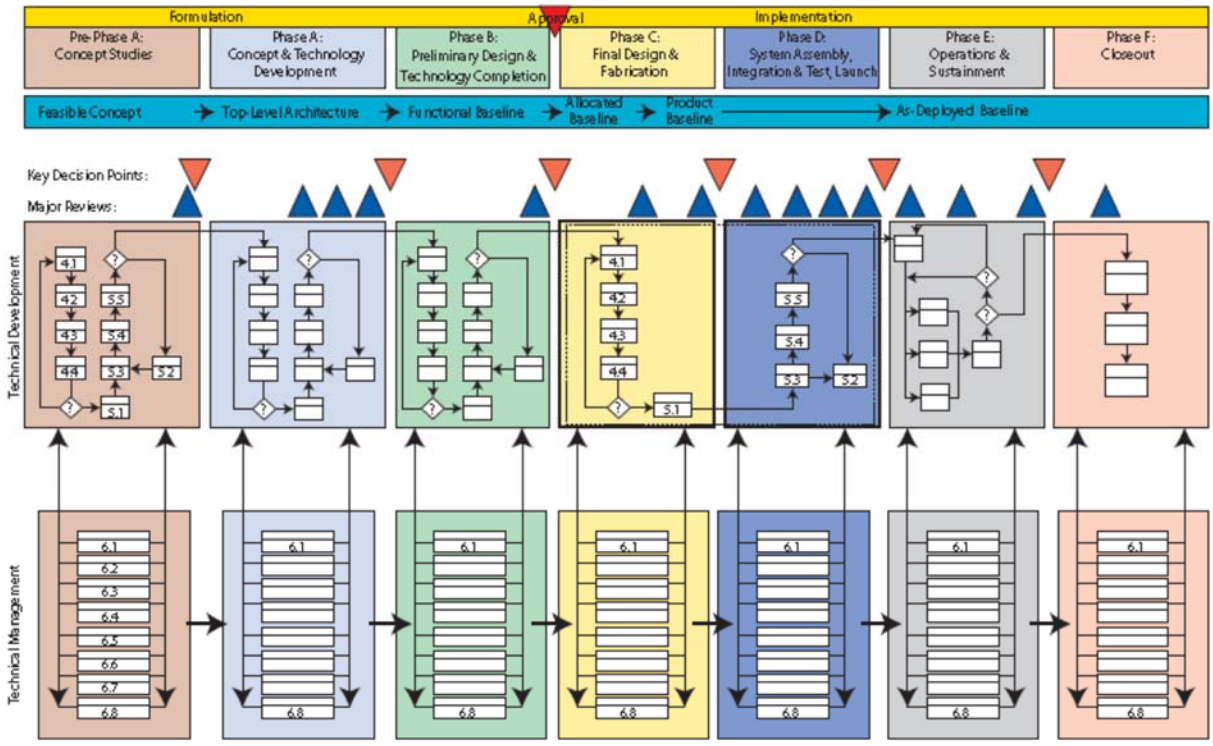


Figure 2.2-1 A Miniature Version of the Poster-Size NASA Project Life Cycle Process Flow for Flight and Ground Systems Accompanying this Guide

2.3 Example of Using the SE Engine

To help in understanding how the SE engine is applied, an example is provided below. Pertinent to this discussion are the phases of the program and project life cycles, which are discussed in greater depth in chapter 3.0 of this document. As described in chapter 3.0, NPR 7120.5 defines the life cycle used for NASA space flight programs and projects. The life cycle phases are described in Table 2.3-1.

Use of the different phases of a life cycle allows the various products of a project to be gradually developed and matured from initial concepts through the fielding of the product and to its final retirement. The SE engine shown in figure 2.1-1 is used throughout all phases.

In Pre-Phase A, the SE engine is used to develop the initial concepts; clearly define the unique roles of humans, hardware, and software in performing the missions objectives; establish the system functional and performance boundaries; develop/identify a preliminary/draft set of key high-level requirements, define one or more initial Concept of Operations (ConOps) scenarios; realize these concepts through iterative modeling, mockups, simulation, or other means; and verify and validate that these concepts and products would be able to meet the key high-level requirements and ConOps. The operational concept must include scenarios for all significant operational situations, including known off-nominal situations. To develop a useful and complete set of scenarios, important malfunctions and degraded-mode operational situations must be considered. The importance of early ConOps development cannot be underestimated. As system requirements become more detailed and contain more complex technical information, it becomes harder for the stakeholders and users to understand what the requirements are conveying; i.e., it may become more difficult to visualize the end product. The ConOps can serve as a check in identifying missing or conflicting requirements.

Note that this Pre-Phase A initial concepts development work is not the formal verification and validation program that is performed on the final product, but rather it is a methodical runthrough ensuring that the concepts that are being developed in this Pre-Phase A are able to meet the likely requirements and expectations of the stakeholders. Concepts are developed to the lowest level necessary to ensure that they are feasible and to a level that reduces the risk low enough to satisfy the project. Academically, this process could proceed down to the circuit board level for every system; however, that would involve a great deal of time and money. There may be a higher level or tier of product than circuit board level that would enable designers to accurately determine the feasibility of accomplishing the project, which is the purpose of Pre-Phase A.

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

During Phase B, the SE engine is applied recursively to further mature requirements and designs for all products in the developing product tree and perform verification and validation of concepts to ensure that the designs are able to meet their requirements. Operational designs and

mission scenarios are evaluated and feasibility of execution within design capabilities and cost estimates are assessed.

Phase C again uses the left side of the SE engine to finalize all requirement updates, finalize the ConOps validation, develop the final designs to the lowest level of the product tree, and begin fabrication. Phase D uses the right side of the SE engine to recursively perform the final implementation, integration, verification, and validation of the end product, and at the final pass, transition the end product to the user. The technical management processes of the SE engine are used in Phases E and F to monitor performance; control configuration; and make decisions associated with the operations, sustaining engineering, and closeout of the system. Any new capabilities or upgrades of the existing system reenter the SE engine as new developments.

Table 2.3-1 Project Life Cycle Phases

Phase		Purpose	Typical Outcomes
Pre-Formulation	Pre-Phase A Concept Studies	To produce a broad spectrum of ideas and alternatives for missions from which new programs/projects can be selected. Determine feasibility of desired system, develop mission concepts, draft system-level requirements, assess performance, cost, and schedule feasibility; identify potential technology needs, and scope.	Feasible system concepts in the form of simulations, analysis, study reports, models, and mockups
Formulation	Phase A Concept and Technology Development	To determine the feasibility and desirability of a suggested new system and establish an initial baseline compatibility with NASA's strategic plans. Develop final mission concept, system-level requirements, needed system technology developments, and program/project technical management plans.	System concept definition in the form of simulations, analysis, engineering models and mockups, and trade study definition
	Phase B Preliminary Design and Technology Completion	To define the project in enough detail to establish an initial baseline capable of meeting mission needs. Develop system structure end product (and enabling product) requirements and generate a preliminary design for each system structure end product.	End products in the form of mockups, trade study results, specification and interface documents, and prototypes
Implementation	Phase C Final Design and Fabrication	To complete the detailed design of the system (and its associated subsystems, including its operations systems), fabricate hardware, and code software. Generate final designs for each system structure end product.	End product detailed designs, end product component fabrication, and software development
	Phase D System Assembly, Integration and Test, Launch	To assemble and integrate the system (hardware, software, and humans), meanwhile developing confidence that it is able to meet the system requirements. Launch and prepare for operations. Perform system end product implementation, assembly, integration and test, and transition to use.	Operations-ready system end product with supporting related enabling products
	Phase E Operations and Sustainment	To conduct the mission and meet the initially identified need and maintain support for that need. Implement the mission operations plan.	Desired system
	Phase F Closeout	To implement the systems decommissioning/disposal plan developed in Phase E and perform analyses of the returned data and any returned samples.	Product closeout

2.3.1 Detailed Example of SE Engine: Space Transportation System

The NASA Space Transportation System (STS) can be used as an example to look at how the SE engine is used in Phase A. This example is simplified to illustrate the application of the SE processes in the engine and is in no way as detailed as necessary to actually build the highly complex vehicle. The SE engine is used recursively to drive out more and more detail with each pass. The icon shown in Figure 2.3-1 is used to keep track of the applicable place in the SE engine. The numbers in the icon correspond to the numbered processes within the SE engine as shown in Figure 2.1-1. As the design is matured, a Product Breakdown Structure (PBS) is developed that identifies all of the products that the project will produce and shows how the final product breaks down into smaller and smaller pieces. For the purposes of this guide, the various layers of this product hierarchy are called "tiers", "layers," or "levels." (See Section 4.3.2.1 for more information on the PBS). But basically, the higher the number of the tier or level, the lower in the product hierarchy the product is going and the more detailed the product is becoming (e.g., going from boxes, to circuit boards, to components).

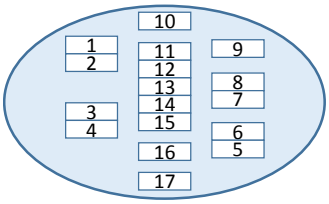


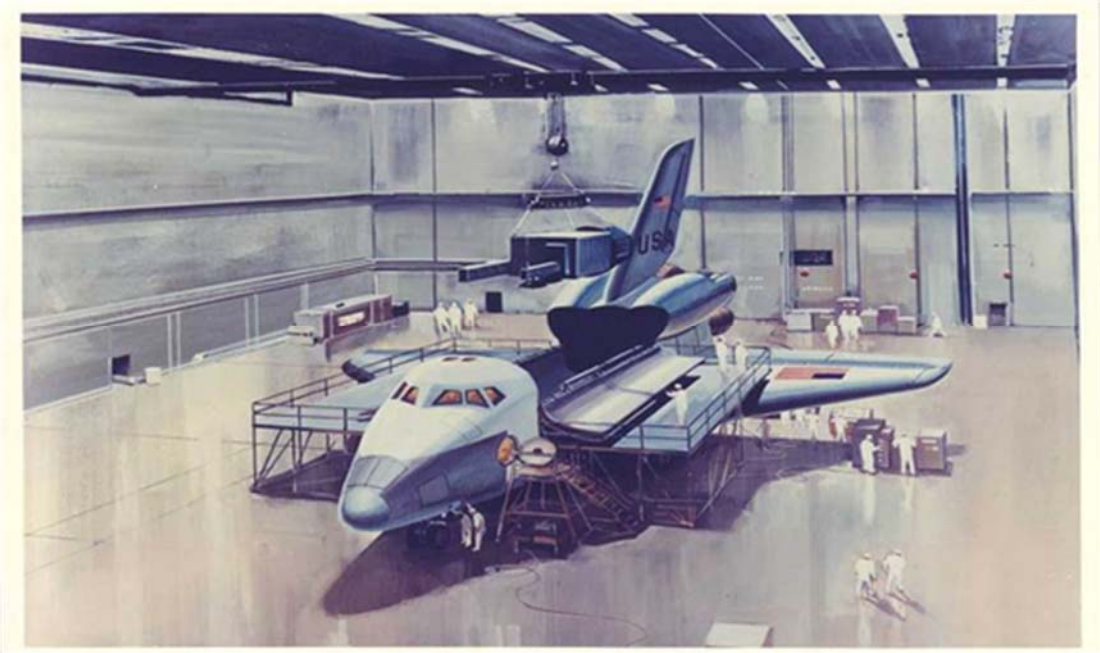
Figure 2.3-1 SE Engine Tracking Icon

2.3.2 Example Premise

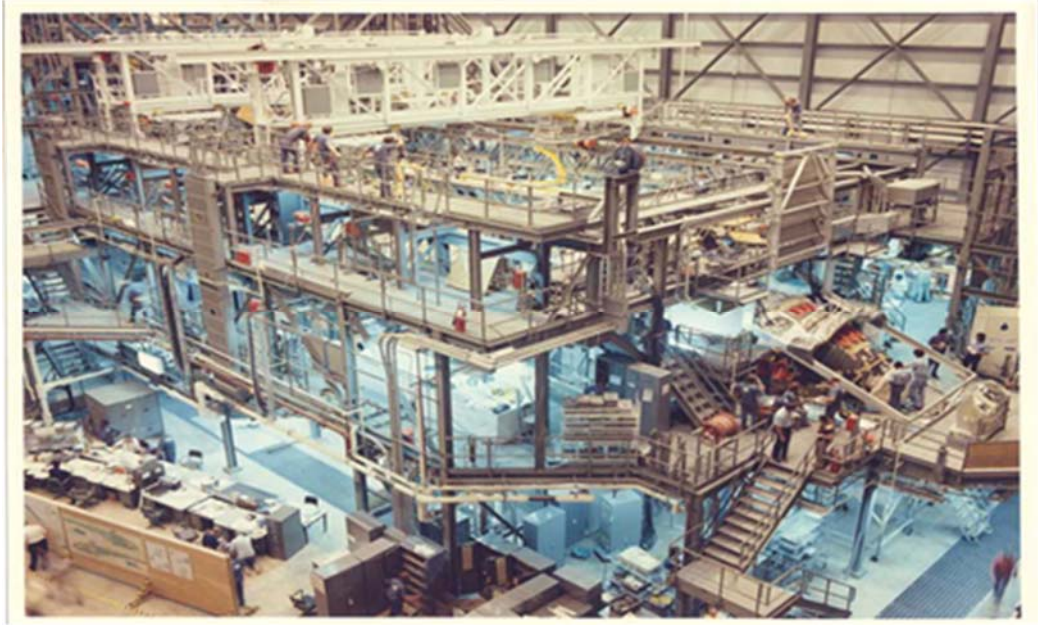
2.3.2.1 Example Pre-Phase A

NASA decides that there is a need for a transportation system that acts like a "truck" to carry large pieces of equipment into Low Earth Orbit (LEO). Referring back to the project life cycle, the project first enters the Pre-Phase A. During this phase, several concept studies are performed, and through combinations of simulations, mockups, analyses, or other like means, it is determined that it is feasible to develop such a "space truck." For simplicity, assume feasibility is proven through concept models and an initial ConOps is developed that preliminarily identifies roles, responsibilities, numbers, and skillsets of humans interacting with the system to ensure full operational effectiveness. The processes and framework of the SE engine are used to design and implement these models.

The project then enters the Phase A activities to refine Pre-Phase A concepts and define the system requirements for the preferred solution. The detailed example begins in Phase A and shows how the SE engine is used. As described in the overview, a similar process is used for the other project phases. Note that the concepts and amount of time spent in this phase are important to thoroughly understand the real-life implementation of these concepts. For example, Figure 2.3-2 shows the initial concept for servicing the Space Shuttle and how it really ended up. Having operations personnel as part of the concept development team helps identify the true life-cycle needs.



Initial Servicing Concept



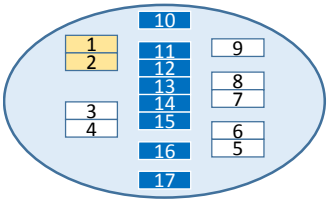
Actual Servicing Configuration

Figure 2.3-2 Initial Shuttle Servicing Concept and Actual Servicing Configuration

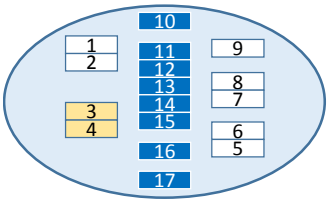
2.3.2.2 Example Phase A System Design Passes

2.3.2.2.1 First Pass

With the preliminary concepts and key system requirements developed during the Pre-Phase A activities, the product enters the SE engine at the first process where the product (i.e., the STS) stakeholders and what they want are determined. During Pre-Phase A, these needs and expectations were very general ideas, such as "The primary mission of the Space Shuttle is the delivery of payloads to and from Earth's orbit and the deployment of a space station." During this Phase A pass, these general concepts are detailed out and agreed upon. The ConOps generated in Pre-Phase A is also detailed out and agreed upon to ensure all stakeholders are in agreement as to what is really expected of the product—in this case the transportation system. The detailed expectations are then converted into good requirement statements. (For more information on what constitutes a good requirement, see appendix C.) For example, a requirement at this stage might be "Nominal separation of the Solid Rocket Boosters (SRBs) from the Orbiter/External Tank (ET) shall occur only after SRB burnout." Subsequent passes and subsequent phases will refine these requirements into specifications that can actually be built. Also note that all of the technical management processes (SE engine processes numbered 10 through 17) are also used during this and all subsequent passes and activities. These ensure that all the proper planning, control, assessment, and decisions are used and maintained, and that the necessary balance between the competing interests and disciplines is achieved. These processes can have significant impacts on the design and can generate additional requirements, so they should not be overlooked (e.g., risk management, interface control, and HSI activities). Although for simplification, the technical management processes are not mentioned in the rest of this example, they are *always* in effect.



Next, using the requirements and the ConOps previously developed, logical decomposition models/diagrams are built up to help bring the requirements into perspective and to show their relationships. Finally, these diagrams, requirements, and ConOps documents are used to develop one or more feasible design solutions. Note that at this point, since this is only the first pass through the SE engine, these design solutions are not detailed enough to actually build anything. Consequently, the design solutions might be summarized as, "To accomplish this transportation system, the best option in our trade studies is a three-part system: a reusable orbiter for the crew and cargo, a large external tank to hold the propellants, and two solid rocket boosters to give extra power for liftoff that can be recovered, refurbished, and reused." (Of course, the actual design solution would be much more descriptive and detailed). This can be illustrated as shown in Figure 2.3-3.



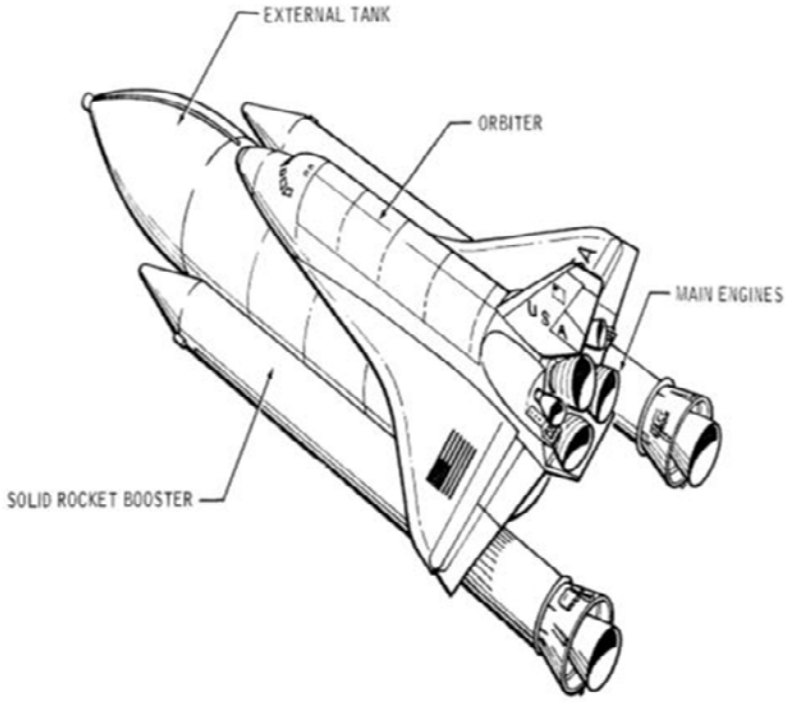


Figure 2.3-3 Initial Architecture Concept for the Space Shuttle

So, for this first pass, the first tier of the product hierarchy might look like Figure 2.3-4. There would also be other enabling products that might appear in the product tree, but for simplicity only, the main products are shown in the example.

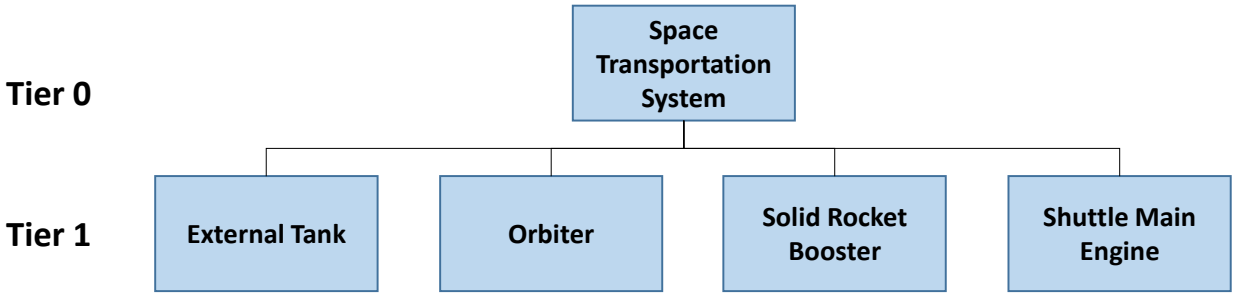
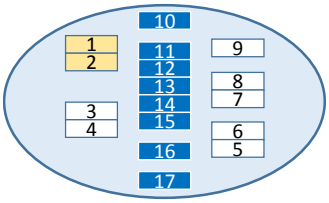


Figure 2.3-4 Product Hierarchy, Tier 1: First Pass through the SE Engine

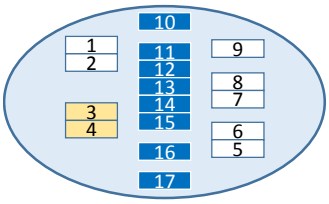
Again, the design solution is not yet at a detailed enough level to actually build the prototypes or models of any of these products. The requirements, ConOps, functional diagrams, and design solutions are still at a very high, general level. Note that the SE processes on the right side (i.e., the product realization processes) of the SE engine have not yet been addressed. The design should first be at a level where something can actually be built, coded, or reused before that side of the SE engine can be used, so a second pass of the left side of the SE engine is started.

2.3.2.2.2 Second Pass

The SE engine is completely recursive; that is, each of the three elements shown in the tier 1 diagram can now be considered a separate product, and the SE engine is therefore applied to each of the three elements separately. For example, the external tank is considered an end product, and the SE engine resets back to the first processes. So just focusing on the external tank, who the stakeholders are and what they expect of the external tank is determined. Of course, one of the main stakeholders will be the owners of the tier 1 requirements and the STS as an end product, but there will also be other new stakeholders. Needs, Goals, and Objectives (NGOs) for the external tank are generated. For example, one NGO might be "The external tank contains and delivers cryogenic propellants for the orbiter's main engines." A new ConOps for how the external tank operates is also generated. The tier 1 requirements that are applicable (allocated) to the external tank are "flowed down" and validated. Usually, some of these requirements are too general to implement into a design, so they have to be decomposed and allocated. To these derived requirements, new requirements generated from the stakeholder expectations are added, as well as other applicable standards for workmanship, safety, quality, maintainability, ground processing, etc.



Next, the external tank requirements and the external tank ConOps are established, and functional diagrams are developed as was done in the first pass with the STS product. Finally, these diagrams, requirements, and ConOps documents are used to develop some feasible design solutions for the external tank. At this pass, there will still not be enough detail to actually build or prototype the external tank. The design solution might be summarized as, "To build this external tank, since our trade studies showed the best option was to use cryogenic propellants, a tank for the liquid hydrogen will be needed as will another tank for the liquid oxygen, instrumentation, and an outer structure of aluminum coated with foam." Thus, the tier 2 product tree for the external tank might look like Figure 2.3-5.



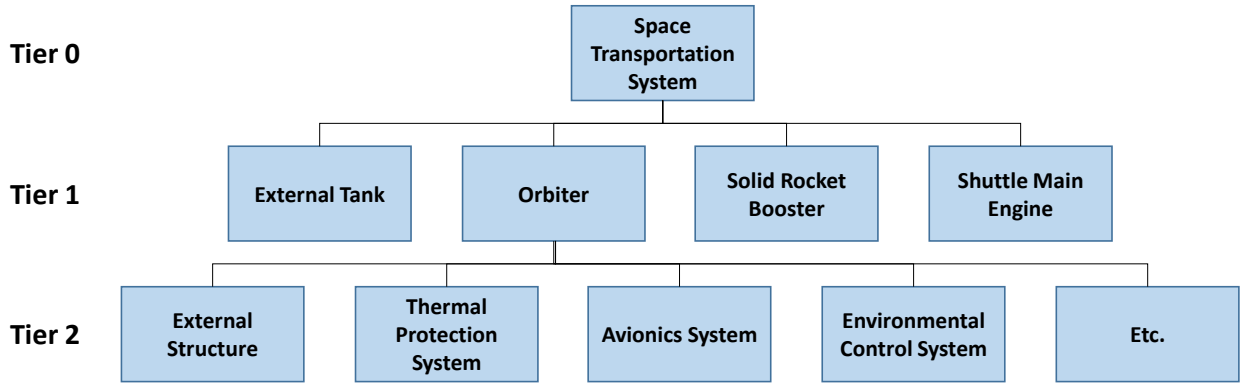


Figure 2.3-5 Product Hierarchy, Tier 2: External Tank Notional Example PBS

In a similar manner, the orbiter would also take another pass through the SE engine to identify the stakeholders and their expectations, develop NGOs, and generate a ConOps for the orbiter element. An example of an orbiter NGO might be: "The orbiter will transport payloads weighing up to 65,000 lbm into near-Earth orbit." The tier 1 requirements that are applicable (allocated) to the orbiter are "flowed down" and validated; new requirements derived from them and any additional requirements (including interfaces with the other elements) are added. An example of an added or derived requirement at this stage might be: "The cabin shall accommodate a total of seven crew members."

Next, the orbiter requirements and the ConOps are taken, functional diagrams are developed, and one or more feasible design solutions for the orbiter are generated. As with the external tank, at this pass, there is not enough detail to actually build or do a complex model of the orbiter. The orbiter design solution might be summarized as, "To build this orbiter will require a winged vehicle with a thermal protection system; an avionics system; a guidance, navigation, and control system; a propulsion system; an environmental control system; etc." So the tier 2 product tree for the orbiter element might look like Figure 2.3-6.

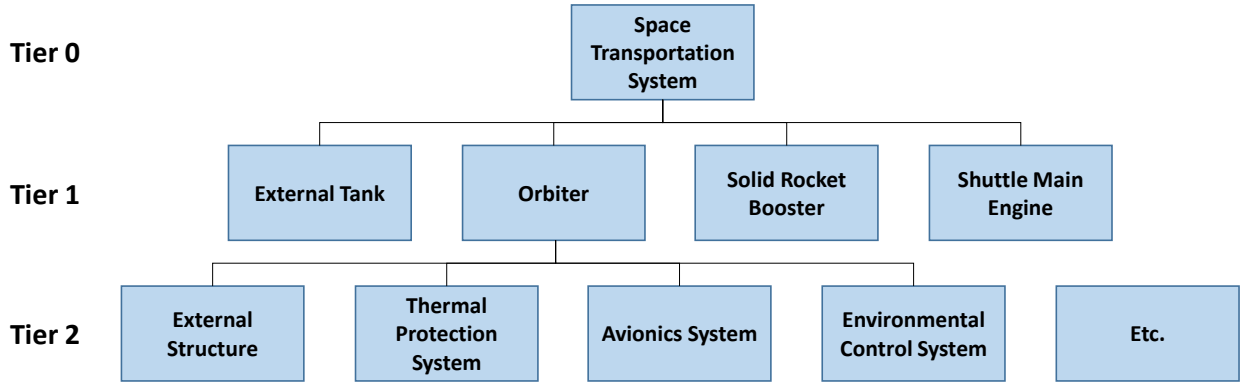


Figure 2.3-6 Product Hierarchy, Tier 2: Orbiter Notional Example PBS

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

2.3.2.2.3 Third Pass

Each of the tier 2 elements is also considered an end product, and each undergoes another pass through the SE engine to define stakeholders, generate ConOps, flow down allocated requirements, generate new and derived requirements, and develop functional diagrams and design solution concepts. As an example of just the avionics system element, the tier 3 product hierarchy tree might look like Figure 2.3-7.

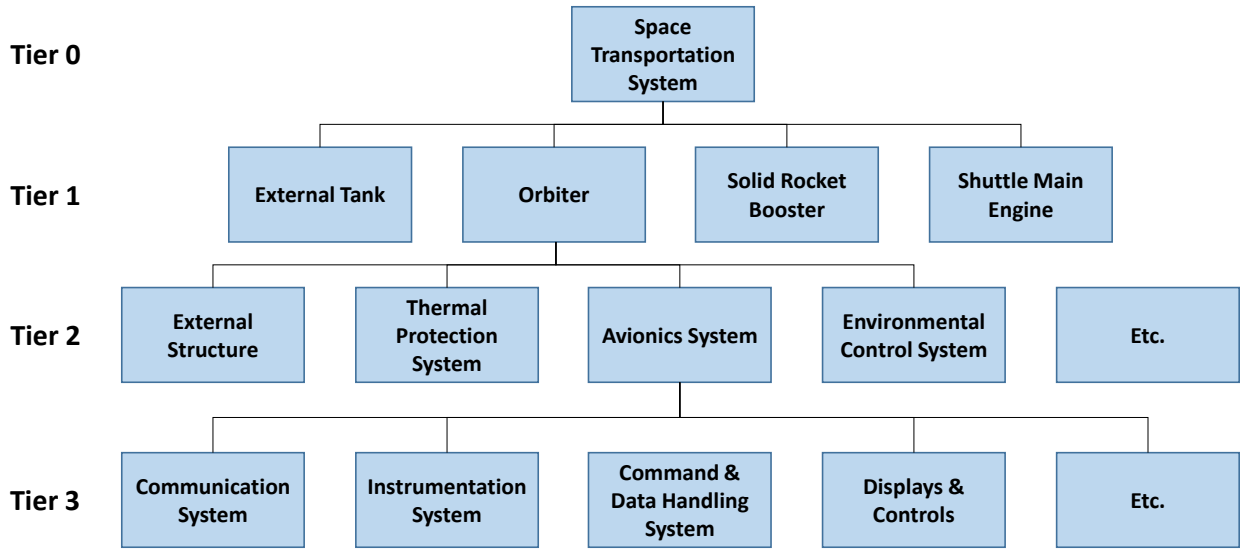
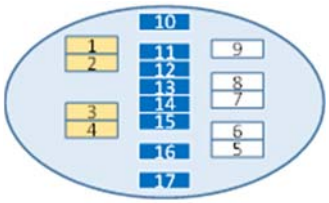


Figure 2.3-7 Product Hierarchy, Tier 3: Orbiter Avionics System Notional PBS

2.3.2.2.4 Passes 4 through n

For the Phase A set of passes, the recursive process is continued for each product (model) on each tier down to the lowest level in the product tree. Note that in some projects it may not be necessary, given an estimated project cost and schedule, to perform this recursive process completely down to the smallest component during Phase A. It is only necessary to go down to the level needed to converge the system level design to the first-order n cost, technical, and schedule trade space. This is not always intuitive; it takes experience to know when to stop at the appropriate level.

In these cases, engineering judgment should be used to determine what level of the product is feasible. Note that the lowest feasible level may occur at different tiers depending on the product complexity. For example, for one type of product it may occur at tier 2; whereas, for a more complex product, it could occur at tier 8. This also means that it will take different amounts of time to reach the bottom. Thus, for any given program or project, products can be at various stages of development. For this Phase A example, Figure 2.3-8 depicts the STS product hierarchy after completely passing through the system design processes side of the SE engine. At the end of this set of passes, each product in the tree has system requirements, a ConOps, and high-level conceptual, functional, and physical architectures. Note that these are not yet the detailed or even preliminary designs for the end products, which come later in the life cycle. At this point, enough conceptual design work has been done to ensure that at least the high-risk requirements are achievable as is shown in the following passes.



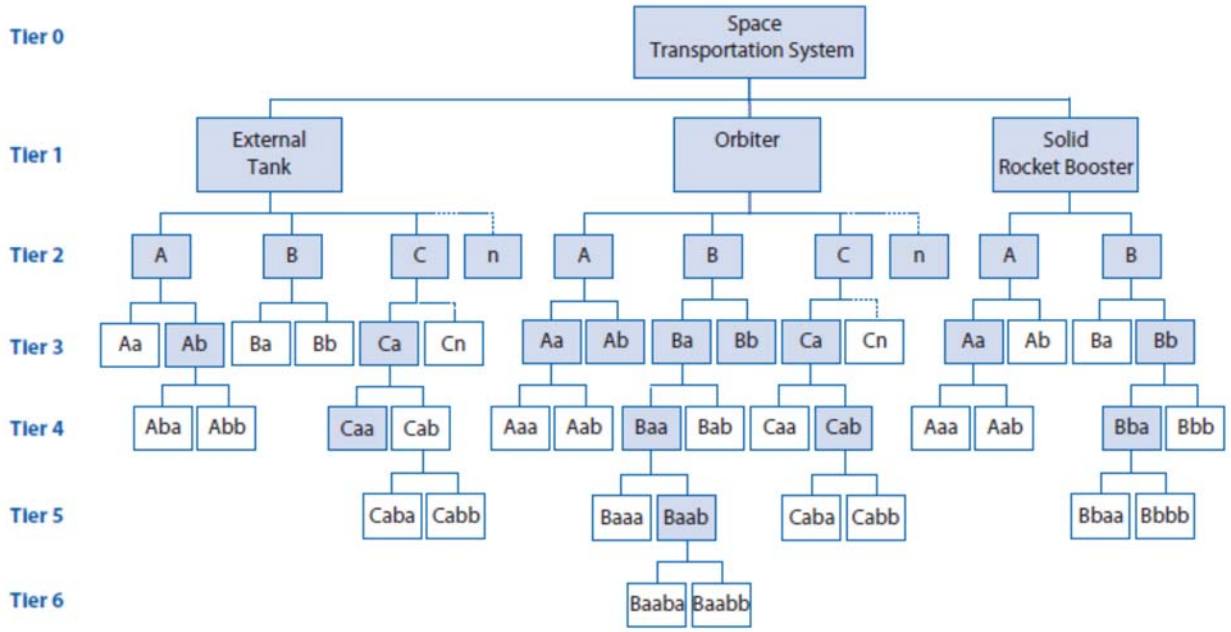


Figure 2.3-8 Product Hierarchy: Complete Pass through System Design Processes Side of the SE Engine

Note: The unshaded boxes represent bottom-level phase products.

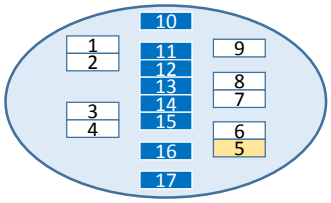
2.3.2.3 Example Product Realization Passes

So now that the requirements and conceptual designs for the principal Phase A products have been developed, they need to be checked to ensure they are achievable. Note that there are two types of products. The first product is the "end product"—the one that will actually be delivered to the final user. The second type of product is called a "phase product." A phase product is generated within a particular life-cycle phase that helps move the project toward delivering a final product. For example, while in Pre-Phase A, a foam-core mockup, an additive manufacturing model, or an interactive computer model might be built to help visualize some of the concepts. Such mockups are not the final "end product," but rather the "phase product." For this Phase A example, assume some computer models are created and simulations performed of these key concepts to show that they are achievable. These are the phase product for our example.

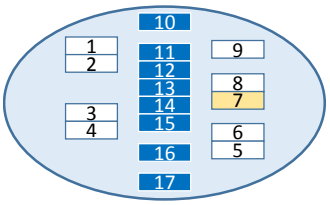
Now the focus shifts to the right side (i.e., product realization processes) of the SE engine, which is applied recursively, starting at the bottom of the product hierarchy and moving upwards.

2.3.2.3.1 First Pass

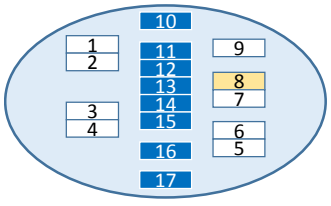
Each of the phase products (i.e., our computer models) for the bottom-level product tier (ones that are unshaded in Figure 2.3-6) is taken individually and realized; that is, it is either bought, built, coded, or reused. For our example, assume the external tank product model Aa is a standard Commercial Off-the-Shelf (COTS) product that is bought. Aba is a model that can be reused from another project, and product Abb is a model that will have to be developed with an in-house design that is to be built. Note that these models are parts of a larger model product that will be assembled or integrated on a subsequent pass of the SE engine. That is, to realize the model for product Ab of the external tank, models for products Aba and Abb should be first implemented and then later integrated together. This pass of the SE engine will be the realizing part. Likewise, each of the unshaded bottom-level model products is realized in this first pass. The models help us understand and plan the method to implement the final end product and ensure the feasibility of the implemented method.



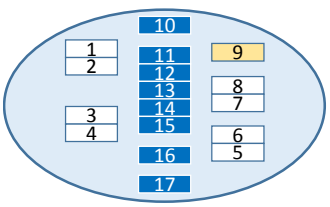
Next, each of the realized models (phase products) is used to verify that the end product would likely meet the requirements as defined in the Technical Requirements Definition Process during the system design pass for the product. This shows the product would likely meet the "shall" statements that were allocated, derived, or generated for it by method of test, analysis, inspection, or demonstration—that it was "built right." Verification is performed for each of the unshaded bottom-level model products. Note that during this Phase A pass, this process is not the formal verification of the final end product. However, using analysis, simulation, models, or other means shows that the requirements are good (verifiable) and that the concepts will most likely satisfy them. This also allows draft verification procedures of key areas to be developed. What can be formally verified, however, is that the phase product (the model) meets the requirements for the model.



After the phase product (models) has been verified and used for planning the end product verification, the models are then used for validation. That is, additional test, analysis, inspection, or demonstrations are conducted to ensure that the proposed conceptual designs will likely meet the expectations of the stakeholders for this phase product and for the end product. This will track back to the ConOps that was mutually developed with the stakeholders during the Stakeholder Expectations Definition Process of the system design pass for this product. This will help ensure that the project has "built the right" product at this level.



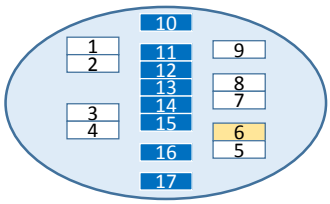
After verification and validation of the phase product (models) and using it for planning the verification and validation of the end product, it is time to prepare the model for transition to the next level up. Depending on complexity, where the model will be transitioned, security requirements, etc., transition may involve crating and shipment, transmitting over a network, or hand carrying over to the next lab. Whatever is



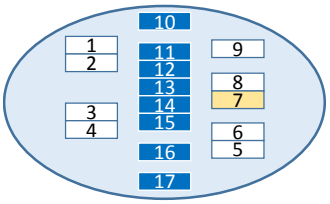
appropriate, each model for the bottom-level product is prepared and handed to the next level up for further integration.

2.3.2.3.2 Second Pass

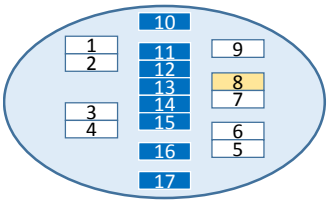
Now that all the models (phase products) for the bottom-level end products are realized, verified, validated, and transitioned, it is time to start integrating them into the next higher level product. For example, for the external tank, realized tier 4 models for product Aba and Abb are integrated to form the model for the tier 3 product Ab. Note that the Product Implementation Process only occurs at the bottommost product. All subsequent passes of the SE engine will employ the Product Integration Process since already realized products will be integrated to form the new higher-level products. Integrating the lower-tier phase products will result in the next-higher-tier phase product. This integration process can also be used for planning the integration of the final end products.



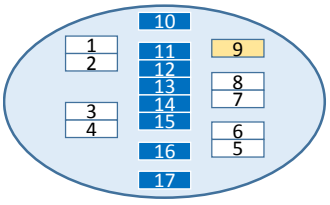
After the new integrated phase product (model) has been formed (tier 3 product Ab, for example), it should now be proven that it meets the allocated, derived, or generated requirements developed for it during the Technical Requirements Definition Process during the system design pass. This ensures that the integrated product was built (assembled) right. Note that just verifying the component parts (i.e., the individual models) that were used in the integration is not sufficient to assume that the integrated product will work right. There are many sources of problems that could occur: incomplete requirements at the interfaces, wrong assumptions during design, etc. The only sure way of knowing if an integrated product is good is to perform verification and validation at each stage. The knowledge gained from verifying this integrated phase product can also be used for planning the verification of the final end products.



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

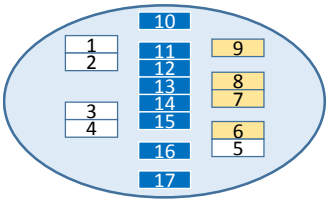


The model for the integrated phase product at this level (tier 3 product Ab for example) is now ready to be transitioned to the next higher level (tier 2 for the example). As with the products in the first pass, the integrated phase product is prepared according to its needs/requirements and shipped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



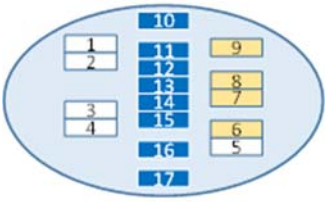
2.3.2.3.3 Passes 3 through n

In a similar manner as the second pass, the tier 3 models for the products are integrated together, realized, verified, validated, and transitioned to the next higher tier. For the example, the realized model for external tank tier 3 integrated phase product Ab is integrated with the model for tier 3 realized phase product Aa to form the tier 2 phase product A. Note that tier 3 product Aa is a bottom-tier product that has yet to go through the integration process. It may also have been realized some time ago and has been waiting for the Ab product to become realized. Part of its transition might have been to place it in secure storage until the Ab product became available. Or it could be that Aa was the long-lead item and product Ab had been completed some time ago and was waiting for the Aa purchase to arrive before they could be integrated together. The length of the branch of the product tree does not necessarily translate to a corresponding length of time. This is why good planning in the first part of a project is so critical.



2.3.2.3.4 Final Pass

At some point, all the models for the tier 1 phase products will each have been used to ensure the system requirements and concepts developed during this Phase A cycle can be implemented, integrated, verified, validated, and transitioned. The elements are now defined as the external tank, the orbiter, and the solid rocket boosters. One final pass through the SE engine will show that they will likely be successfully implemented, integrated, verified, and validated. The final of these products—in the form of the baselined system requirements, ConOps, conceptual, functional, and physical designs—is made to provide inputs into the next life-cycle phase (B) where they will be further matured. In later phases, the products will actually be built into physical form. At this stage of the project, the key characteristics of each product are passed downstream in key SE documentation, as noted.



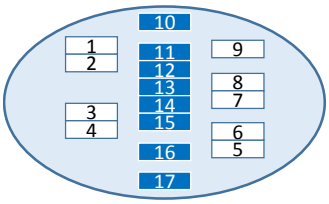
2.3.2.4 Example Use of the SE Engine in Phases B through D

Phase B begins the preliminary design of the final end-product. The recursive passes through the SE engine are repeated in a similar manner to that discussed in the detailed Phase A example. At this phase, the phase product might be a prototype of the product(s). Prototypes can be developed and then put through the planned verification and validation processes to ensure the design will likely meet all the requirements and expectations prior to the build of the final flight units. Any mistakes found in prototypes are much easier and less costly to correct than if not found until the flight units are built and undergoing the certification process. Prototypes include, but are not limited to, physical mockups, interactive human integration computer models, and additive manufacturing models (or test components).

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

in Phase D, we have the final fully realized end product, the STS, ready to be delivered for launch. The supporting infrastructure elements (e.g., launch facility, mission control systems, communication relays, and trained ground operators) are also delivered.

2.3.2.5 Example Use of the SE Engine in Phases E and F



Although the project may be in Phase E (Operations and Sustainment) and Phase F (Closeout) of the life cycle, it is extremely important to continue to practice good systems engineering and to execute the SE engine. The technical management processes in the SE engine are still being used. During the operations phase of a project, a number of activities are still going on. In addition to the day-to-day use of the product, there is a need to monitor or manage various aspects of the system. This is where the Technical Performance Measures (TPMs) that were defined in the early stages of development continue to play a part. (TPMs are described in Section 6.7.2.) Monitoring these measures ensures that the product continues to perform as designed and expected. Configurations are still under control, still executing the Configuration Management Process. Decisions are still being made using the Decision Analysis Process. Indeed, all of the technical management processes still apply.

For this discussion, the term "systems management" is used for this aspect of operations. Systems management includes monitoring the systems day-to-day, working anomalies as they occur, and working other technical issues associated with the system.

In addition to systems management and systems operation, there may also be a need for periodic refurbishment, repairing broken parts, cleaning, sparing, logistics, or other activities. For this discussion, the term "sustaining engineering" is used for these activities.

Again, all of the technical management processes still apply to these activities. Figure 2.3-9 represents these three activities occurring simultaneously and continuously throughout the operational lifetime of the final product. Some portions of the SE processes need to continue even after the system becomes nonoperational to handle retirement, decommissioning, and disposal. This is consistent with the basic SE principle of handling the full system life cycle from "cradle to grave."

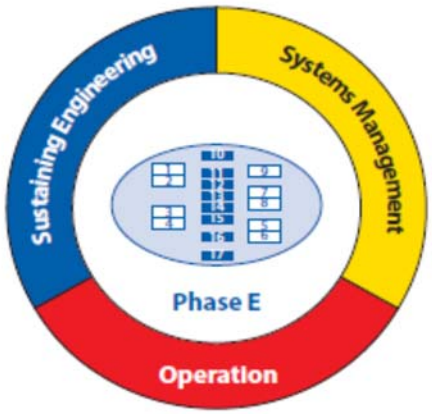


Figure 2.3-9 Model of Typical SE Activities during Operational Phase (Phase E) of a Product

However, if at any point in this phase a new product, a change that affects the design or certification of a product, or an upgrade to an existing product is needed, the development processes of the SE engine are reentered at the beginning. That is, the first thing that is done for an upgrade is to determine who the stakeholders are and what they expect. The entire SE engine is used just as for a newly developed product. This might be pictorially portrayed as in Figure 2.3-10. Note that although the SE engine is shown only once in the figure, it is used recursively down through the product hierarchy for upgraded products, just as described in our detailed example for the initial product.

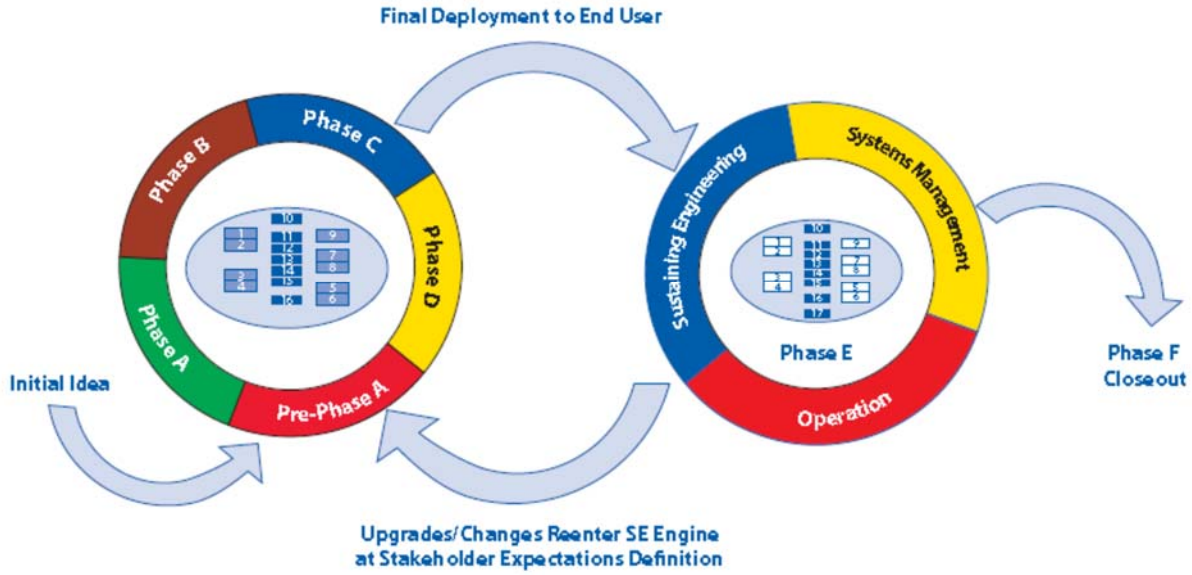


Figure 2.3-10 New Products or Upgrades Reentering the Design Cycle of the SE Engine

2.4 Distinctions between Product Verification and Product Validation

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

- Verification of a product shows proof of compliance with requirements—that the product can meet each "shall" statement as proven though performance of a test, analysis, inspection, or demonstration (or combination of these).
- Validation of a product shows that the product accomplishes the intended purpose in the intended environment—that it meets the expectations of the customer and other stakeholders as shown through performance of a test, analysis, inspection, or demonstration.

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

Validation relates back to the ConOps document. Validation testing is conducted under realistic conditions (or simulated conditions) on end products for the purpose of determining the effectiveness and suitability of the product for use in mission operations by typical users. Validation can be performed in each development phase using phase products (e.g., models) and not only at delivery using end products.

It is appropriate for verification and validation methods to differ between phases as designs advance. The ultimate success of a program or project may relate to the frequency and diligence of validation efforts during the design process, especially in Pre-Phase A and Phase A during which corrections in the direction of product design might still be made cost-effectively. The question should be continually asked, "Are we building the right product for our users and other stakeholders?" The selection of the verification or validation method is based on engineering judgment as to which is the most effective way to reliably show the product's conformance to requirements or that it will operate as intended and described in the ConOps.

2.5 Cost Effectiveness Considerations

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

A cost-effective and safe system should provide a particular kind of balance between effectiveness and cost. This causality is an indefinite one because there are usually many designs that meet the cost-effective condition. Think of each possible design as a point in the tradeoff space between effectiveness and cost. A graph plotting the maximum achievable effectiveness of designs available with current technology as a function of cost would, in general, yield a curved line such as the one shown in Figure 2.5-1. In the figure, all the dimensions of effectiveness are represented by the ordinate (vertical axis) and all the dimensions of cost by the abscissa (horizontal axis). In other words, the curved line represents the envelope of the currently available technology in terms of cost-effectiveness.

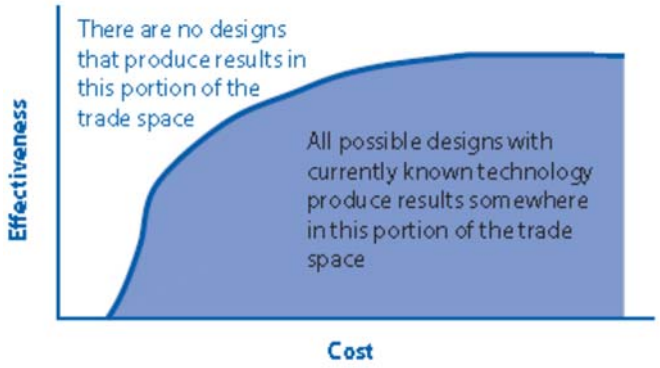


Figure 2.5-1 The Enveloping Surface of Non-dominated Designs

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

System Cost, Effectiveness, and Cost-Effectiveness

- **Cost:** The cost of a system is the value of the resources needed to design, build, operate, and dispose of it. Because resources come in many forms—work performed by NASA personnel and contractors; materials; energy; the use of facilities and equipment such as wind tunnels, factories, offices, information technology, and the cost of user, maintainer, and controller personnel—it is convenient to express these values in common terms by using monetary units (such as dollars of a specified year). Note that cost may not be one-dimensional in terms of value to a program, NASA or the public. For example, even when funding and effectiveness are balanced, it may be more desirable for programmatic (e.g., schedule) reasons to spend funds on hardware instead of software (or vice versa).
- **Effectiveness:** The effectiveness of a system is a quantitative measure of the degree to which the system's purpose is achieved. Effectiveness measures are dependent upon the individual and integrated performance of the system components: hardware, software, and human elements. During development, ongoing validation of a system's ConOps can be a key indicator that the integrated system's planned purpose will be achieved. Note that total mission effectiveness may comprise multiple mission goals or acceptable variants within a goal; i.e., goals for complex missions should include nominal, off-nominal, and contingency scenarios.
- **Cost-Effectiveness:** The cost-effectiveness of a system compares the relative cost and effective outcome of the program's/project's accomplished objectives and available budget.

Design trade studies, an important part of the systems engineering process, often attempt to find designs that provide the best combination of the various dimensions of cost and effectiveness. When the starting point for a design trade study is inside the envelope, there are alternatives that either reduce costs without reducing effectiveness or increase effectiveness without increasing cost (i.e., moving closer to the envelope curve). In such "win-win" cases, the systems engineer's decision is easy. When the alternatives in a design trade study require trading cost for effectiveness, or even one dimension of effectiveness for another at the same cost (i.e., moving parallel to the envelope curve), the decisions become harder.

The process of finding the most cost-effective design is further complicated by uncertainty, which is shown in Figure 2.5-2. Exactly what outcomes will be realized by a particular system design cannot be known in advance with certainty, so the projected cost and effectiveness of a design are better described by a probability distribution than by a point. This distribution can be thought of as a cloud that is thickest at the most likely value and thinnest farthest away from the most likely point, as is shown for design concept A in the figure. Distributions resulting from designs that have little uncertainty are dense and highly compact, as is shown for concept B. Distributions associated with risky designs may have significant probabilities of producing highly undesirable outcomes, as is suggested by the presence of an additional low-effectiveness/high-cost cloud for concept C in addition to the desirable outcome that is represented as the distribution shown on the line. (Of course, the envelope of such clouds cannot be a sharp line such as is shown in the figure, but should itself be rather fuzzy. The line can now be thought of as representing the envelope at some fixed confidence level, that is, a specific, numerical probability of achieving that effectiveness.)

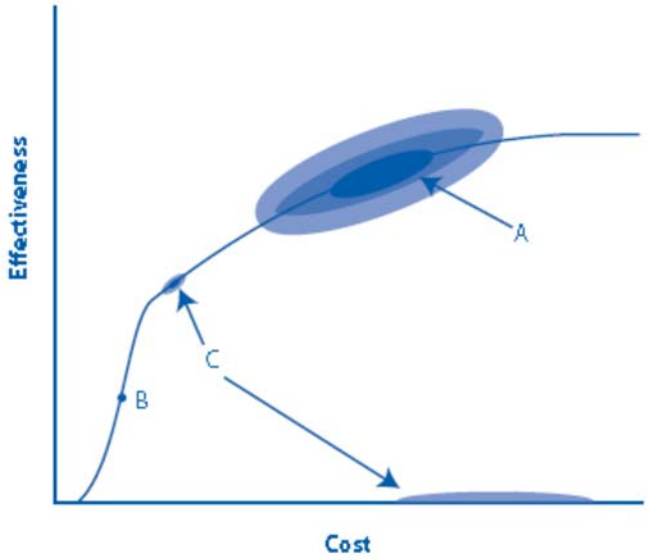


Figure 2.5-2 Estimates of Outcomes to be Obtained from Several Design Concepts including Uncertainty

Note: A, B, and C are design concepts with different risk patterns.

Both effectiveness and cost may consist of several attributes. Even the Echo balloons (circa 1960), in addition to their primary mission as communications satellites, obtained scientific data on the electromagnetic environment and atmospheric drag. Furthermore, Echo was the first satellite visible to the naked eye, an unquantifiable—but not unrecognized at the beginning of the space race—aspect of its effectiveness. Sputnik (circa 1957), for example, drew much of its effectiveness from the fact that it was a "first." Costs, the expenditure of limited resources, may be measured in attributes of funding, personnel, use of facilities, and so on. Schedule may appear as an attribute of effectiveness or cost, or as a constraint. A mission to Mars that misses its launch window has to wait about two years for another opportunity—a clear schedule constraint.

The Systems Engineer's Dilemma

At each cost-effective solution:

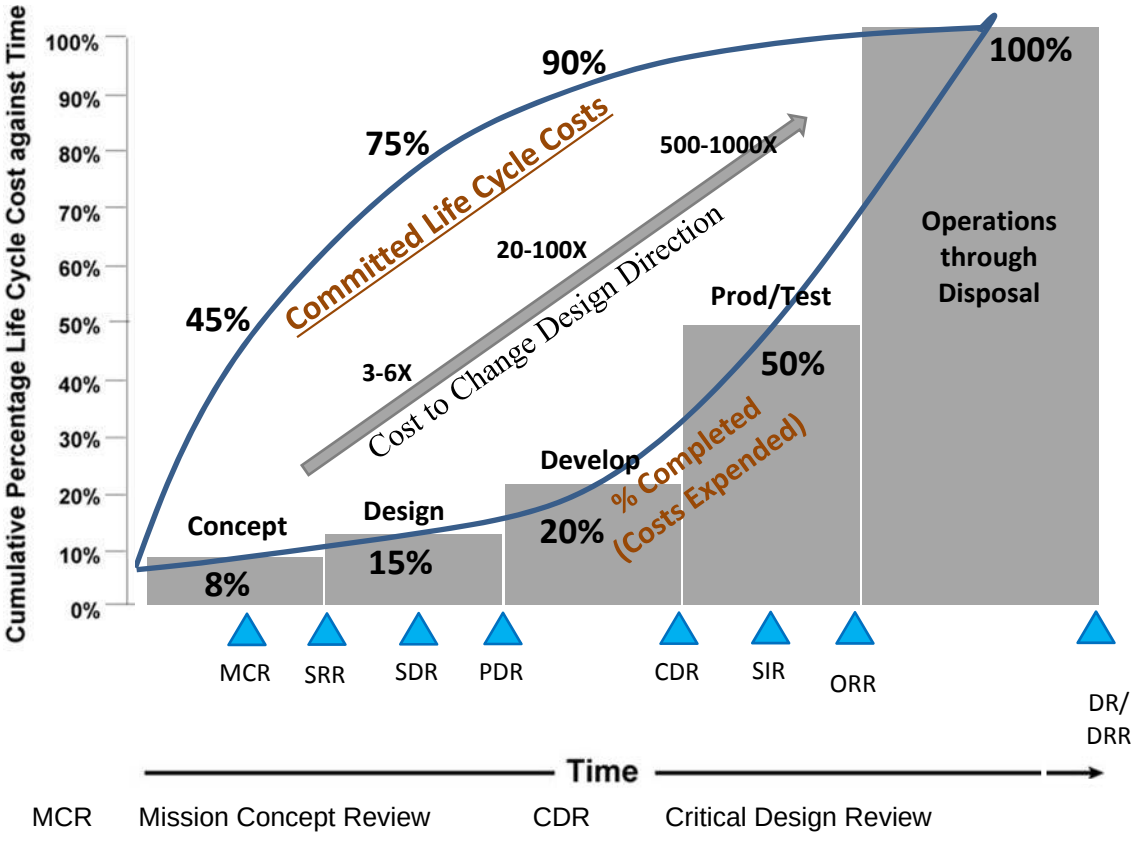
- To reduce cost at constant risk, performance must be reduced.
- To reduce risk at constant cost, performance must be reduced.
- To reduce cost at constant performance, higher risks must be accepted.
- To reduce risk at constant performance, higher costs must be accepted.

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

Note that in this portrayal of cost-effectiveness as an optimized ratio (per Figure 2.5-1), "cost" is the total life-cycle cost of a system.

Often, programs/projects are divided between design and operations phases, with handover of a developed design as the operations phase (Phase E) of a system's life cycle begins. Without continuous and fluid ops-to-development communication, design and development managers may make decisions based solely on design and production cost-effectiveness without fully considering resulting operations costs. Effectively applying HSI processes (see Section 2.6) may help control life-cycle cost by identifying and applying key operations performance parameters early in design and development. Additional forces that can divide system development and operations phases are discussed in Section 7.1, Engineering with Contracts.

Figure 2.5-3 shows that the life-cycle costs of a program or project tend to get "locked in" early in design and development. The cost curves clearly show that late identification of and fixes to problems cost considerably more later in the life cycle. Conversely, descopes taken later versus earlier in the project life cycle result in reduced cost savings. This figure, obtained from the Defense Acquisition University, is an example of how these costs are determined by the early concepts and designs. The numbers will vary from project to project, but the general shape of the curves and the message they send will be similar. For example, the figure shows that during design, only about 15% of the costs might be expended, but the design itself will commit about 75% of the life cycle costs. This is because the way the system is designed will determine how expensive it will be to test, manufacture, integrate, operate, and sustain. If these factors have not been considered during design, they pose significant cost risks later in the lifecycle. Also note that the cost to change the design increases as you get later in the life cycle. If the project waits until verification to do any type of test or analysis, any problems found will have a significant cost impact to redesign and reverify.



SRR	System Requirements Review	SIR	System Integration Review
SDR	System Definition Review	ORR	Operational Readiness Review
PDR	Preliminary Design Review	DR/DRR	Decommissioning/Disposal Readiness Review

Adapted from INCOSE-TP-2003-002-04, 2015

Figure 2.5-3 Life-Cycle Cost Impacts from Early Phase Decision-Making

In some contexts, it is appropriate to seek the most effectiveness possible within a fixed budget and with a fixed risk; in other contexts, it is more appropriate to seek the least cost possible with specified effectiveness and risk. In these cases, there is the question of what level of effectiveness to specify or what level of costs to fix. In practice, these may be mandated in the form of performance or cost requirements. It then becomes appropriate to ask whether a slight relaxation of requirements could produce a significantly cheaper system or whether a few more resources could produce a significantly more effective system. Generally, however, quality systems engineering is always cognizant of the impacts of near-term decisions on the strategic outcome of a project; i.e., on the life-cycle cost implications of any trade decision.

When determining the cost-effectiveness of a system, it is important to consider the operational costs with development costs over the entire system life cycle (section 6.1.2.2). System failure or unavailability can be a significant portion of these life-cycle costs and are an essential consideration to completely characterize the system's life-cycle cost. Technologies and methodologies such as fault management, addressed in greater detail in Section 7.7, can help manage life-cycle costs by reducing downtime, repairs, and the risk of catastrophic failures, and by improving the project payoff during the lifetime of a project.

The technical team may have to choose among designs that differ in terms of numerous attributes. A variety of methods have been developed that can be used to help uncover preferences between attributes and to quantify subjective assessments of relative value. When this can be done, trades between attributes can be assessed quantitatively. Often, however, the attributes are incompatible. In the end, decisions need to be made in spite of the given variety of attributes. There are several decision analysis techniques (section 6.8) that can aid in complex decision analysis. The systems engineer should always keep in mind the information that needs to be available to help the decision-makers choose the most cost-effective option.

2.6 Human Systems Integration (HSI) in the SE Process

As noted at the beginning of NPR 7123.1, the "systems approach is applied to all elements of a system (i.e., hardware, software, human systems integration)," (NPR 7123.1, Preface, paragraph P.1, Purpose). In short, the systems engineering approach must equally address and integrate these three key elements: hardware, software, and human systems integration. Therefore, the human element is something that integration and systems engineering processes must address. The definition of "system" in NPR 7123.1 is inclusive; i.e., a system is "the combination of elements that function together to produce the capability required to meet a need. The elements include all hardware, software, equipment, facilities, personnel, processes, and procedures needed for this purpose. (Refer to NPR 7120.5.)"

Throughout time, engineering has been executed to serve and fulfill human needs. Considering the human contexts in which a system will be developed, delivered, and operated should be a natural part of every systems engineer's thinking.

Human Systems Integration (HSI), a growing field, is intended to ensure the following:

- The system comprises hardware, software, and humans, all of which interact and operate within an environment,
- Consideration of "the human element" (sometimes referred to as "the human as a system") comprises all human interactions with a system—end users (e.g., pilots, passengers), maintainers, ground controllers, logistics personnel, etc.,
- HSI is managed to consider input from all human-related disciplines (e.g., human factors engineering, habitability, survivability, human-centered safety concerns, human-rating, etc.) and to provide integrated human-element inputs to a program's/project's systems engineering and program management processes, and
- HSI is established as a core part of a program's/project's systems engineering and program management processes as early as possible (Pre-Phase A) to ensure the human elements of a system are appropriately considered in the critical path throughout the design and development life cycle of the system.

Figure 2.5-3, Life-Cycle Cost Impacts from Early Phase Decision-Making, in the previous section of this guide applies as equally to human element considerations as to hardware and software concerns; i.e., if all human interactions are not consciously considered as part of a systems' design, early program decisions might result in high and unexpected operations costs. An example of this is when a designer assumes that "we can train the operator to manage our system design" rather than designing a system's operation to be more intuitive for the intended user population.

As defined in NPR 7123.1 and in this guide (appendix B), Human Systems Integration (HSI) is an interdisciplinary and comprehensive management and technical process that focuses on the integration of human capabilities and limitations into the system acquisition and development processes to enhance human system design, reduce life cycle ownership cost, and optimize total system performance. HSI processes are fully integrated with NASA systems engineering. HSI maps consideration of a system's human elements into Requirements Definition, Stakeholder Expectations, Logical Decomposition, Design Realization, and all other processes of the systems

engineering engine outlined in this guide, just as hardware and software considerations are similarly mapped into these processes.

It is recommended program/project practice to plan early for HSI and to develop a programmatic HSI Plan that is either part of the SEMP or a stand-alone subset of the SEMP. A recommended outline for an HSI Plan is included in appendix R of this guide. The outline indicates that the HSI Plan should document planned goals, metrics, standards, deliverables, processes, and in particular, the roles and responsibilities of the parties responsible for a program's/project's HSI implementation. HSI should be managed by system integrators skilled in HSI and human element considerations in design and development. The HSI responsible parties should update the HSI Plan throughout the program/project life cycle to keep it current.

Further information on HSI as part of the systems engineering process can be found in Section 7.9, Human Systems Integration (HSI) in the SE Process, of the Crosscutting Topics chapter of this guide.

2.7 Competency Model for Systems Engineers

Table 2.7-1 provides a summary of the Competency Model for Systems Engineering. One of the most important characteristics of a successful systems engineer is the ability to lead the technical team. Systems engineering is both a science and an art. Following the rules of the NPRs and balancing requirements, capabilities, and budgets are all part of the science. The art of systems engineering includes seeing the big picture, looking beyond the current metrics to see the overall goals and objectives, and leading a technical team of highly capable people.

Technical leadership balances creativity, broad technical knowledge, instinct, and skills in problem-solving and communication to work with members of the technical team and less technical stakeholders to develop new systems or services. A good technical leader can bring out the best in members of the technical team to contribute their individual talents to the overall objective of not just doing the job, but doing the job right.

NASA provides leadership training for systems engineers and other technical experts. Talk to your supervisor about training opportunities including formal classes, on-the-job training, and rotational opportunities.

Table 2.7-1 NASA System Engineering Competency Model

Competency Area	Competency	Description
SE 1.0 System Design	SE 1.1 Stakeholder Expectation Definition & Management	Eliciting and defining use cases, scenarios, concept of operations and stakeholder expectations. This includes identifying stakeholders, establishing support strategies, establishing a set of Measures of Effectiveness (MOEs), validating stakeholder expectation statements, and obtaining commitments from the customer and other stakeholders, as well as using the baselined stakeholder expectations for product validation during product realization
	SE 1.2 Technical Requirements Definition	Transforming the baseline stakeholder expectations into unique, quantitative, and measureable technical requirements expressed as "shall" statements that can be used for defining the design solution. This includes analyzing the scope of the technical problems to be solved, defining constraints affecting the designs, defining the performance requirements, validating the resulting technical requirement statements, defining the Measures of Performance (MOPs) for each MOE, and defining appropriate Technical Performance Measures (TPMs) by which technical progress will be assessed.
	SE 1.3 Logical Decomposition	Transforming the defined set of technical requirements into a set of logical decomposition models and their associated set of derived technical requirements for lower levels of the system, and for input to the design solution efforts. This includes decomposing and analyzing by function, time, behavior, data flow, object, and other models. It also includes allocating requirements to these decomposition models, resolving conflicts between derived requirements as revealed by the models, defining a system architecture for establishing the levels of allocation, and validating the derived technical requirements.
	SE 1.4 Design Solution Definition	Translating the decomposition models and derived requirements into one or more design solutions, and using the Decision Analysis process to analyze each alternative and for selecting a preferred alternative that will satisfy the technical requirements. A full technical data package is developed describing the selected solution. This includes generating a full design description for the selected solution; developing a set of 'make-to,' 'buy-to,' 'reuse-to,' specifications; and initiating the development or acquisition of system products and enabling products.
SE 2.0 Product realization	SE 2.1 Product Implementation	Generating a specific product through buying, making, or reusing so as to satisfy the design requirements. This includes preparing the implementation strategy; building or coding the produce; reviewing vendor technical information; inspecting delivered, built, or reused products; and preparing product support documentation for integration.
	SE 2.2 Product Integration	Assembling and integrating lower-level validated end products into the desired end product of the higher-level product. This includes preparing the product integration strategy, performing detailed planning, obtaining products to integrate, confirming that the products are ready for integration, preparing the integration environment, and preparing product support documentation.
	SE 2.3 Product Verification	Proving the end product conforms to its requirements. This includes preparing for the verification efforts, analyzing the outcomes of verification (including identifying anomalies and establishing recommended corrective actions), and preparing a product verification report providing the evidence of product conformance with the applicable requirements.
	SE 2.4 Product Validation	Confirming that a verified end product satisfies the stakeholder expectations for its intended use when placed in its intended environment and ensuring that any anomalies discovered during validation are appropriately resolved prior to product transition. This includes preparing to conduct product validation, performing the product validation, analyzing the results of validation (including identifying anomalies and establishing recommended corrective actions), and preparing a product validation report providing the evidence of product conformance with the stakeholder expectations baseline.

Competency Area	Competency	Description
	SE 2.5 Product Transition	Transitioning the verified and validated product to the customer at the next level in the system structure. This includes preparing to conduct product transition, evaluating the product and enabling product readiness for product transition, preparing the product for transition (including handling, storing, and shipping preparation), preparing sites, and generating required documentation to accompany the product
SE 3.0 Technical Management	SE 3.1 Technical Planning	Planning for the application and management of each common technical process, as well as identifying, defining, and planning the technical effort necessary to meet project objectives. This includes preparing or updating a planning strategy for each of the technical processes, and determining deliverable work products from technical efforts; identifying technical reporting requirements; identifying entry and success criteria for technical reviews; identifying product and process measures to be used; identifying critical technical events; defining cross domain interoperability and collaboration needs; defining the data management approach; identifying the technical risks to be addressed in the planning effort; identifying tools and engineering methods to be employed; and defining the approach to acquire and maintain technical expertise needed. This also includes preparing the Systems Engineering Management Plan (SEMP) and other technical plans; obtaining stakeholder commitments to the technical plans; and issuing authorized technical work directives to implement the technical work
	SE 3.2 Requirements Management	Managing the product requirements, including providing bidirectional traceability, and managing changes to establish requirement baselines over the life cycle of the system products. This includes preparing or updating a strategy for requirements management; selecting an appropriate requirements management tool; training technical team members in established requirement management procedures; conducting expectation and requirements traceability audits; managing expectation and requirement changes; and communicating expectation and requirement change information
	SE 3.3 Interface Management	Establishing and using formal interface management to maintain internal and external interface definition and compliance among the end products and enabling products. This includes preparing interface management procedures, identifying interfaces, generating and maintaining interface documentation, managing changes to interfaces, disseminating interface information, and conducting interface control
	SE 3.4 Technical Risk Management	Examining on a continual basis the risks of technical deviations from the plans, and identifying potential technical problems before they occur. Planning, invoking, and performing risk-handling activities as needed across the life of the product or project to mitigate impacts on meeting technical objectives. This includes developing the strategy for technical risk management, identifying technical risks, and conducting technical risk assessment; preparing for technical risk mitigation, monitoring the status of each technical risk, and implementing technical risk mitigation and contingency action plans when applicable thresholds have been triggered.
	SE 3.5 Configuration Management	Identifying the configuration of the product at various points in time, systematically controlling changes to the configuration of the product, maintaining the integrity and traceability of product configuration, and preserving the records of the product configuration throughout its life cycle. This includes establishing configuration management strategies and policies, identifying baselines to be under configuration control, maintaining the status of configuration documentation, and conducting configuration audits

Competency Area	Competency	Description
	SE 3.6 Technical Data Management	Identifying and controlling product-related data throughout its life cycle; acquiring, accessing, and distributing data needed to develop, manage, operate, support, and retire system products; managing and disposing data as records; analyzing data use; obtaining technical data feedback for managing the contracted technical efforts; assessing the collection of appropriate technical data and information; maintaining the integrity and security of the technical data, effectively managing authoritative data that defines, describes, analyzes, and characterizes a product life cycle; and ensuring consistent, repeatable use of effective Product Data and Life-cycle Management processes, best practices, interoperability approaches, methodologies, and traceability. This includes establishing technical data management strategies and policies; maintaining revision, status, and history of stored technical data and associated metadata; providing approved, published technical data; providing technical data to authorized parties; and collecting and storing required technical data.
	SE 3.7 Technical Assessment	Monitoring progress of the technical effort and providing status information for support of the system design, product realization, and technical management efforts. This includes developing technical assessment strategies and policies, assessing technical work productivity, assessing product quality, tracking and trending technical metrics, and conducting technical, peer, and life cycle reviews.
	SE 3.8 Technical Decision Analysis	Evaluating technical decision issues, identifying decision criteria, identifying alternatives, analyzing alternatives, and selecting alternatives. Performed throughout the system life cycle to formulate candidate decision alternatives, and evaluate their impacts on health and safety, technical, cost, and schedule performance. This includes establishing guidelines for determining which technical issues are subject to formal analysis processes; defining the criteria for evaluating alternative solutions; identifying alternative solutions to address decision issues; selecting evaluation methods; selecting recommended solutions; and reporting the results and findings with recommendations, impacts, and corrective actions.

There are four levels of proficiencies associated with each of these competencies:

- Team Practitioner/Technical Engineer
- Team Lead/Subsystem Lead
- Project Systems Engineer
- Chief Engineer

3.0 NASA Program/Project Life Cycle

One of the fundamental concepts used within NASA for the management of major systems is the program/project life cycle, which categorizes everything that should be done to accomplish a program or project into distinct phases that are separated by Key Decision Points (KDPs). *KDPs are the events at which the decision authority determines the readiness of a program/project to progress to the next phase of the life cycle (or to the next KDP).* Phase boundaries are defined so that they provide natural points for "go" or "no-go" decisions. Decisions to proceed may be qualified by liens that should be removed within an agreed-to time period. A program or project that fails to pass a KDP may be allowed to try again later after addressing deficiencies that precluded passing the KDP, or it may be terminated.

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

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

For NASA projects, the life cycle is defined in the applicable governing document:

- For spaceflight projects: NPR 7120.5, NASA Space Flight Program and Project Management Requirements
- For information technology: NPR 7120.7, NASA Information Technology and Institutional Infrastructure Program and Project Management Requirements
- For NASA research and technology: NPR 7120.8, NASA Research and Technology Program and Project Management Requirements
- For software: NPR 7150.2 NASA Software Engineering Requirements

For example, NPR 7120.5 defines the major NASA life-cycle phases as Formulation and Implementation. For space flight systems projects, the NASA life-cycle phases of Formulation and Implementation divide into the following seven incremental pieces. The phases of the project life cycle are:

Program Pre-Formulation:

- Pre-Phase A: Concept Studies

Program Formulation

- Phase A: Concept and Technology Development
- Phase B: Preliminary Design and Technology Completion

Program Implementation:

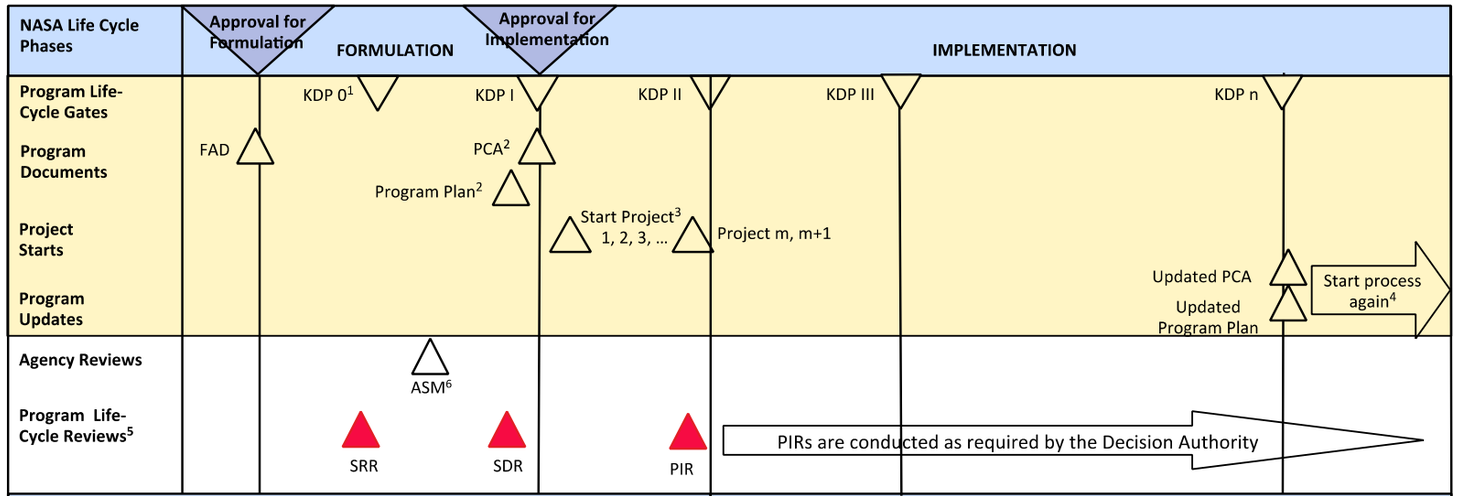
- Phase C: Final Design and Fabrication
- Phase D: System Assembly, Integration and Test, Launch
- Phase E: Operations and Sustainment
- Phase F: Closeout

Similar phases for IT projects are discussed in the Office of the Chief Information Officer's (OCIO) *IT Systems Engineering Handbook*. NASA distinguishes between highly specialized IT and IT that is not highly specialized. Highly specialized IT is defined in Appendix B, Glossary, and is subject to NPR 7120.5, NASA Space Flight Program and Project Management Requirements, or to NPR 7120.8, NASA Research and Technology Program and Project Management Requirements, depending on the program or project content.

Figure 3.0-1, 3.0-2 and 3.0-3 (NASA space-flight program life cycles) and Figure 3.0-4 (NASA project life cycle) identify the KDPs and reviews that characterize the phases. Sections 3.1 and 3.2 contain narrative descriptions of the purposes, major activities, products, and KDPs of the NASA program life-cycle phases. Sections 3.3 to 3.9 contain narrative descriptions of the purposes, major activities, products, and KDPs of the NASA project life-cycle phases. Section 3.10 describes the NASA budget cycle within which program/project managers and systems engineers should operate.

Additional information on life cycles can be found in *SP-2014-3705, NASA Space Flight Program and Project Management Handbook*.

Table 3.0-1 is taken from NPR 7123.1 and represents the product maturity for the major SE products developed and matured during the product life cycle.



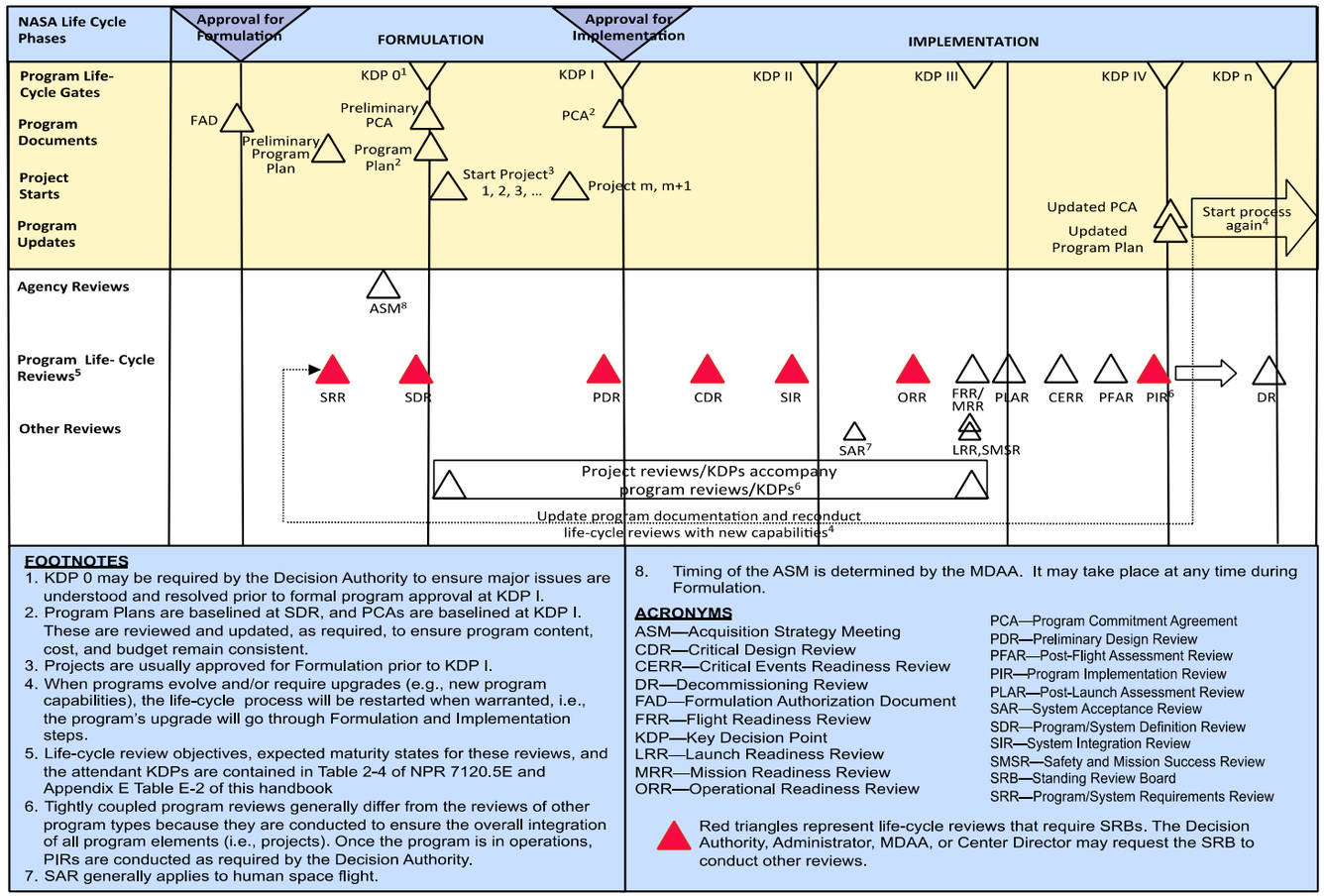
FOOTNOTES

1. KDP 0 may be required by the Decision Authority to ensure major issues are understood and resolved prior to formal program approval at KDP I.
2. Program Plans are baselined at SDR, and PCAs are baselined at KDP I. These are reviewed and updated, as required, to ensure program content, cost, and budget remain consistent.
3. Projects, in some instances, may be approved for Formulation prior to KDP I. Initial project pre-Formulation generally occurs during program Formulation.
4. When programs evolve and/or require upgrades (e.g., new program capabilities), the life-cycle process will be restarted when warranted, i.e., the program's upgrade will go through Formulation and Implementation steps.
5. Life-cycle review objectives and expected maturity states for these reviews and the attendant KDPs are contained in Table 2-3 of NPR 7120.5E and Appendix E Table E-1 of this handbook
6. Timing of the ASM is determined by the MDAA. It may take place at any time during Formulation.

ACRONYMS

ASM—Acquisition Strategy Meeting
FAD—Formulation Authorization Document
KDP—Key Decision Point
PCA—Program Commitment Agreement
PIR—Program Implementation Review
SDR—System Definition Review
SRB—Standing Review Board
SRR—System Requirements Review

Red triangles represent life-cycle reviews that require SRBs. The Decision Authority, Administrator, MDAA, or Center Director may request the SRB conduct other reviews.



FOOTNOTES

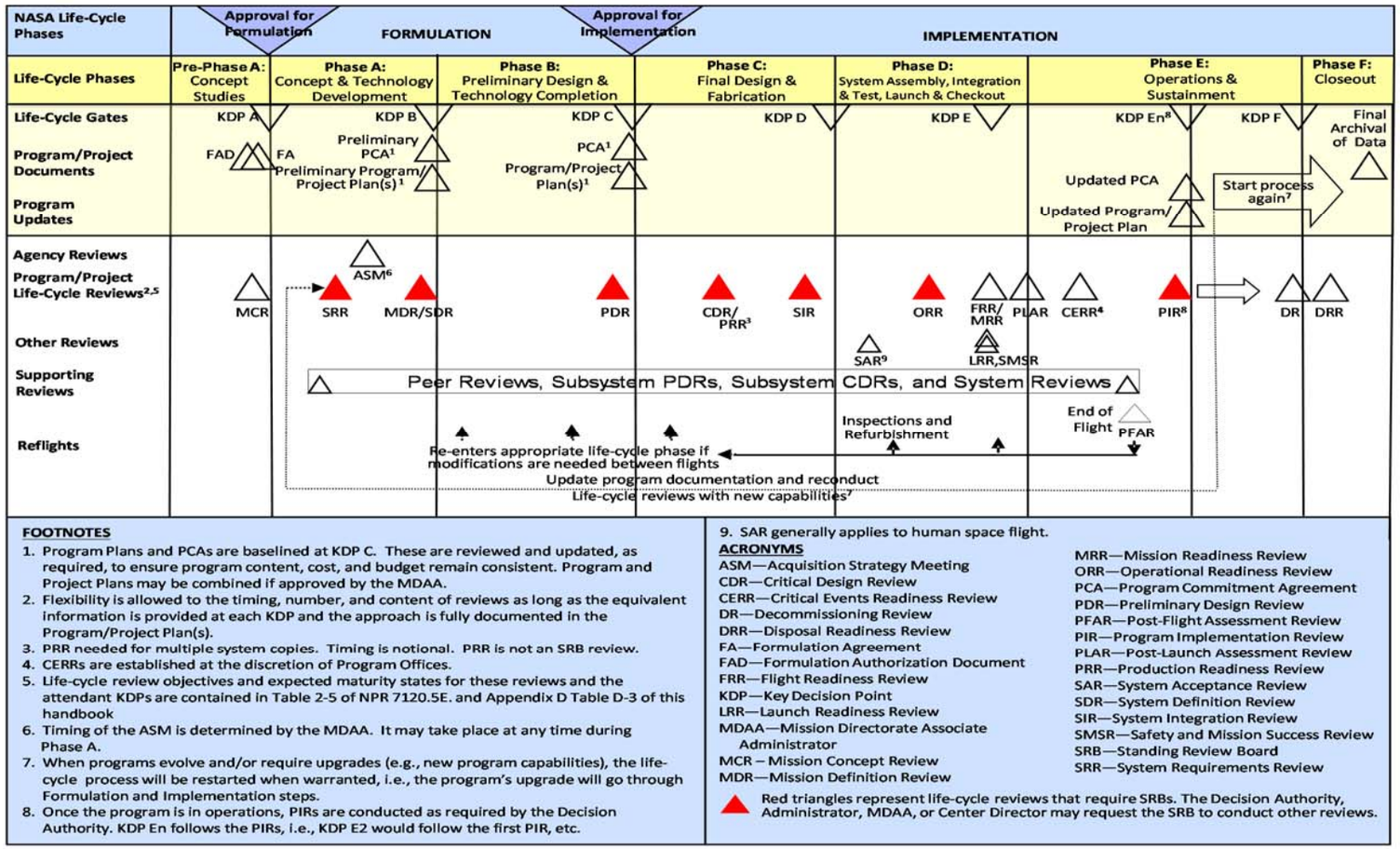
1. KDP 0 may be required by the Decision Authority to ensure major issues are understood and resolved prior to formal program approval at KDP I.
2. Program Plans are baselined at SDR, and PCAs are baselined at KDP I. These are reviewed and updated, as required, to ensure program content, cost, and budget remain consistent.
3. Projects are usually approved for Formulation prior to KDP I.
4. When programs evolve and/or require upgrades (e.g., new program capabilities), the life-cycle process will be restarted when warranted, i.e., the program's upgrade will go through Formulation and Implementation steps.
5. Life-cycle review objectives, expected maturity states for these reviews, and the attendant KDPs are contained in Table 2-4 of NPR 7120.5E and Appendix E Table E-2 of this handbook
6. Tightly coupled program reviews generally differ from the reviews of other program types because they are conducted to ensure the overall integration of all program elements (i.e., projects). Once the program is in operations, PIRs are conducted as required by the Decision Authority.
7. SAR generally applies to human space flight.
8. Timing of the ASM is determined by the MDAA. It may take place at any time during Formulation.

ACRONYMS

ASM—Acquisition Strategy Meeting
CDR—Critical Design Review
CERR—Critical Events Readiness Review
DR—Decommissioning Review
FAD—Formulation Authorization Document
FRR—Flight Readiness Review
KDP—Key Decision Point
LRR—Launch Readiness Review
MRR—Mission Readiness Review
ORR—Operational Readiness Review
PCA—Program Commitment Agreement
PDR—Preliminary Design Review
PFAR—Post-Flight Assessment Review
PIR—Program Implementation Review
PLAR—Post-Launch Assessment Review
SAR—System Acceptance Review
SDR—Program/System Definition Review
SIR—System Integration Review
SMSR—Safety and Mission Success Review
SRB—Standing Review Board
SRR—Program/System Requirements Review

Red triangles represent life-cycle reviews that require SRBs. The Decision Authority, Administrator, MDAA, or Center Director may request the SRB to conduct other reviews.

Figure 3.0-2 NASA Tightly Coupled Program Life Cycle



FOOTNOTES

1. Program Plans and PCAs are baselined at KDP C. These are reviewed and updated, as required, to ensure program content, cost, and budget remain consistent. Program and Project Plans may be combined if approved by the MDAA.
2. Flexibility is allowed to the timing, number, and content of reviews as long as the equivalent information is provided at each KDP and the approach is fully documented in the Program/Project Plan(s).
3. PRR needed for multiple system copies. Timing is notional. PRR is not an SRB review.
4. CERRs are established at the discretion of Program Offices.
5. Life-cycle review objectives and expected maturity states for these reviews and the attendant KDPs are contained in Table 2-5 of NPR 7120.5E. and Appendix D Table D-3 of this handbook
6. Timing of the ASM is determined by the MDAA. It may take place at any time during Phase A.
7. When programs evolve and/or require upgrades (e.g., new program capabilities), the life-cycle process will be restarted when warranted, i.e., the program's upgrade will go through Formulation and Implementation steps.
8. Once the program is in operations, PIRs are conducted as required by the Decision Authority. KDP En follows the PIRs, i.e., KDP E2 would follow the first PIR, etc.
9. SAR generally applies to human space flight.

ACRONYMS

ASM—Acquisition Strategy Meeting
CDR—Critical Design Review
CERR—Critical Events Readiness Review
DR—Decommissioning Review
DRR—Disposal Readiness Review
FA—Formulation Agreement
FAD—Formulation Authorization Document
FRR—Flight Readiness Review
KDP—Key Decision Point
LRR—Launch Readiness Review
MDAA—Mission Directorate Associate Administrator
MCR – Mission Concept Review
MDR—Mission Definition Review
MRR—Mission Readiness Review
ORR—Operational Readiness Review
PCA—Program Commitment Agreement
PDR—Preliminary Design Review
PFAR—Post-Flight Assessment Review
PIR—Program Implementation Review
PLAR—Post-Launch Assessment Review
PRR—Production Readiness Review
SAR—System Acceptance Review
SDR—System Definition Review
SIR—System Integration Review
SMSR—Safety and Mission Success Review
SRB—Standing Review Board
SRR—System Requirements Review

▲ Red triangles represent life-cycle reviews that require SRBs. The Decision Authority, Administrator, MDAA, or Center Director may request the SRB to conduct other reviews.

Figure 3.0-3 NASA Single-Project Program Life Cycle

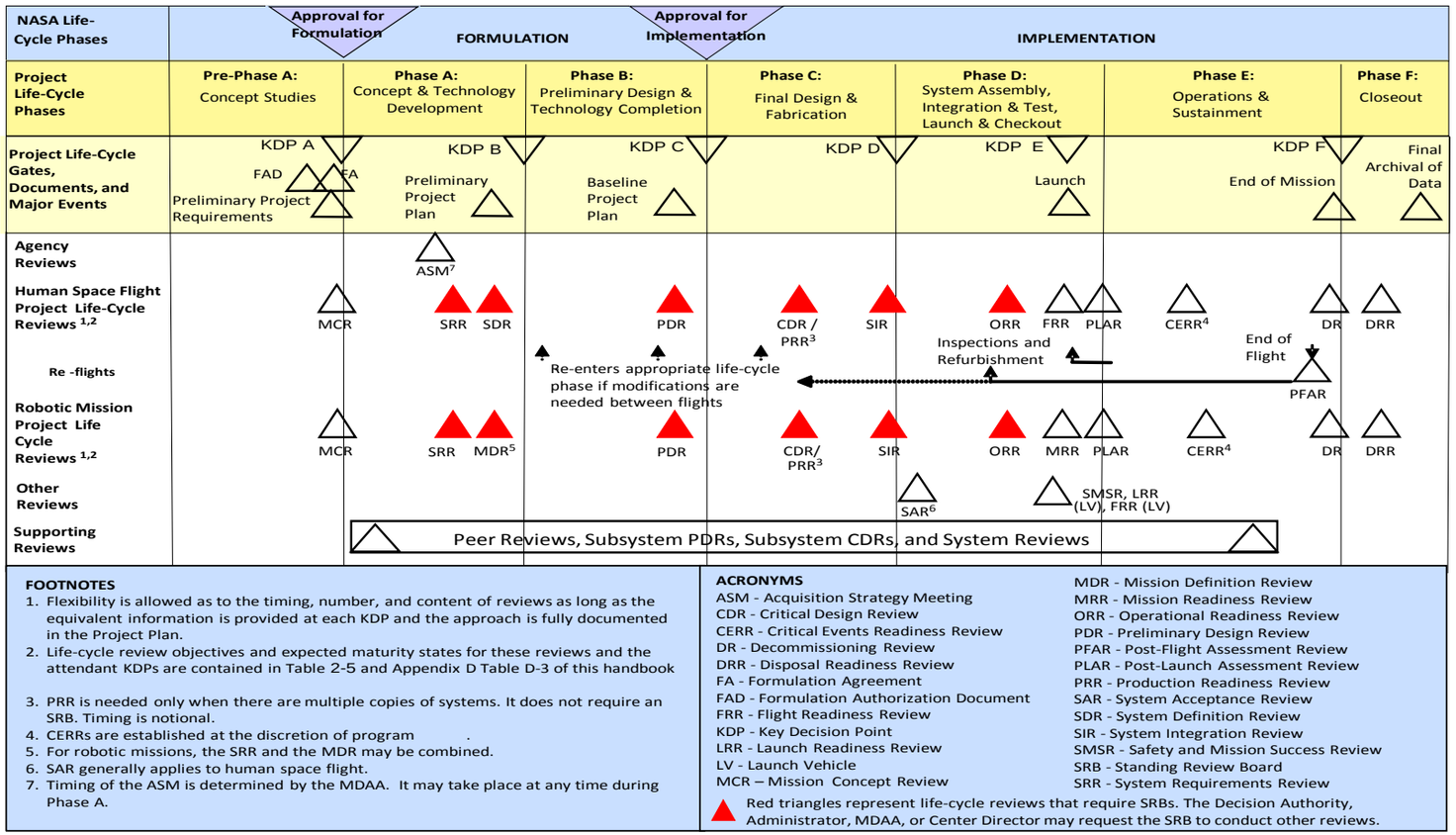


Figure 3.0-4 NASA Project Life Cycle

Table 3.0-1 SE Product Maturity

Products		Formulation				Implementation					
	Uncoupled/ loosely Coupled	KDP 0	KDP I	Periodic KDPs							
	Tightly Coupled Programs	KDP 0		KDP I	KDP II		KDP III		Periodic KDPs		
	Projects and single Project Programs	Pre-Phase A	Phase A		Phase B	Phase C		Phase D		Phase E	Phase F
		KDP A	KDP B		KDP C	KDP D		KDP E		KDP F	
		MCR	SRR	MDR/SDR	PDR	CDR	SIR	ORR	FRR	DR	DRR
Stakeholder identification and		**Baseline	Update	Update	Update						
Concept definition		**Baseline	Update	Update	Update	Update					
Measure of Effectiveness definition		**Approve									
Cost and schedule for technical		Initial	Update	Update	Baseline	Update	Update	Update	Update	Update	Update
SEMP		Preliminary	**Baseline[1]	**Baseline[1]	Update	Update	Update				
Requirements		Preliminary	**Baseline	Update	Update	Update					
Technical Performance Measures definition				**Approve							
Architecture definition				**Baseline							
Allocation of requirements to next lower level				**Baseline							
Required leading indicator trends				**Initial	Update	Update	Update				
Design solution definition				Preliminary	**Preliminary	**Baseline	Update	Update			
Interface definition(s)				Preliminary	Baseline	Update	Update				
Implementation plans (Make/code, buy, reuse)				Preliminary	Baseline	Update					
Integration plans				Preliminary	Baseline	Update	**Update				
Verification and Validation plans		Approach		Preliminary	Baseline	Update	Update				
Verification and Validation results							**Initial	**Preliminary	**Baseline		
Transportation criteria and instructions						Initial	Final	Update			
Operations plans					Baseline	Update	Update	**Update			
Operational procedures						Preliminary	Baseline	**Update	Update		
Certification (flight/use)								Preliminary	**Final		
Decommissioning plans					Preliminary	Preliminary	Preliminary	**Baseline	Update	**Update	
Disposal plans					Preliminary	Preliminary	Preliminary	**Baseline	Update	Update	**Update

**Item is a required product for that review

[1]SEMP is Baselined at SRR for projects, tightly coupled programs and single -project programs, and at MDR/SDR for uncoupled, and loosely coupled programs

3.1 Program Formulation

The program Formulation Phase establishes a cost-effective program that is demonstrably capable of meeting Agency and mission directorate goals and objectives. The program Formulation Authorization Document (FAD) authorizes a Program Manager (PM) to initiate the planning of a new program and to perform the analyses required to formulate a sound program plan. The lead systems engineer provides the technical planning and concept development or this phase of the program life cycle. Planning includes identifying the major technical reviews that are needed and associated entrance and exit criteria. Major reviews leading to approval at KDP I are the SRR, SDR, PDR, and governing Program Management Council (PMC) review. (See a full list of reviews in the program and project life-cycle figures in Section 3.0.) A summary of the required gate products for the program Formulation Phase can be found in the governing NASA directive (e.g., NPR 7120.5 for space flight programs, NPR 7120.7 for IT projects, NPR 7120.8 for research and technology projects). Formulation for all program types is the same, involving one or more program reviews followed by KDP I where a decision is made approving a program to begin implementation.

Space Flight Program Formulation

Purpose

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

Typical Activities and Their Products for Space Flight Programs

- Identify program stakeholders and users
- Develop program requirements based on user expectations and allocate them to initial projects
- Identify NASA risk classification
- Define and approve program acquisition strategies
- Develop interfaces to other programs
- Start developing technologies that cut across multiple projects within the program
- Derive initial cost estimates and approve a program budget based on the project's life-cycle costs
- Perform required program Formulation technical activities defined in NPR 7120.5
- Satisfy program Formulation reviews' entrance/success criteria detailed in NPR 7123.1
- Develop a clear vision of the program's benefits and usage in the operational era and document it in a ConOps

Reviews

- MCR (pre-Formulation)
- SRR
- SDR

3.2 Program Implementation

During the program Implementation phase, the PM works with the Mission Directorate Associate Administrator (MDAA) and the constituent project managers to execute the program plan cost-effectively. Program reviews ensure that the program continues to contribute to Agency and mission directorate goals and objectives within funding constraints. A summary of the required gate products for the program Implementation Phase can be found in the governing NASA directive; e.g., NPR 7120.5 for space-flight programs. The program life cycle has two different implementation paths, depending on program type. Each implementation path has different types of major reviews. It is important for the systems engineer to know what type of program a project falls under so that the appropriate scope of the technical work, documentation requirements, and set of reviews can be determined.

Types of Space Flight Programs

Uncoupled Program - Programs implemented under a broad theme and/or a common program implementation concept, such as providing frequent flight opportunities for cost-capped projects selected through Announcement(s) of Opportunity (AO) or NASA Research Announcements. Each such project is independent of the other projects within the program

Loosely Coupled Program – Programs that address specific objectives through multiple space flight projects of varied scope. While each individual project has an assigned set of mission objectives, architectural and technological synergies and strategies that benefit the program as a whole are explored during the Formulation process. For instance, Mars orbiters designed for more than one Mars year in orbit are required to carry a communication system to support present and future landers.

Tightly Coupled Program - Programs with multiple projects that execute portions of a mission or missions. No single project is capable of implementing a complete mission. Typically, multiple NASA Centers contribute to the program. Individual projects may be managed at different Centers. The program may also include contributions from other agencies or international partners.

Single-Project Program - Programs that tend to have long development and/or operational lifetimes represent a large investment of Agency resources and have contributions from multiple organizations/agencies. These programs frequently combine program and project management approaches, which they document through tailoring.

For uncoupled and loosely coupled programs (see Figure 3.0-1), the Implementation Phase only requires Program Status Reviews (PSRs) and Program Implementation Reviews (PIRs) to assess the program's performance and make a recommendation on its authorization at KDPs approximately every two years. Single-project and tightly coupled programs are more complex. For single-project programs, the Implementation Phase program reviews shown in Figure 3.0-3 are synonymous (not duplicative) with the project reviews in the project life cycle (see Figure 3.0-4) through Phase D. Once in operations, these programs usually have biennial KDPs preceded by attendant PSRs/PIRs. Tightly coupled programs (see Figure 3.0-2) have program reviews tied to the project reviews during implementation to ensure the proper integration of projects into the larger system. Once in operations, tightly coupled programs also have biennial PSRs/PIRs/KDPs to assess the program's performance and authorize its continuation.

Space Flight Program Implementation

Purpose

To execute the program and constituent projects and ensure that the program continues to contribute to Agency goals and objectives within funding constraints

Typical Activities and Their Products

- Initiate projects through direct assignment or competitive process (e.g., Request for Proposal (RFP), Announcement of Opportunity (AO)
- Monitor project's formulation, approval, implementation, integration, operation, and ultimate decommissioning
- Adjust program as resources and requirements change
- Perform required program Implementation technical activities from NPR 7120.5
- Satisfy program Implementation reviews' entrance/ success criteria from NPR 7123.1

Reviews

- PSR/PIR (uncoupled and loosely coupled programs only)
- Reviews synonymous (not duplicative) with the project reviews in the project life cycle (see figure 3.0-4) through Phase D (single-project and tightly coupled programs only)

3.3 Project Pre-Phase A: Concept Studies

The purpose of Pre-Phase A is to produce a broad spectrum of ideas and alternatives for missions from which new programs/projects can be selected. During Pre-Phase A, a study or proposal team analyses a broad range of mission concepts that can fall within technical, cost, and schedule constraints and that contribute to program and Mission Directorate goals and objectives. Pre-Phase A effort could include focused examinations on high-risk or high technology development areas. These advanced studies, along with interactions with customers and other potential stakeholders, help the team to identify promising mission concept(s). The key stakeholders (including the customer) are determined and expectations for the project are gathered from them. If feasible concepts can be found, one or more may be selected to go into Phase A for further development. Typically, the system engineers are heavily involved in the development and assessment of the concept options.

In projects governed by NPR 7120.5, the descope options define what the system can accomplish if the resources are not available to accomplish the entire mission. This could be in the form of fewer instruments, a less ambitious mission profile, accomplishing only a few goals, or using cheaper, less capable technology. Descope options can also reflect what the mission can accomplish in case a hardware failure results in the loss of a portion of the spacecraft architecture; for example, what an orbiter can accomplish after the loss of a lander. The success criteria are reduced to correspond with a descoped mission.

Descope options are developed when the NGOs or other stakeholder expectation documentation is developed. The project team develops a preliminary set of mission descope options as a gate product for the MCR, but these preliminary descope options are not baselined or maintained. They are kept in the documentation archive in case they are needed later in the life cycle.

It is important in Pre-Phase A to define an accurate group of stakeholders and users to help ensure that mission goals and operations concepts meet the needs and expectations of the end users. In addition, it is important to estimate the composition of the technical team and identify any unique facility or personnel requirements.

Advanced studies may extend for several years and are typically focused on establishing mission goals and formulating top-level system requirements and ConOps. Conceptual designs may be developed to demonstrate feasibility and support programmatic estimates. The emphasis is on establishing feasibility and desirability rather than optimality. Analyses and designs are accordingly limited in both depth and number of options, but each option should be evaluated for its implications through the full life cycle, i.e., through Operations and Disposal. It is important in Pre-Phase A to develop and mature a clear vision of what problems the proposed program will address, how it will address them, and how the solution will be feasible and cost-effective.

Space Flight Pre-Phase A: Concept Studies

Purpose

To produce a broad spectrum of ideas and alternatives for missions from which new programs and projects can be selected. Determine feasibility of desired system; develop mission concepts; draft system-level requirements; assess performance, cost, and schedule feasibility; identify potential technology needs and scope.

Typical Activities and Products

- Review/identify any initial customer requirements or scope of work, which may include:
  - Mission
  - Science
  - Top-level system
- Identify and involve users and other stakeholders
  - Identify key stakeholders for each phase of the life cycle
  - Capture and baseline expectations as Needs, Goals, and Objectives (NGOs)
  - Define measures of effectiveness
- Develop and baseline the Concept of Operations
  - Identify and perform tradeoffs and analyses of alternatives (AoA)
  - Perform preliminary evaluations of possible missions
- Identify risk classification
- Identify initial technical risks
- Identify the roles and responsibilities in performing mission objectives (i.e., technical team, flight, and ground crew) including training
- Develop plans
  - Develop preliminary SEMP
  - Develop and baseline Technology Development Plan
  - Define preliminary verification and validation approach
- Prepare program/project proposals, which may include:
  - Mission justification and objectives;
  - A ConOps that exhibits clear understanding of how the program's outcomes will cost-effectively satisfy mission objectives;
  - High-level Work Breakdown Structures (WBSs);
  - Life-cycle rough order of magnitude (ROM) cost, schedule, and risk estimates; and
  - Technology assessment and maturation strategies.
- Satisfy MCR entrance/success criteria from NPR 7123.1

Reviews

- MCR
- Informal proposal review

3.4 Project Phase A: Concept and Technology Development

The purpose of Phase A is to develop a proposed mission/system architecture that is credible and responsive to program expectations, requirements, and constraints on the project, including resources. During Phase A, activities are performed to fully develop a baseline mission concept, begin or assume responsibility for the development of needed technologies, and clarify expected reliance on human elements to achieve full system functionality or autonomous system development. This work, along with interactions with stakeholders, helps mature the mission concept and the program requirements on the project. Systems engineers are heavily involved during this phase in the development and assessment of the architecture and the allocation of requirements to the architecture elements.

In Phase A, a team—often associated with a program or informal project office—readdresses the mission concept first developed in Pre-Phase A to ensure that the project justification and practicality are sufficient to warrant a place in NASA's budget. The team's effort focuses on analyzing mission requirements and establishing a mission architecture. Activities become formal, and the emphasis shifts toward optimizing the concept design. The effort addresses more depth and considers many alternatives. Goals and objectives are solidified, and the project develops more definition in the system requirements, top-level system architecture, and ConOps. Conceptual designs and analyses (including engineering units and physical models, as appropriate) are developed and exhibit more engineering detail than in Pre-Phase A. Technical risks are identified in more detail, and technology development needs become focused. A Systems Engineering Management Plan (SEMP) is baselined in Phase A to document how NASA systems engineering requirements and practices of NPR 7123.1 will be addressed throughout the program life cycle.

In Phase A, the effort focuses on allocating functions to particular items of hardware, software, and to humans. System functional and performance requirements, along with architectures and designs, become firm as system tradeoffs and subsystem tradeoffs iterate back and forth, while collaborating with subject matter experts in the effort to seek out more cost-effective designs. A method of determining life-cycle cost (i.e., system-level cost-effectiveness model) is refined in order to compare cost impacts for each of the different alternatives. (Trade studies should precede—rather than follow— system design decisions.) Major products to this point include an accepted functional baseline for the system and its major end items. The project team conducts the security categorization of IT systems required by NPR 2810.1 and Federal Information Processing Standard Publication (FIPS PUB) 199. The effort also produces various engineering and management plans to prepare for managing the project's downstream processes such as verification and operations.

Space Flight Phase A: Concept and Technology Development

Purpose

To determine the feasibility and desirability of a suggested new system and establish an initial baseline compatibility with NASA's strategic plans. Develop final mission concept, system-level requirements, needed system technology developments, and program/project technical management plans.

Typical Activities and Their Products

- Review and update documents baselined in Pre-Phase A if needed
- Monitor progress against plans
- Develop and baseline top-level requirements and constraints including internal and external interfaces, integrated logistics and maintenance support, and system software functionality
- Allocate system requirements to functions and to next lower level
- Validate requirements
- Baseline plans
  - Systems Engineering Management Plan
  - Human Systems Integration Plan
  - Control plans such as the Risk Management Plan, Configuration Management Plan, Data Management Plan, Safety and Mission Assurance Plan, and Software Development or Management Plan (See NPR 7150.2)
  - Other cross-cutting and specialty plans such as environmental compliance documentation, acquisition surveillance plan, contamination control plan, electromagnetic interference/electromagnetic compatibility control plan, reliability plan, quality control plan, parts management plan, logistics plan
- Develop preliminary Verification and Validation Plan
- Establish human rating plan and perform initial evaluations
- Develop and baseline mission architecture
  - Develop breadboards, engineering units or models identify and reduce high risk concepts
  - Demonstrate that credible, feasible design(s) exist
  - Perform and archive trade studies
  - Initiate studies on human systems interactions
- Initiate environmental evaluation/National Environmental Policy Act process
- Develop initial orbital debris assessment (NASA-STD-8719.14)
- Perform technical management
  - Provide technical cost estimate and range and develop system-level cost-effectiveness model
  - Define the WBS
  - Develop SOWs
  - Acquire systems engineering tools and models
  - Establish technical resource estimates
- Identify, analyze and update risks
- Perform required Phase A technical activities from NPR 7120.5 as applicable
- Satisfy Phase A reviews' entrance/success criteria from NPR 7123.1

Reviews

- SRR
- MDR/SDR

3.5 Project Phase B: Preliminary Design and Technology Completion

The purpose of Phase B is for the project team to complete the technology development, engineering prototyping, heritage hardware and software assessments, and other risk-mitigation activities identified in the project Formulation Agreement (FA) and the preliminary design. The project demonstrates that its planning, technical, cost, and schedule baselines developed during Formulation are complete and consistent; that the preliminary design complies with its requirements; that the project is sufficiently mature to begin Phase C; and that the cost and schedule are adequate to enable mission success with acceptable risk. It is at the conclusion of this phase that the project and the Agency commit to accomplishing the project's objectives for a given cost and schedule. For projects with a Life-Cycle Cost (LCC) greater than $250 million, this commitment is made with the Congress and the U.S. Office of Management and Budget (OMB). This external commitment is the Agency Baseline Commitment (ABC). Systems engineers are involved in this phase to ensure the preliminary designs of the various systems will work together, are compatible, and are likely to meet the customer expectations and applicable requirements.

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

It is important in Phase B to validate design decisions against the original goals and objectives and ConOps. All aspects of the life cycle should be considered, including design decisions that affect training, operations resource management, human factors, safety, habitability and environment, and maintainability and supportability.

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

Phase B culminates in a series of PDRs, containing the system-level PDR and PDRs for lower level end items as appropriate. The PDRs reflect the successive refinement of requirements into designs. (See the doctrine of successive refinement in Section 4.4.1.2 and Figure 4.4-2.) Design issues uncovered in the PDRs should be resolved so that final design can begin with unambiguous design-to specifications. From this point on, almost all changes to the baseline are expected to represent successive refinements, not fundamental changes. As noted in Figure 2.5-3, significant design changes at and beyond Phase B become increasingly expensive.

Space Flight Phase B: Preliminary Design and Technology Completion

Purpose

To define the project in enough detail to establish an initial baseline capable of meeting mission needs. Develop system structure end product (and enabling product) requirements and generate a preliminary design for each system structure end product.

Typical Activities and Their Products

- Review and update documents baselined in previous phases
- Monitor progress against plans
- Develop the preliminary design
  - Identify one or more feasible preliminary designs including internal and external interfaces
  - Perform analyses of candidate designs and report results
  - Conduct engineering development tests as needed and report results
  - Perform human systems integration assessments
  - Select a preliminary design solution
- Develop operations plans based on matured ConOps
  - Define system operations as well as Principal Investigator (PI)/contract proposal management, review, and access and contingency planning
- Report technology development results
- Update cost range estimate and schedule data (Note that after PDR changes are incorporated and costed, at KDP C this will turn into the Agency Baseline Commitment)
- Improve fidelity of models and prototypes used in evaluations
- Identify and update risks
- Develop appropriate level safety data package and security plan
- Develop preliminary plans
  - Orbital Debris Assessment
  - Decommissioning Plan
  - Disposal Plan
- Perform required Phase B technical activities from NPR 7120.5 as applicable
- Satisfy Phase B reviews' entrance/success criteria from NPR 7123.1

Reviews

- PDR
- Safety review

3.6 Project Phase C: Final Design and Fabrication

The purpose of Phase C is to complete and document the detailed design of the system that meets the detailed requirements and to fabricate, code, or otherwise realize the products. During Phase C, activities are performed to establish a complete design (product baseline), fabricate or produce hardware, and code software in preparation for integration. Trade studies continue and results are used to validate the design against project goals, objectives, and ConOps. Engineering test units more closely resembling actual hardware are built and tested to establish confidence that the design will function in the expected environments. Human subjects representing the user population participate in operations evaluations of the design, use, maintenance, training procedures, and interfaces. Engineering specialty and crosscutting analysis results are integrated into the design, and the manufacturing process and controls are defined and valid. Systems engineers are involved in this phase to ensure the final detailed designs of the various systems will work together, are compatible, and are likely to meet the customer expectations and applicable requirements. During fabrication, the systems engineer is available to answer questions and work any interfacing issues that might arise.

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

Phase C contains a series of CDRs containing the system-level CDR and CDRs corresponding to the different levels of the system hierarchy. A CDR for each end item should be held prior to the start of fabrication/production for hardware and prior to the start of coding of deliverable software products. Typically, the sequence of CDRs reflects the integration process that will occur in the next phase; that is, from lower level CDRs to the system-level CDR. Projects, however, should tailor the sequencing of the reviews to meet the needs of the project. If there is a production run of products, a PRR will be performed to ensure the production plans, facilities, and personnel are ready to begin production. Phase C culminates with an SIR. Training requirements and preliminary mission operations procedures are created and baselined. The final

product of this phase is a product ready for integration.

Space Flight Phase C: Final Design and Fabrication

Purpose

To complete the detailed design of the system (and its associated subsystems, including its operations systems), fabricate hardware, and code software. Generate final designs for each system structure end product.

Typical Activities and Their Products

- Review and update documents baselined in previous phases
- Monitor progress against plans
- Develop and document hardware and software detailed designs
  - Fully mature and define selected preliminary designs
  - Add remaining lower level design specifications to the system architecture
  - Perform and archive trade studies
  - Perform development testing at the component or subsystem level
  - Fully document final design and develop data package
- Develop/refine and baseline plans
  - Interface definitions
  - Implementation plans
  - Integration plans
  - Verification and validation plans
  - Operations plans
- Develop/refine preliminary plans
  - Decommissioning and disposal plans, including human capital transition
  - Spares
  - Communications (including command and telemetry lists)
- Develop/refine procedures for
  - Refine integration
  - Manufacturing and assembly
  - Verification and validation
- Fabricate (or code) the product
- Identify and update risks
- Monitor project progress against project plans
- Prepare launch site checkout and post launch activation and checkout
- Finalize appropriate level safety data package and updated security plan
- Identify opportunities for preplanned product improvement
- Refine orbital debris assessment
- Perform required Phase C technical activities from NPR 7120.5 as applicable
- Satisfy Phase C review entrance/success criteria from NPR 7123.1

Reviews

- CDR
- PRR
- SIR
- Safety review

3.7 Project Phase D: System Assembly, Integration and Test, Launch

The purpose of Phase D is to assemble, integrate, verify, validate, and launch the system. These activities focus on preparing for the Flight Readiness Review (FRR)/Mission Readiness Review (MRR). Activities include assembly, integration, verification, and validation of the system, including testing the flight system to expected environments within margin. Other activities include updating operational procedures, rehearsals and training of operating personnel and crew members, and implementation of the logistics and spares planning. For flight projects, the focus of activities then shifts to prelaunch integration and launch. System engineering is involved in all aspects of this phase including answering questions, providing advice, resolving issues, assessing results of the verification and validation tests, ensuring that the V&V results meet the customer expectations and applicable requirements, and providing information to decision makers for go/no-go decisions.

The planning for Phase D activities was initiated in Phase A. For IT projects, refer to the *IT Systems Engineering Handbook*. The planning for the activities should be performed as early as possible since changes at this point can become costly. Phase D concludes with a system that has been shown to be capable of accomplishing the purpose for which it was created.

Space Flight Phase D: System Assembly, Integration and Test, Launch

Purpose

To assemble and integrate the system (hardware, software, and humans), meanwhile developing confidence that it will be able to meet the system requirements. Launch and prepare for operations. Perform system end product implementation, assembly, integration and test, and transition to use.

Typical Activities and Their Products

- Update documents developed and baselined in previous phases
- Monitor project progress against plans
- Identify and update risks
- Integrate/assemble components according to the integration plans
- Perform verification and validation on assemblies according to the V&V Plan and procedures
  - Perform system qualification verifications, including environmental verifications
  - Perform system acceptance verifications and validation(s) (e.g., end-to-end tests encompassing all elements; i.e., space element, ground system, data processing system)
  - Assess and approve verification and validation results
  - Resolve verification and validation discrepancies
  - Archive documentation for verifications and validations performed
  - Baseline verification and validation report
- Prepare and baseline
  - Operator's manuals
  - Maintenance manuals
  - Operations handbook
- Prepare launch, operations, and ground support sites including training as needed
  - Train initial system operators and maintainers
  - Train on contingency planning
  - Confirm telemetry validation and ground data processing
  - Confirm system and support elements are ready for flight
  - Provide support to the launch and checkout of the system
  - Perform planned on-orbit operational verification(s) and validation(s)
- Document lessons learned. Perform required Phase D technical activities from NPR 7120.5
- Satisfy Phase D reviews' entrance/success criteria from NPR 7123.1

Reviews

- Test Readiness Reviews (TRRs)
- System Acceptance Review (SAR) or pre-Ship Review
- ORR
- FRR
- System functional and physical configuration audits
- Safety review

3.8 Project Phase E: Operations and Sustainment

The purpose of Phase E is to conduct the prime mission to meet the initially identified need and to maintain support for that need. The products of the phase are the results of the mission and performance of the system.

Systems engineering personnel continue to play a role during this phase since integration often overlaps with operations for complex systems. Some programs have repeated operations/flights which require configuration changes and new mission objectives with each occurrence. And systems with complex sustainment needs or human involvement will likely require evaluation and adjustments that may be beyond the scope of operators to perform. Specialty engineering disciplines, like maintainability and logistics servicing, will be performing tasks during this phase as well. Such tasks may require reiteration and/or recursion of the common systems engineering processes.

Systems engineering personnel also may be involved in in-flight anomaly resolution. Additionally, software development may continue well into Phase E. For example, software for a planetary probe may be developed and uplinked while in-flight. Another example would be new hardware developed for space station increments.

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

For additional information on systems engineering in Phase E, see appendix T.

Space Flight Phase E: Operations and Sustainment

Purpose

To conduct the mission and meet the initially identified need and maintain support for that need. Implement the mission operations plan.

Typical Activities and Their Products

- Conduct launch vehicle performance assessment. Commission and activate science instruments
- Conduct the intended prime mission(s)
- Provide sustaining support as planned
  - Implement spares plan
  - Collect engineering and science data
  - Train replacement operators and maintainers
  - Train the flight team for future mission phases (e.g., planetary landed operations)
  - Maintain and approve operations and maintenance logs
  - Maintain and upgrade the system
  - Identify and update risks
  - Address problem/failure reports
  - Process and analyze mission data
  - Apply for mission extensions, if warranted
- Prepare for deactivation, disassembly, decommissioning as planned (subject to mission extension)
- Capture lessons learned
- Complete post-flight evaluation reports
- Develop final mission report
- Perform required Phase E technical activities from NPR 7120.5
- Satisfy Phase E reviews' entrance/success criteria from NPR 7123.1

Reviews

- Post-Launch Assessment Review (PLAR)
- Critical Event Readiness Review (CERR)
- Post-Flight Assessment Review (PFAR) (human space flight only)
- DR
- System upgrade review
- Safety review

3.9 Project Phase F: Closeout

The purpose of Phase F is to implement the systems decommissioning and disposal planning and analyze any returned data and samples. The products of the phase are the results of the mission. The system engineer is involved in this phase to ensure all technical information is properly identified and archived, to answer questions, and to resolve issues as they arise.

Phase F deals with the final closeout of the system when it has completed its mission; the time at which this occurs depends on many factors. For a flight system that returns to Earth after a short mission duration, closeout may require little more than de-integrating the hardware and returning it to its owner. On flight projects of long duration, closeout may proceed according to established

plans or may begin as a result of unplanned events, such as failures. Refer to NASA Policy Directive (NPD) 8010.3, Notification of Intent to Decommission or Terminate Operating Space Systems and Terminate Missions, for terminating an operating mission. Alternatively, technological advances may make it uneconomical to continue operating the system either in its current configuration or an improved one.

Phase F: Closeout

Purpose

To implement the systems decommissioning/disposal plan developed in Phase E and perform analyses of the returned data and any returned samples.

Typical Activities and Their Products

- Dispose of the system and supporting processes
- Document lessons learned
- Baseline mission final report
- Archive data
- Capture lessons learned
- Perform required Phase F technical activities from NPR 7120.5
- Satisfy Phase F reviews' entrance/success criteria from NPR 7123.1

Reviews

- DRR

To limit space debris, NPR 8715.6, NASA Procedural Requirements for Limiting Orbital Debris, provides requirements for removing Earth-orbiting robotic satellites from their operational orbits at the end of their useful life. For Low Earth Orbit (LEO) missions, the satellite is usually deorbited. For small satellites, this is accomplished by allowing the orbit to slowly decay until the satellite eventually burns up in the Earth's atmosphere. Larger, more massive satellites and observatories should be designed to demise or deorbit in a controlled manner so that they can be safely targeted for impact in a remote area of the ocean. The Geostationary (GEO) satellites at 35,790 km above the Earth cannot be practically deorbited, so they are boosted to a higher orbit well beyond the crowded operational GEO orbit.

In addition to uncertainty about when this part of the phase begins, the activities associated with safe closeout of a system may be long and complex and may affect the system design. Consequently, different options and strategies should be considered during the project's earlier phases along with the costs and risks associated with the different options.

3.10 Funding: The Budget Cycle

NASA operates with annual funding from Congress. This funding results, however, from a continuous rolling process of budget formulation, budget enactment, and finally, budget execution. NASA's *Financial Management Requirements (FMR) Volume 4* provides the concepts, the goals, and an overview of NASA's budget system of resource alignment referred to as Planning, Programming, Budgeting, and Execution (PPBE) and establishes guidance on the programming and budgeting phases of the PPBE process, which are critical to budget formulation for NASA. Volume 4 includes strategic budget planning and resources guidance, program review, budget development, budget presentation, and justification of estimates to the Office of Management and Budget (OMB) and to Congress. It also provides detailed descriptions of the roles and responsibilities for key players in each step of the process. It consolidates current legal, regulatory, and administrative policies and procedures applicable to NASA. A highly simplified representation of the typical NASA budget cycle is shown in Figure 3.10-1.

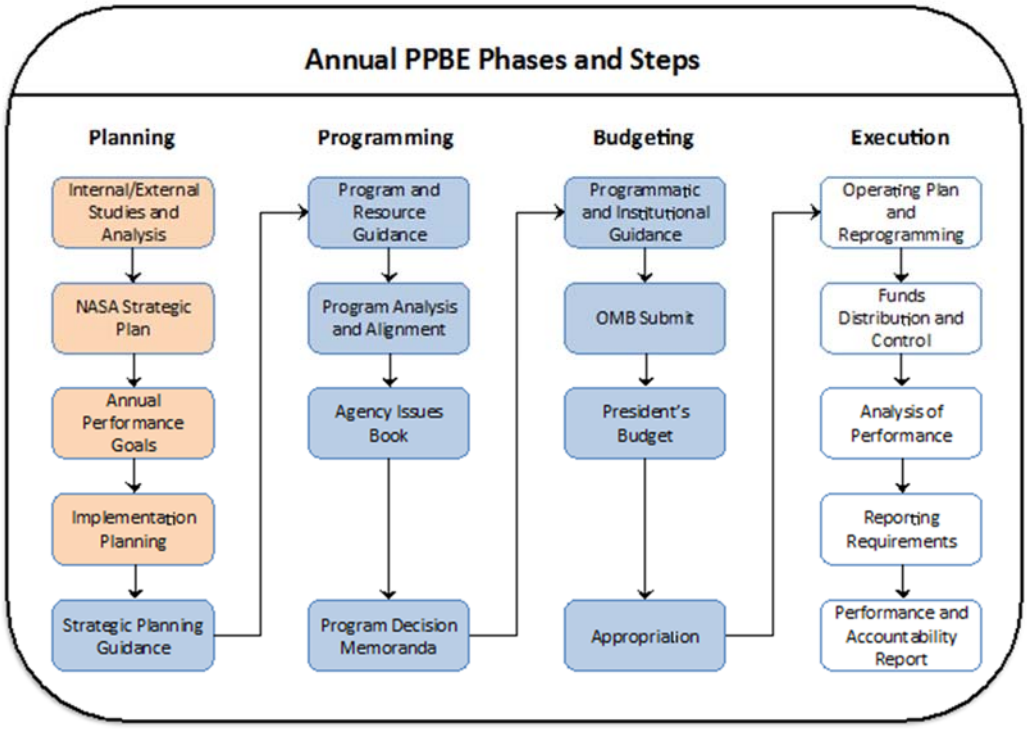


Figure 3.10-1 Typical NASA Budget Cycle

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

delayed past the start of the fiscal year on October 1. In those years, NASA has operated on a continuing appropriations resolution passed by Congress.

With annual funding, there is an implicit funding control gate at the beginning of every fiscal year. While these gates place planning requirements on the project and can make significant replanning necessary, they are not part of an orderly systems engineering process. Rather, they constitute one of the sources of uncertainty that affect project risks, and they are essential to consider in project planning. NASA systems engineers need to remain vigilant of life-cycle cost. Without vigilance and tools to help track life-cycle costs, budget constraints during a product's development can lead to pushing costs down the road to the operational phase in addition to compromises to design. For example, a product that may have been cost-efficient to develop subsequently may exhibit high operations, maintenance, and logistics costs due to design and production insensitivity to operational phase needs and demands during development.

3.11 Tailoring and Customization of NPR 7123.1 Requirements

In this section, the term "requirements" refers to the "shall" statements imposed from Agency directives. This discussion focuses on the tailoring of the requirements contained in NPR 7123.1.

3.11.1 Introduction

NASA policy recognizes the need to accommodate the unique aspects of each program or project to achieve mission success in an efficient and economical manner. Tailoring is a process used to accomplish this.

NPR 7123.1 defines "tailoring" as "the process used to seek relief from SE NPR requirements consistent with program or project objectives, allowable risk, and constraints." Tailoring results in deviations or waivers (see NPR 7120.5, Section 3.5) to SE requirements and is documented in the next revision of the SEMP (e.g., via the Compliance Matrix).

Since NPR 7123.1 was written to accommodate programs and projects regardless of size or complexity, the NPR requirements leave considerable latitude for interpretation. Therefore, the term "customization" is introduced and is defined as "the modification of recommended SE practices that are used to accomplish the SE requirements." Customization does not require waivers or deviations, but significant customization should be documented in the SEMP.

Tailoring and customization are essential systems engineering tools that are an accepted and expected part of establishing the proper SE NPR requirements for a program or project. Although tailoring is expected for all sizes of projects and programs, small projects present opportunities and challenges that are different from those of large, traditional projects such as the Shuttle, International Space Station, Hubble Space Telescope, and Mars Science Laboratory.

While the technical aspects of small projects are generally narrower and more focused, they can also be challenging when their objectives are to demonstrate advanced technologies or provide "one of a kind" capabilities. At the same time, their comparatively small budgets and restricted schedules dictate lean and innovative implementation approaches to project management and systems engineering. Tailoring and customization allow programs and projects to be successful in achieving technical objectives within cost and schedule constraints. The key is effective

tailoring that reflects lessons learned and best practices. Tailoring the SE requirements and customizing the SE best practices to the specific needs of the project helps to obtain the desired benefits while eliminating unnecessary overhead. To accomplish this, an acceptable risk posture must be understood and agreed upon by the project, customer/stakeholder, Center management, and independent reviewers. Even with this foundation, however, the actual process of appropriately tailoring SE requirements and customizing NPR 7123.1 practices to a specific project can be complicated and arduous. Effective approaches and experienced mentors make the tailoring process for any project more systematic and efficient.

Chapter 6 of the *NASA Software Engineering Handbook* provides guidance on tailoring SE requirements for software projects.

3.11.2 Criteria for Tailoring

NPR 8705.4, Risk Classification for NASA Payloads, is intended for assigning a risk classification to projects and programs. It establishes baseline criteria that enable users to define the risk classification level for NASA payloads on human or non-human-rated launch systems or carrier vehicles. It is also a starting point for understanding and defining criteria for tailoring.

The extent of acceptable tailoring depends on several characteristics of the program/project such as the following:

1. **Type of mission**. For example, the requirements for a human space flight mission are much more rigorous than those for a small robotic mission.
2. **Criticality of the mission** in meeting the Agency Strategic Plan. Critical missions that absolutely must be successful may not be able to get relief from NPR requirements.
3. **Acceptable risk level.** If the Agency and the customer are willing to accept a higher risk of failure, some NPR requirements may be waived.
4. **National significance.** A project that has great national significance may not be able to get relief from NPR requirements.
5. **Complexity.** Highly complex missions may require more NPR requirements in order to keep systems compatible, whereas simpler ones may not require the same level of rigor.
6. **Mission lifetime.** Missions with a longer lifetime need to more strictly adhere to NPR requirements than short-lived programs/projects.
7. **Cost of mission.** Higher cost missions may require stricter adherence to NPR requirements to ensure proper program/project control.
8. **Launch constraints.** If there are several launch constraints, a project may need to be more fully compliant with Agency requirements.

3.11.3 Tailoring SE NPR Requirements Using the Compliance Matrix

NPR 7123.1 includes a Compliance Matrix (appendix H.2) to assist programs and projects in verifying that they meet the specified NPR requirements. The Compliance Matrix documents the program/project's compliance or intent to comply with the requirements of the NPR or

justification for tailoring. The Compliance Matrix can be used to assist in identifying where major customization of the way (e.g., formality and rigor) the NPR requirements will be accomplished and to communicate that customization to the stakeholders. The tailoring process (which can occur at any time in the program or project's life cycle) results in deviations or waivers to the NPR requirements depending on the timing of the request. Deviations and waivers of the requirements can be submitted separately to the Designated Governing Authority or via the Compliance Matrix. The Compliance Matrix is attached to the Systems Engineering Management Plan (SEMP) when submitted for approval. Alternatively, if there is no stand-alone SEMP and the contents of the SEMP are incorporated into another document such as the project plan, the Compliance Matrix can be captured within that plan.

Figure 3.11-2 illustrates a notional tailoring process for a space flight project. Project management (such as the project manager / the Principal Investigator / the task lead, etc.) assembles a project team to tailor the NPR requirements codified in the Compliance Matrix. To properly classify the project, the team (chief engineer, lead systems engineer, safety and mission assurance, etc.) needs to understand the building blocks of the project such as the needs, goals, and objectives as well as the appropriate risk posture.

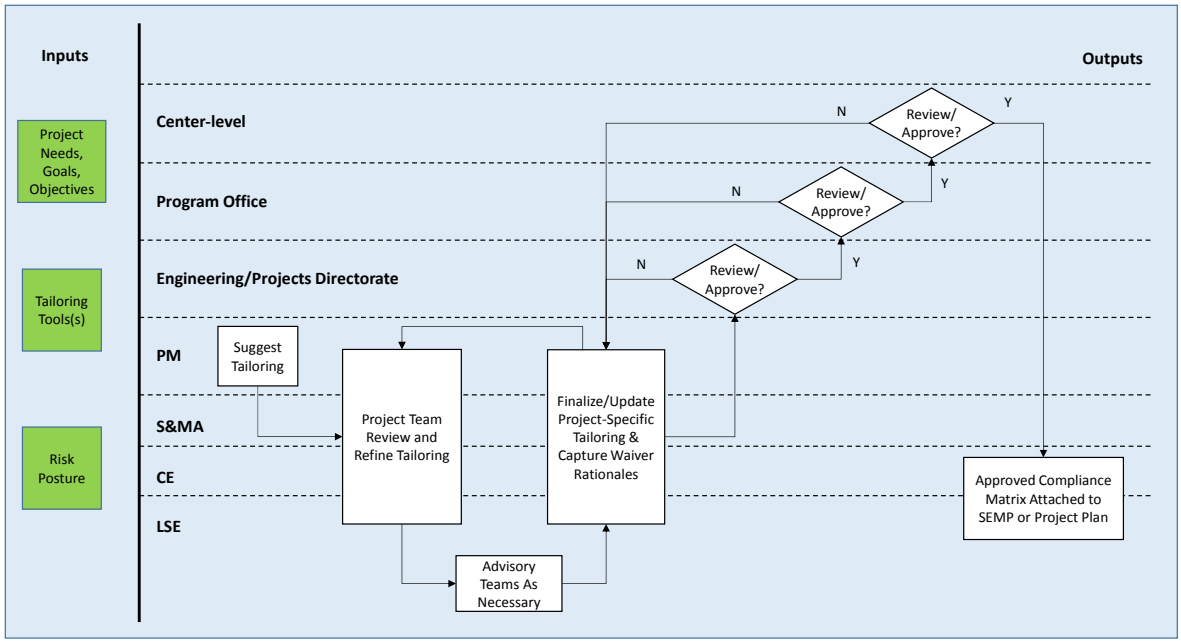


Figure 3.11-2 Notional Space Flight Products Tailoring Process

Through an iterative process, the project team goes through the NPR requirements in the Compliance Matrix to tailor the requirements. A tailoring tool with suggested guidelines may make the tailoring process easier if available. Several NASA Centers including LaRC and MSFC have developed tools for use at their Centers which could be adapted for other Centers. Guidance from Subject Matter Experts (SMEs) should be sought to determine the appropriate amount of tailoring for a specific project. The Compliance Matrix provides rationales for each of the NPR requirements to assist in understanding. Once the tailoring is finalized and recorded in the Compliance Matrix with appropriate rationales, the requested tailoring proceeds through the appropriate governance model for approval.

3.11.4 Ways to Tailor a SE Requirement

Tailoring often comes in three areas:

1. Eliminating a requirement that does not apply to the specific program/project.
2. Eliminating a requirement that is overly burdensome (i.e., when the cost of implementing the requirement adds more risk to the project by diverting resources than the risk of not complying with the requirement).
3. Scaling the requirement in a manner that better balances the cost of implementation and the project risk.

Customizing SE practices can include the following:

1. Adjusting the way each of the 17 SE processes is implemented.
2. Adjusting the formality and timing of reviews.

3.11.4.1 Non-Applicable NPR Requirements

Each requirement in NPR 7123.1 is assessed for applicability to the individual project or program. For example, if the project is to be developed completely in-house, the requirements of the NPR's chapter 4 on contracts would not be applicable. If a system does not contain software, then none of the NPR requirements for developing and maintaining software would be applicable.

3.11.4.2 Adjusting the Scope

Depending on the project or program, some relief on the scope of a requirement may be appropriate. For example, although the governing project management directive (e.g., NPR 7120.5, 7150.2, 7120.7, 7120.8) for a program/project may require certain documents to be standalone, the SE NPR does not require any additional stand-alone documents. For small projects, many of the plans can be described in just a few paragraphs or pages. In these types of projects, any NPR requirements stating that the plans need to be stand-alone document would be too burdensome. In these cases, the information can simply be written and included as part of the project plan or SEMP. If the applicable project management directive (e.g., NPR 7120.5 or NPR 7120.8) requires documents to be stand-alone, a program/project waiver/deviation is needed. However, if there is no requirement or Center expectation for a stand-alone document, a project can customize where that information is recorded and no waiver or deviation is required. Capturing where this information is documented within the systems engineering or project management Compliance Matrix would be useful for clarity.

3.11.4.3 Formality and Timing of Reviews

The governing project management directive identifies the required or recommended life cycle for the specific type of program/project. The life cycle defines the number and timing of the various reviews; however, there is considerable discretion concerning the formality of the review and how to conduct it. NPR 7123.1, appendix G, provides extensive guidance for suggested review entrance and success criteria. It is expected that the program/project will customize these criteria in a manner that makes sense for their program/project. The SE NPR does not require a

waiver/deviation for this customization; however, departures from review elements required by other NPRs need to be addressed by tailoring those documents.

If a program/project decides it does not need one of the required reviews, a waiver or deviation is needed. However, the SE NPR does not specify a minimum amount of spacing for these reviews. A small project may decide to combine the SRR and the SDR (or Mission Definition Review (MDR)) for example. As long as the intent for *both* reviews is accomplished, the SE NPR does not require a waiver or deviation. (Note that even though the SE NPR does not require it, a waiver or deviation may still be required in the governing project management NPR.) This customization and/or tailoring should be documented in the Compliance Matrix and/or the review plan or SEMP.

Unless otherwise required by the governing project management directives, the formality of the review can be customized as appropriate for the type of program/project. For large projects, it might be appropriate to conduct a very formal review with a formal Review Item Discrepancy (RID)/ Request for Action (RFA) process, a summary, and detailed presentations to a wide audience including boards and pre-boards over several weeks. For small projects, that same review might be done in a few hours across a tabletop with a few stakeholders and with issues and actions simply documented in a word or PowerPoint document.

The NASA Engineering Network Systems Engineering Community of Practice, located at <https://nen.nasa.gov/web/se> includes document templates for milestone review presentations required by the NASA SE process.

3.11.5 Examples of Tailoring and Customization

Table 3.11-1 shows an example of the types of missions that can be defined based on a system that breaks projects into various types ranging from a very complex type A to a much simpler type F. When tailoring a project, the assignment of specific projects to particular types should be viewed as guidance, not as rigid characterization. Many projects will have characteristics of multiple types, so the tailoring approach may permit more tailoring for those aspects of the project that are simpler and more open to risk and less tailoring for those aspects of the project where complexity and/or risk aversion dominate. These tailoring criteria and definitions of project "types" may vary from Center to Center and from Mission Directorate to Mission Directorate according to what is appropriate for their missions. Table 3.11-2 shows an example of how the documentation required of a program/project might also be tailored or customized. The general philosophy is that the simpler, less complex projects should require much less documentation and fewer formal reviews. Project products should be sensibly scaled.

Table 3.11-1 Example of Program/Project Types

	Type A	Type B	Type C	Type D	Type E	Type F
s	Human Space Flight or Very Large Science/Robotic Missions	Non-Human Space Flight or Science/Robotic Missions	Small Science or Robotic Missions	Smaller Science or Technology Missions (ISS payload)	Suborbital or Aircraft or Large Ground based Missions	Aircraft or Ground based technology demonstrations
	High priority, very low (minimized) risk	High priority, low risk	Medium priority, medium risk	Low priority, high risk	Low priority, high risk	Low to very low priority, high risk
	Very high	High	Medium	Medium to Low	Low	Very Low
	Very high to high	High to Medium	Medium to Low	Medium to Low	Low	Low to Very Low
ry	Long. >5 years	Medium. 2-5 years	Short. <2 years	Short. <2 years	N/A	N/A
	High (greater than ~$1B)	High to Medium (~$500M - $1B)	Medium to Low (~$100M - $500M)	Low (~$50M - $100M)	(~$10-50M)	(less than $10-15M)
	Critical	Medium	Few	Few to none	Few to none	N/A
ht	No alternative or re-flight opportunities	Few or no alternative or re-flight opportunities	Some or few alternative or re-flight opportunities	Significant alternative or re-flight opportunities	Significant alternative or re-flight opportunities	Significant alternative or re-flight opportunities
n	All practical measures are taken to achieve minimum risk to mission success. The highest assurance standards are used.	Stringent assurance standards with only minor compromises in application to maintain a low risk to mission success.	Medium risk of not achieving mission success may be acceptable. Reduced assurance standards are permitted.	Medium or significant risk of not achieving mission success is permitted. Minimal assurance standards are permitted.	Significant risk of not achieving mission success is permitted. Minimal assurance standards are permitted.	Significant risk of not achieving mission success is permitted. Minimal assurance standards are permitted.

Criteria	Type A	Type B	Type C	Type D	Type E	Type F
Examples	HST, Cassini, JIMO, JWST, MPCV, SLS, ISS	MER, MRO, Discovery payloads, ISS Facility Class payloads, Attached ISS payloads	ESSP, Explorer payloads, MIDES, ISS complex subrack payloads, PA-1, ARES 1-X, MEDLI, CLARREO, SAGE III, Calipso	SPARTAN, GAS Can, technology demonstrators, simple ISS, express middeck and subrack payloads, SMEX, MISSE-X, EV-2	IRVE-2, IRVE-3, HiFIRE, HyBoLT, ALHAT, STORRM, Earth Venture I	DAWNAir, InFlame, Research, technology demonstrations

Table 3.11-2 Example of Tailoring NPR 7120.5 Required Project Products

	Type A	Type B	Type C	Type D	Type E	Type F
Example Project Technical Products						
Concept Documentation	Fully Compliant	Fully Compliant	Fully Compliant	Tailor	Tailor	Tailor
Mission, Spacecraft, Ground, and Payload Architectures	Fully Compliant	Fully Compliant	Fully Compliant	Tailor	Tailor	Tailor
Project-Level, System and Subsystem Requirements	Fully Compliant	Fully Compliant	Fully Compliant	Fully Compliant	Tailor	Tailor
Design Documentation	Fully Compliant	Fully Compliant	Fully Compliant	Fully Compliant	Tailor	Tailor
Operations Concept	Fully Compliant	Fully Compliant	Fully Compliant	Tailor	Tailor	Tailor
Technology Readiness Assessment Documentation	Fully Compliant	Fully Compliant	Fully Compliant	Tailor	Tailor	Tailor
Human Systems Integration Plan	Fully Compliant	Fully Compliant	Fully Compliant	Tailor	Tailor	Tailor
Heritage Assessment Documentation	Fully Compliant	Fully Compliant	Fully Compliant	Tailor	Tailor	Tailor
Safety Data Packages	Fully Compliant	Fully Compliant	Fully Compliant	Fully Compliant	Tailor	Tailor
ELV Payload Safety Process Deliverables	Fully Compliant	Fully Compliant	Fully Compliant	Fully Compliant	Fully Compliant	Not Applicabl
Verification and Validation Report	Fully Compliant	Fully Compliant	Fully Compliant	Tailor	Tailor	Tailor
Operations Handbook	Fully Compliant	Fully Compliant	Fully Compliant	Tailor	Tailor	Not Applicabl
End of Mission Plans	Fully Compliant	Fully Compliant	Fully Compliant	Tailor	Tailor	Tailor
Mission Report	Fully Compliant	Fully Compliant	Tailor	Tailor	Tailor	Tailor
Example Project Plan Control Plans						
Risk Management	Fully Compliant	Fully Compliant	Fully Compliant	Tailor	Tailor	Not Applicable

	Type A	Type B	Type C	Type D	Type E	Type F
Plan						
Technology Development plan	Fully Compliant	Fully Compliant	Fully Compliant	Fully Compliant	Not Applicable	Not Applicable
Systems Engineering Management Plan	Fully Compliant	Fully Compliant	Fully Compliant	Tailor	Tailor	Tailor
Software Management plan	Fully Compliant	Fully Compliant	Tailor	Tailor	Tailor	Tailor
Verification and Validation Plan	Fully Compliant	Fully Compliant	Tailor	Tailor	Tailor	Tailor
Review Plan	Fully Compliant	Fully Compliant	Fully Compliant	Tailor	Tailor	Tailor
integrated Logistics Support Plan	Fully Compliant	Fully Compliant	Fully Compliant	Tailor	Tailor	Not Applicable
Science Data Management Plan	Fully Compliant	Fully Compliant	Fully Compliant	Tailor	Tailor	Not Applicable
Integration Plan	Fully Compliant	Fully Compliant	Fully Compliant	Fully Compliant	Tailor	Tailor
Configuration Management Plan	Fully Compliant	Fully Compliant	Fully Compliant	Fully Compliant	Tailor	Tailor
Technology Transfer (formerly Export) Control Plan	Fully Compliant	Fully Compliant	Fully Compliant	Fully Compliant	Tailor	Tailor
Lessons Learned Plan	Fully Compliant	Fully Compliant	Fully Compliant	Fully Compliant	Tailor	Tailor
Human Rating Certification Package	Fully Compliant	Not Applicable	Not Applicable	Not Applicable	Not Applicable	Not Applicable

3.11.6 Approvals for Tailoring

Deviations and waivers of the requirements for the SE NPR can be submitted separately to the requirements owners or in bulk using the appropriate Compliance Matrix found in NPR 7123.1 appendix H. If it is a Center that is requesting tailoring of the NPR requirements for standard use at the Center, appendix H.1 is completed and submitted to the OCE for approval upon request or as changes to the Center processes occur. If a program/project whose responsibility has been delegated to a Center is seeking a waiver/deviation from the NPR requirements, the Compliance

Matrix in appendix H.2 is used. In these cases, the Center Director or designee will approve the waiver/deviation.

The result of this tailoring, whether for a Center or for a program/project, should also be captured in the next revision of the SEMP along with supporting rationale and documented approvals from the requirement owner. This allows communication of the approved waivers/deviations to the entire project team as well as associated managers. If an independent assessment is being conducted on the program/project, this also allows appropriate modification of expectations and assessment criteria. Table 3.11-3 provides some examples of tailoring captured within the H.2 Compliance Matrix.

Table 3.11-3 Example Use of a Compliance Matrix

Req ID	SE NPR Section	Requirement Statement	Rationale	Req. Owner	Comply?	Justification
SE-05	2.1.5.2	For those requirements owned by Center Directors, the technical team shall complete the Compliance Matrix in Appendix H.2 and include it in the SEMP.	For programs and projects, the Compliance Matrix in Appendix H.2 is filled out showing that the program/project is compliant with the requirements of this NPR (or a particular Center's implementation of NPR 7123.1, whichever is applicable) or any tailoring thereof is identified and approved by the Center Director or designee as part of the program/project SEMP.	CD	Fully Compliant	
SE-06	2.1.6.1	The DGA shall approve the SEMP, waiver authorizations, and other key technical documents to ensure independent assessment of technical content.	The DGA, who is often the TA, provides an approval of the SEMPs, waivers to technical requirements and other key technical document to provide assurance of the applicability and technical quality of the products.	CD	Fully Complaint	
SE-24	4.2.1	The NASA technical team shall define the engineering activities for the periods before contract award, during contract performance, and upon contract completion in the SEMP.	It is important for both the government and contractor technical teams to understand what activities will be handled by which organization throughout the product life cycle. The contractor(s) will typically develop a SEMP or its equivalent to describe the technical activities in their portion of the project, but an overarching SEMP is needed that will describe all technical activities across the life cycle whether contracted or not.	CD	Not Applicable	Project is conducted entirely in-house and therefore there are no contracts involved

4.0 System Design Processes

This chapter describes the activities in the system design processes listed in Figure 2.1-1. The chapter is separated into sections corresponding to processes 1 to 4 listed in Figure 2.1-1. The tasks within each process are discussed in terms of inputs, activities, and outputs. Additional guidance is provided using examples that are relevant to NASA projects.

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

Figure 4.0-1 illustrates the recursive relationship among the four system design processes. These processes start with a study team collecting and clarifying the stakeholder expectations, including the mission objectives, constraints, design drivers, operational objectives, and criteria for defining mission success. This set of stakeholder expectations and high-level requirements is used to drive an iterative design loop where a strawman architecture/design, the concept of operations, and derived requirements are developed. These three products should be consistent with each other and will require iterations and design decisions to achieve this consistency. Once consistency is achieved, analyses allow the project team to validate the proposed design against the stakeholder expectations. A simplified validation asks the questions: Will the system work as expected? Is the system achievable within budget and schedule constraints? Does the system provide the functionality and fulfill the operational needs that drove the project's funding approval? If the answer to any of these questions is no, then changes to the design or stakeholder expectations will be required, and the process starts again. This process continues until the system—architecture, ConOps, and requirements—meets the stakeholder expectations.

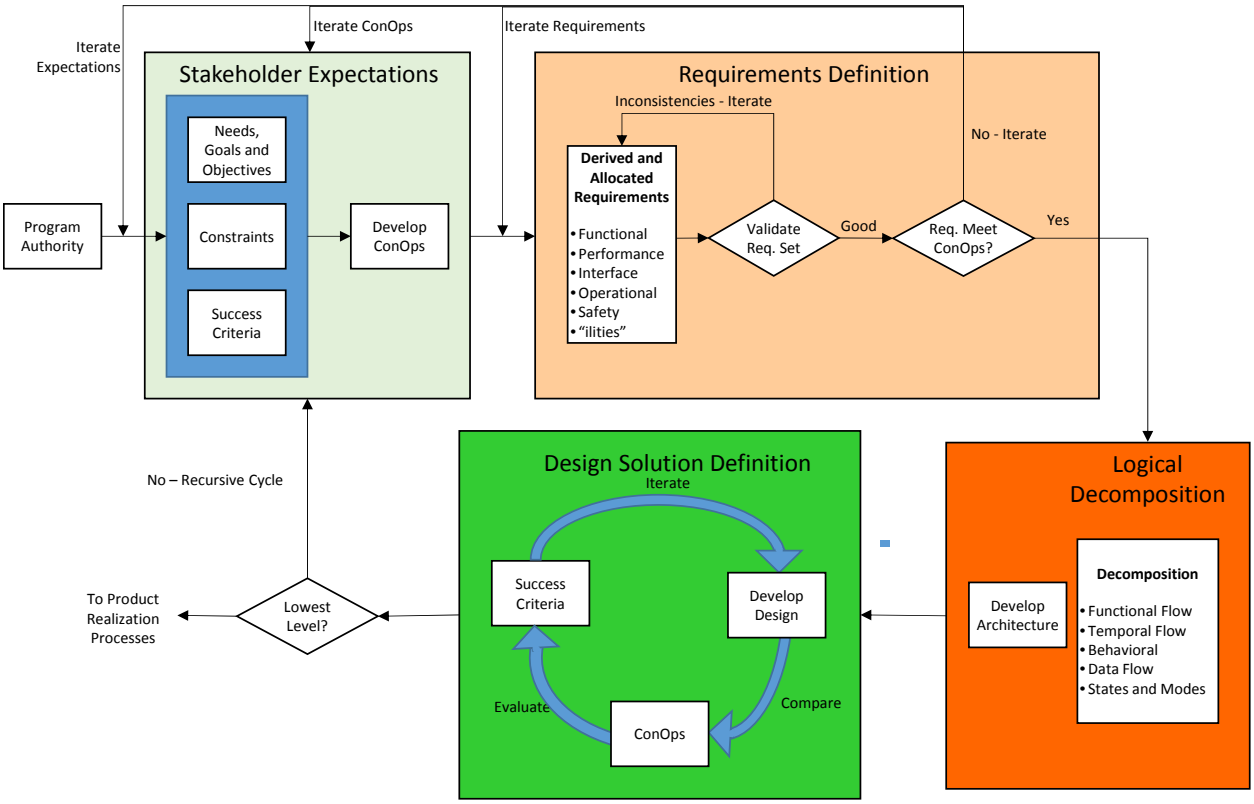


Figure 4.0‑1 Interrelationships among the System Design Processes

The depth of the design effort should be sufficient to allow analytical verification of the design to the requirements. The design should be feasible and credible when judged by a knowledgeable independent review team and should have sufficient depth to support cost modeling and operational assessment.

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

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

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

System Design Keys

- Successfully understanding and defining the mission objectives and the concept of operations are keys to capturing the stakeholder expectations, which will translate into quality requirements and operational efficiencies over the life cycle of the project.
- Complete and thorough requirements traceability is a critical factor in successful validation of requirements.
- Clear and unambiguous requirements will help avoid misunderstanding when developing the overall system and when making major or minor changes.
- Document all decisions made during the development of the original design concept in the technical data package. This will make the original design philosophy and negotiation results available to assess future proposed changes and modifications against.
- The design solution verification occurs when an acceptable design solution has been selected and documented in a technical data package. The design solution is verified against the system requirements and constraints. However, the validation of a design solution is a continuing recursive and iterative process during which the design solution is evaluated against stakeholder expectations.

4.1 Stakeholder Expectations Definition

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

4.1.1 Process Description

Figure 4.1-1 provides a typical flow diagram for the Stakeholder Expectations Definition Process and identifies typical inputs, outputs, and activities to consider in defining stakeholder expectations.

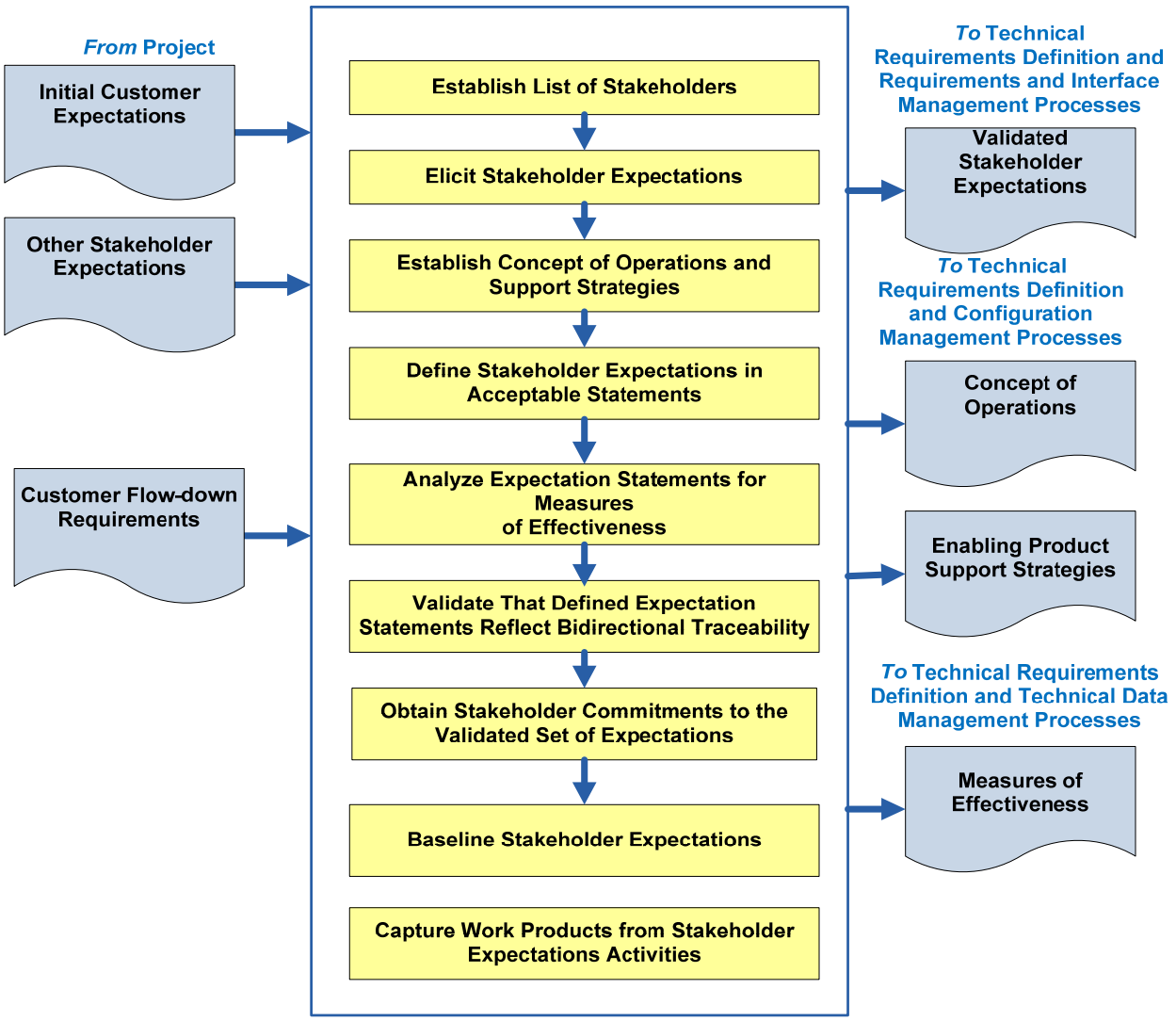


Figure 4.1-1 Stakeholder Expectations Definition Process

4.1.1.1 Inputs

Typical inputs needed for the Stakeholder Expectations Definition Process include the following:

- **Initial Customer Expectations:** These are the needs, goals, objectives, desires, capabilities, and other constraints that are received from the customer for the product within the product layer. For the top-tier products (final end item), these are the expectations of the originating customer who requested the product. For an end product within the product layer, these are the expectations of the recipient of the end item when transitioned.

- **Other Stakeholder Expectations:** These are the expectations of key stakeholders other than the customer. For example, such stakeholders may be the test team that will be receiving the transitioned product (end product and enabling products) or the trainers that will be instructing the operators or managers that are accountable for the product at this layer.

- **Customer Flow-down Requirements:** These are any requirements that are being flowed down or allocated from a higher level (i.e., parent requirements). They are helpful in establishing the expectations of the customer at this layer.

4.1.1.2 Process Activities

4.1.1.2.1 Identify Stakeholders

A "stakeholder" is a group or individual that is affected by or has a stake in the product or project. The key players for a project/product are called the key stakeholders. One key stakeholder is always the "customer." The customer may vary depending on where the systems engineer is working in the PBS. For example, at the topmost level, the customer may be the person or organization that is purchasing the product. For a systems engineer working three or four levels down in the PBS, the customer may be the leader of the team that takes the element and integrates it into a larger assembly. Regardless of where the systems engineer is working within the PBS, it is important to understand what is expected by the customer.

Other interested parties are those who affect the project by providing broad, overarching constraints within which the customers' needs should be achieved. These parties may be affected by the resulting product, the manner in which the product is used, or have a responsibility for providing life-cycle support services. Examples include Congress, advisory planning teams, program managers, maintainers, and mission partners. It is important that the list of stakeholders be identified early in the process, as well as the primary stakeholders who will have the most significant influence over the project.

The customer and users of the system are usually easy to identify. The other key stakeholders may be more difficult to identify and they may change depending on the type of the project and the phase the project is in. Table 4.1-1 provides some examples of stakeholders in the life-cycle phase that should be considered.

Table 4.1-1 Stakeholder Identification throughout the Life Cycle

Life-Cycle Stage	Example Stakeholders
Pre-Phase A	NASA Headquarters, NASA Centers, Presidential Directives, NASA advisory committees, the National Academy of Sciences

Phase A	Mission Directorate, customer, potential users, engineering disciplines, safety organization
Phase B	Customer, engineering disciplines, safety, crew, operations, logistics, production facilities, suppliers, principle investigators
Phase C	Customer, engineering disciplines, safety, crew, operations, logistics, production facilities, suppliers, principle investigators
Phase D	Customer, engineering disciplines, safety, crew, operations, training, logistics, verification team, Flight Readiness Board members
Phase E	Customer, system managers, operations, safety, logistics, sustaining team, crew, principle investigators, users
Phase F	Customer, NASA Headquarters, operators, safety, planetary protection, public

4.1.1.2.2 Understand Stakeholder Expectations

Thoroughly understanding the customer and other key stakeholders' expectations for the project/product is one of the most important steps in the systems engineering process. It provides the foundation upon which all other systems engineering work depends. It helps ensure that all parties are on the same page and that the product being provided will satisfy the customer. When the customer, other stakeholders, and the systems engineer mutually agree on the functions, characteristics, behaviors, appearance, and performance the product will exhibit, it sets more realistic expectations on the customer's part and helps prevent significant requirements creep later in the life cycle.

Through interviews/discussions, surveys, marketing groups, e-mails, a Statement of Work (SOW), an initial set of customer requirements, or some other means, stakeholders specify what is desired as an end state or as an item to be produced and put bounds on the achievement of the goals. These bounds may encompass expenditures (resources), time to deliver, life-cycle support expectations, performance objectives, operational constraints, training goals, or other less obvious quantities such as organizational needs or geopolitical goals. This information is reviewed, summarized, and documented so that all parties can come to an agreement on the expectations.

Figure 4.1-2 shows the type of information needed when defining stakeholder expectations and depicts how the information evolves into a set of high-level requirements. The yellow lines depict validation paths. Examples of the types of information that would be defined during each step are also provided.

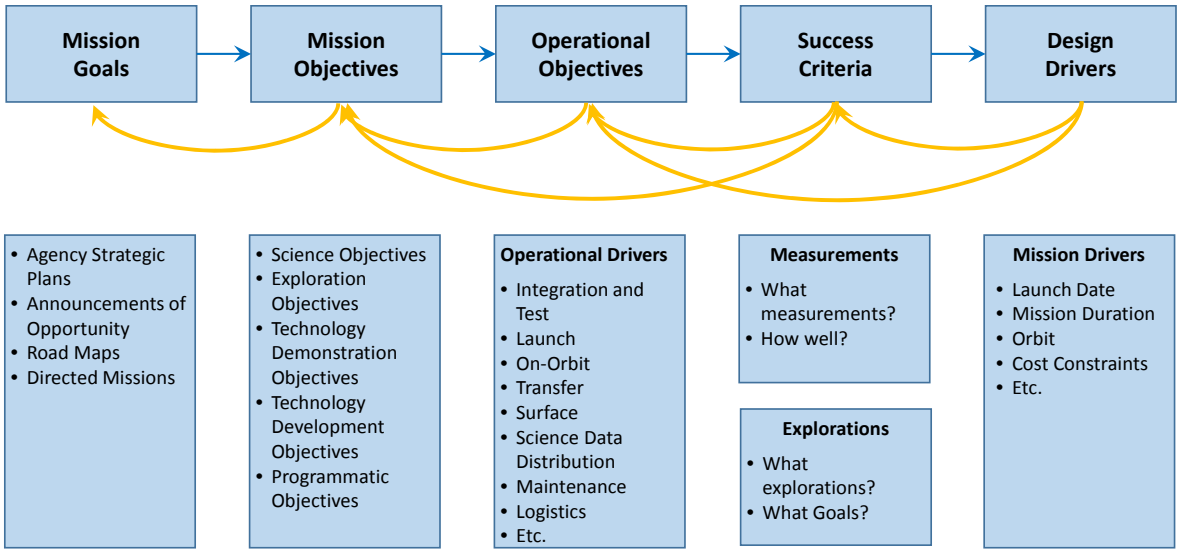


Figure 4.1-2 Information Flow for Stakeholder Expectations

Defining stakeholder expectations begins with the *mission authority* and *strategic objectives* that the mission is meant to achieve. Mission authority changes depending on the category of the mission. For example, science missions are usually driven by NASA Science Mission Directorate strategic plans, whereas the exploration missions may be driven by a Presidential directive. Understanding the objectives of the mission helps ensure that the project team is working toward a common vision. These goals and objectives form the basis for developing the mission, so they need to be clearly defined and articulated.

The project team should also identify the *constraints* that may apply. A "constraint" is a condition that is to be met. Sometimes a constraint is dictated by external factors such as orbital mechanics, an existing system that must be utilized (external interface), a regulatory restriction, or the state of technology; sometimes constraints are the result of the overall budget environment. Concepts of operation and constraints also need to be included in defining the stakeholder expectations. These identify how the system should be operated to achieve the mission objectives.

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

In identifying the full set of expectations, the systems engineer will need to interact with various communities, such as those working in the areas of orbital debris, space asset protection, human systems integration, quality assurance, and reliability. Ensuring that a complete set of expectations is captured will help prevent "surprise" features from arising later in the life cycle. For example, space asset protection may require additional encryption for the forward link commands, additional shielding or filtering for RF systems, use of a different frequency, or other design changes that might be costly to add to a system that has already been developed.

4.1.1.2.3 Identify Needs, Goals, and Objectives

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

Needs, Goals, and Objectives (NGOs) provide a mechanism to ensure that everyone (implementer, customer, and other stakeholders) is in agreement at the beginning of a project in terms of defining the problem that needs to be solved and its scope. NGOs are not contractual requirements or designs.

Needs are defined in the answer to the question "What problem are we trying to solve?" Goals address what must be done to meet the needs; i.e., what the customer wants the system to do. Objectives expand on the goals and provide a means to document specific expectations. (Rationale should be provided where needed to explain why the need, goal, or objective exists, any assumptions made, and any other information useful in understanding or managing the NGO.)

Well-written NGOs provide clear traceability from the needs, then to the goals, and then to objectives. For example, if a given goal does not support a need, or an objective does not support a goal, it should not be part of the integrated set of NGOs. This traceability helps ensure that the team is actually providing what is needed.

The following definitions (source: *Applied Space Systems Engineering* edited by Larson, Kirkpatrick, Sellers, Thomas, and Verma) are provided to help the reader interpret the NGOs contained in this product.

- **Need**: A single statement that drives everything else. It should relate to the problem that the system is supposed to solve but not be the solution. The need statement is singular. Trying to satisfy more than one need requires a trade between the two, which could easily result in failing to meet at least one, and possibly several, stakeholder expectations.
- **Goals**: An elaboration of the need, which constitutes a specific set of expectations for the system. Goals address the critical issues identified during the problem assessment. Goals need not be in a quantitative or measurable form, but they should allow us to assess whether the system has achieved them.
- **Objectives**: Specific target levels of outputs the system must achieve. Each objective should relate to a particular goal. Generally, objectives should meet four criteria. (1) They should be specific enough to provide clear direction, so developers, customers, and testers will understand them. They should aim at results and reflect what the system needs to do but not outline how to implement the solution. (2) They should be measurable, quantifiable, and verifiable. The project needs to monitor the system's success in achieving each objective. (3) They should be aggressive but attainable, challenging but reachable, and targets need to be realistic. Objectives "To Be Determined" (TBD) may be included until trade studies occur,

operations concepts solidify, or technology matures. Objectives need to be feasible before requirements are written and systems designed. (4) They should be results-oriented focusing on desired outputs and outcomes, not on the methods used to achieve the target (what, not how). It is important to always remember that objectives are not requirements. Objectives are identified during pre-Phase A development and help with the eventual formulation of a requirements set, but it is the requirements themselves that are contractually binding and will be verified against the "as-built" system design.

These stakeholder expectations are captured and are considered as initial until they can be further refined through development of the concept of operations and final agreement by the stakeholders.

4.1.1.2.4 Establish Concept of Operations and Support Strategies

After the initial stakeholder expectations have been established, the development of a Concept of Operations (ConOps) will further ensure that the technical team fully understands the expectations and how they may be satisfied by the product, and that that understanding has been agreed to by the stakeholders. This may lead to further refinement of the initial set of stakeholder expectations if gaps or ambiguous statements are discovered. These scenarios and concepts of how the system will behave provide an implementation-free understanding of the stakeholders' expectations by defining what is expected without addressing how (the design) to satisfy the need. It captures required behavioral characteristics and the manner in which people will interact with the system. Support strategies include provisions for fabrication, test, deployment, operations, sustainment, and disposal.

Additional information on the development of the ConOps is discussed in Section 4.1.2.1. Appendix S contains one possible outline for developing a ConOps. The specific sections of the ConOps will vary depending on the scope and purpose of the project.

4.1.1.2.5 Define Stakeholder Expectations in Acceptable Statements

Once the ConOps has been developed, any gaps or ambiguities have been resolved, and understanding between the technical team and stakeholders about what is expected / intended for the system/product has been achieved, the expectations can be formally documented. This often comes in the form of NGOs, mission success criteria, and design drivers. These may be captured in a document, spreadsheet, model, or other form appropriate to the product.

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

4.1.1.2.6 Analyze Expectations Statements for Measures of Effectiveness

The *mission success criteria* define what the mission needs to accomplish to be successful. This could be in the form of science missions, exploration concept for human exploration missions, or

a technological goal for technology demonstration missions. The success criteria also define how well the concept measurements or exploration activities should be accomplished. The success criteria capture the stakeholder expectations and, along with programmatic requirements and constraints, are used within the high-level requirements.

Measures of Effectiveness (MOEs) are the measures of success that are designed to correspond to accomplishment of the system objectives as defined by the stakeholder's expectations. They are stated from the stakeholder's point of view and represent criteria that are to be met in order for the stakeholder to consider the project successful. As such, they can be synonymous with mission / project success criteria. MOEs are developed when the NGOs or other stakeholder expectation documentation is developed. Additional information on MOEs is contained in Section 6.7.2.4 of this guide.

4.1.1.2.7 Validate That Defined Expectation Statements Reflect Bidirectional Traceability

The NGOs or other stakeholder expectation documentation should also capture the source of the expectation. Depending on the location within the product layer, the expectation may be traced to an NGO or a requirement of a higher layer product, to organizational strategic plans, or other sources. Later functions and requirements will be traced to these NGOs. The use of a requirements management tool or model or other application is particularly useful in capturing and tracing expectations and requirements.

4.1.1.2.8 Obtain Stakeholder Commitments to the Validated Set of Expectations

Once the stakeholder and the technical team are in agreement with the expressed stakeholder expectations and the concept of operations, signatures or other forms of commitment are obtained. In order to obtain these commitments, a concept review is typically held on a formal or informal basis depending on the scope and complexity of the system (see Section 6.7). The stakeholder expectations (e.g., NGOs), MOEs, and concept of operations are presented, discussed, and refined as necessary to achieve final agreement. This agreement shows that both sides have committed to the development of this product.

4.1.1.2.9 Baseline Stakeholder Expectations

The set of stakeholder expectations (e.g., NGOs and MOEs) and the concept of operations that are agreed upon are now baselined. Any further changes will be required to go through a formal or informal (depending on the nature of the product) approval process involving both the stakeholder and the technical team.

4.1.1.2.10 Capture Work Products

In addition to developing, documenting, and baselining stakeholder expectations, the ConOps and MOEs discussed above and other work products from this process should be captured. These may include key decisions made, supporting decision rationale and assumptions, and lessons learned in performing these activities.

4.1.1.3 Outputs

Typical outputs for capturing stakeholder expectations include the following:

- **Validated Stakeholder Expectations**: These are the agreed-to set of expectations for this product layer. They are typically captured in the form of needs, goals, and objectives with constraints and assumptions identified. They may also be in the form of models or other graphical forms.
- **Concept of Operations:** The ConOps describes how the system will be operated during the life-cycle phases that will meet stakeholder expectations. It describes the system characteristics from an operational perspective and helps facilitate an understanding of the system goals and objectives and other stakeholder expectations. Examples would be the ConOps document, model, or a Design Reference Mission (DRM).
- **Enabling Product Support Strategies:** These include any special provisions that might be needed for fabrication, test, deployment, operations sustainment, and disposal of the end product. They identify what support will be needed and any enabling products that will need to be developed in order to generate the end product.
- **Measures of Effectiveness:** A set of MOEs is developed based on the stakeholder expectations. These are measures that represent expectations that are critical to the success of the system, and failure to satisfy these measures will cause the stakeholder to deem the system unacceptable.

Other outputs that might be generated:

- **Human/Systems Function Allocation:** This describes the interaction of the hardware and software systems with all personnel and their supporting infrastructure. In many designs (e.g., human space flight) human operators are a critical total-system component and the roles and responsibilities of the humans-in-the-system should be clearly understood. This should include all human/system interactions required for a mission including assembly, ground operations, logistics, in-flight and ground maintenance, in-flight operations, etc.

4.1.2 Stakeholder Expectations Definition Guidance

4.1.2.1 Concept of Operations

The ConOps is an important component in capturing stakeholder expectations and is used in defining requirements and the architecture of a project. It stimulates the development of the requirements and architecture related to the user elements of the system. It serves as the basis for subsequent definition documents such as the operations plan, launch and early orbit plan, and operations handbook, and it provides the foundation for the long-range operational planning activities such as operational facilities, staffing, and network scheduling.

The ConOps is an important driver in the system requirements and therefore should be considered early in the system design processes. Thinking through the ConOps and use cases often reveals requirements and design functions that might otherwise be overlooked. For example, adding system requirements to allow for communication during a particular phase of a mission may require an additional antenna in a specific location that may not be required during the nominal mission. The ConOps should include scenarios for all significant operational situations, including known off-nominal situations. To develop a useful and complete set of scenarios, important malfunctions and degraded-mode operational situations should be

considered. The ConOps is also an important aide to characterizing life-cycle staffing goals and function allocation between humans and systems. In walking through the accomplishment of mission objectives, it should become clear when decisions need to be made as to what the human operators are contributing vs. what the systems are responsible for delivering.

Developing a comprehensive ConOps can be critical to developing a thorough life-cycle acquisition strategy. As noted in Section 7.1, Engineering with Contracts, in some acquisition strategies, the SE development phases are contracted separately from SE operations phases. The responsibility for coordinating and integrating between the two in a manner that cost-effectively accomplishes mission objectives may fall entirely on NASA. Even in development-only projects, the ConOps should be developed with an eye to the long-term, strategic view and should address nominal and off-nominal performance, maintenance, logistics and other similar considerations. Having a long-term view helps ensure that the development phase produces results that fit into a larger conceptual, operational, and cost framework.

The ConOps is important for all projects. For science projects, the ConOps describes how the systems will be operated to achieve the measurement set required for a successful mission. They are usually driven by the data volume of the measurement set. The ConOps for human-crewed exploration projects is likely to be more complex. There are typically more operational phases, more configuration changes, and additional communication links required for human interaction. In general, functions and objectives should be clearly allocated between human operators and systems early in the project and assessed at each life-cycle phase. For any project, the human resources required for operating, maintaining, and supplying the system should be characterized in the ConOps to avoid cost surprises later in the project's life cycle.

The ConOps should consider all aspects of operations including nominal and off-nominal operations during integration, test, and launch through disposal. Typical information contained in the ConOps includes a description of the major phases; operation timelines; operational scenarios and/or DRM; fault management strategies, description of human interaction and required training, end-to-end communications strategy; command and data architecture; operational facilities; integrated logistic support (resupply, maintenance, and assembly); staffing levels and required skill sets; and critical events. The operational scenarios describe the dynamic view of the systems' operations and include how the system is perceived to function throughout the various modes and mode transitions, including interactions with external interfaces, response to anticipated hazard and faults, and during failure mitigations. For exploration missions, multiple DRMs make up a ConOps. The design and performance analysis leading to the requirements should satisfy all of them. Figure 4.1-3 illustrates typical ConOps development for a science mission, and Figure 4.1-4 is an example of an end-to-end operational architecture. For more information about developing the ConOps, see appendix S.

Concept of Operations vs. Operations Concept

Concept of Operations

Developed early in Pre-Phase A by the technical team, describes the overall high-level concept of how the system will be used to meet stakeholder expectations, usually in a time sequenced manner. It describes the system from an operational perspective and helps facilitate an understanding of the system goals. It stimulates the development of the requirements and architecture related to the user elements of the system. It serves as the basis for subsequent definition documents and provides the foundation for the long-range operational planning activities.

Operations Concept

A description of how the flight system and the ground system are used together to ensure that the concept of operation is reasonable. This might include how mission data of interest, such as engineering or scientific data, are captured, returned to Earth, processed, made available to users, and archived for future reference. It is typically developed by the operational team. (See NPR 7120.5.)

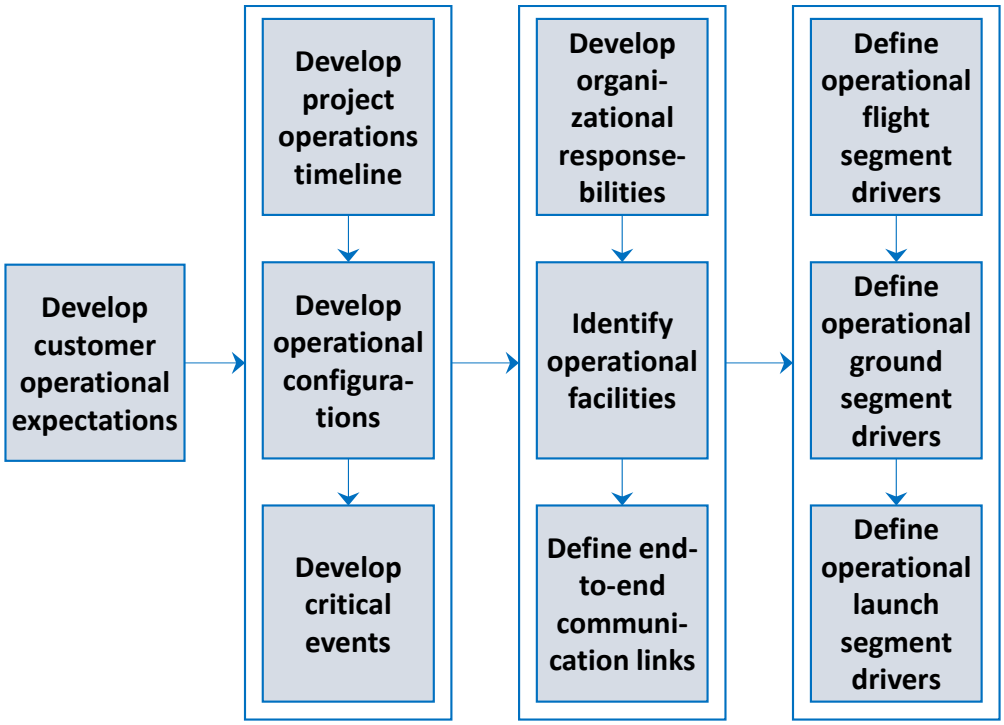


Figure 4.1-3 Typical ConOps Development for a Science Mission

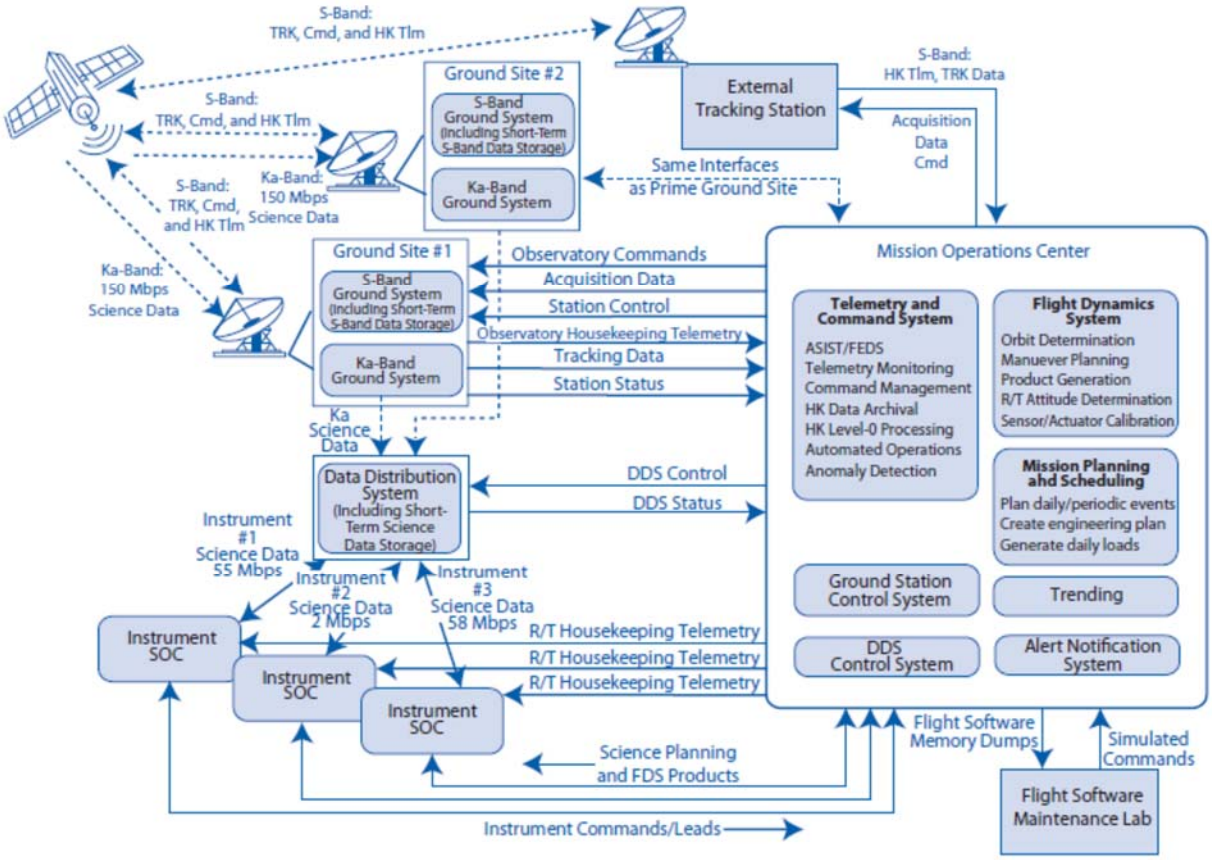


Figure 4.1-4 Example of an Associated End-to-End Operational Architecture

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

The timeline matures along with the design. It starts as a simple time-sequenced order of the major events and matures into a detailed description of subsystem operations during all major mission modes or transitions. An example of a lunar sortie timeline and DRM early in the life cycle is shown in Figures 4.1-5a and b, respectively. An example of a more detailed, integrated timeline later in the life cycle for a science mission is shown in Figure 4.1-6.

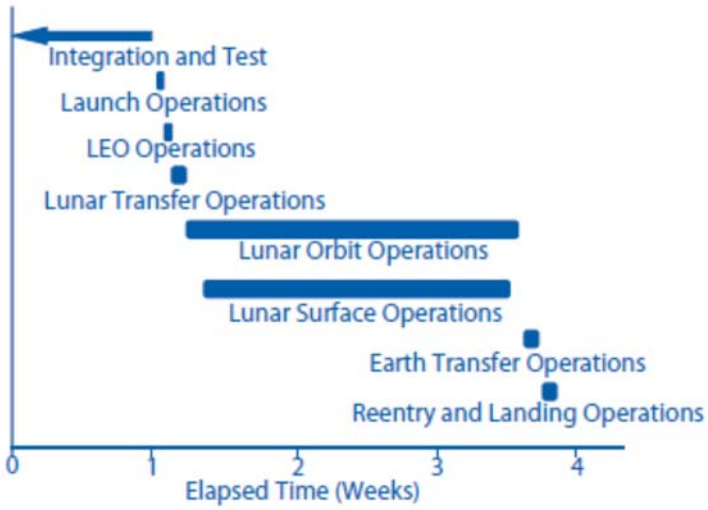


Figure 4.1-5a Example of a Lunar Sortie Timeline Developed Early in the Life Cycle

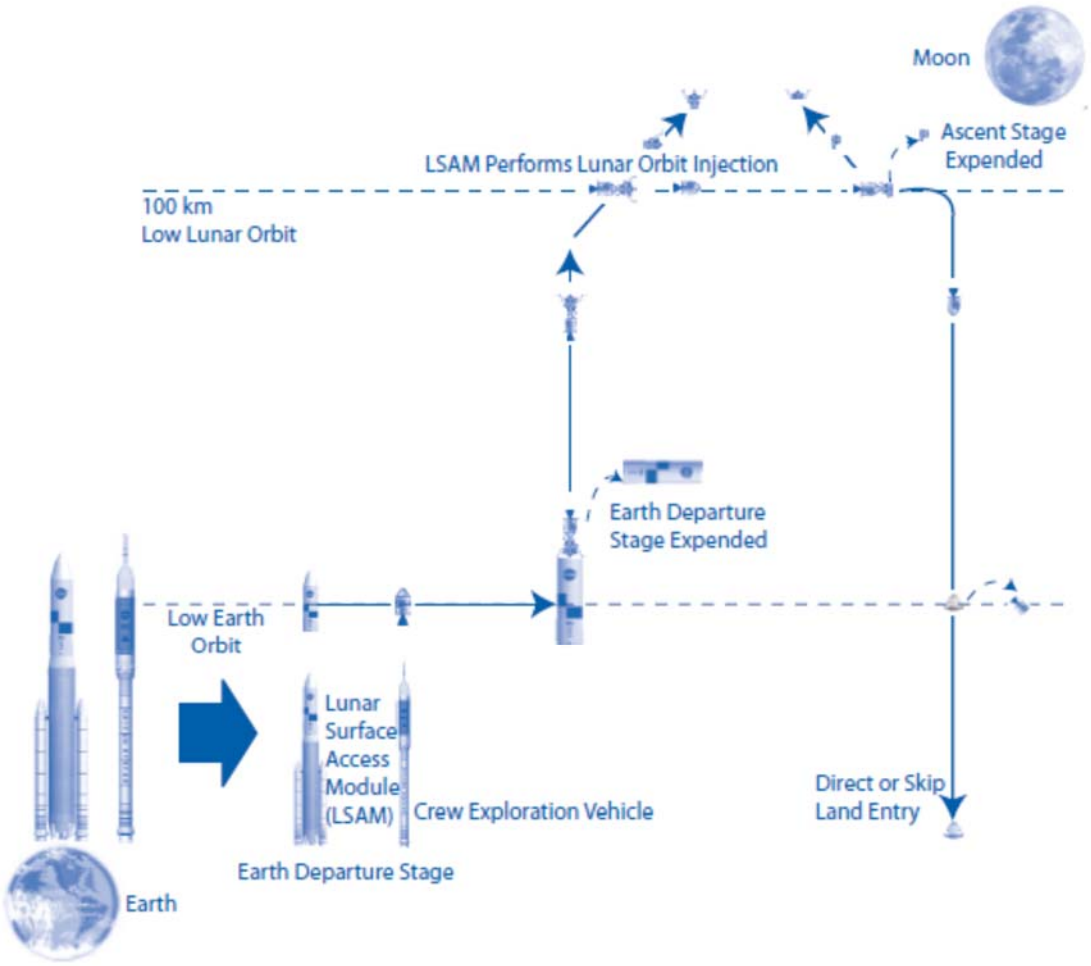


Figure 4.1-5b Example of a Lunar Sortie DRM Early in the Life Cycle

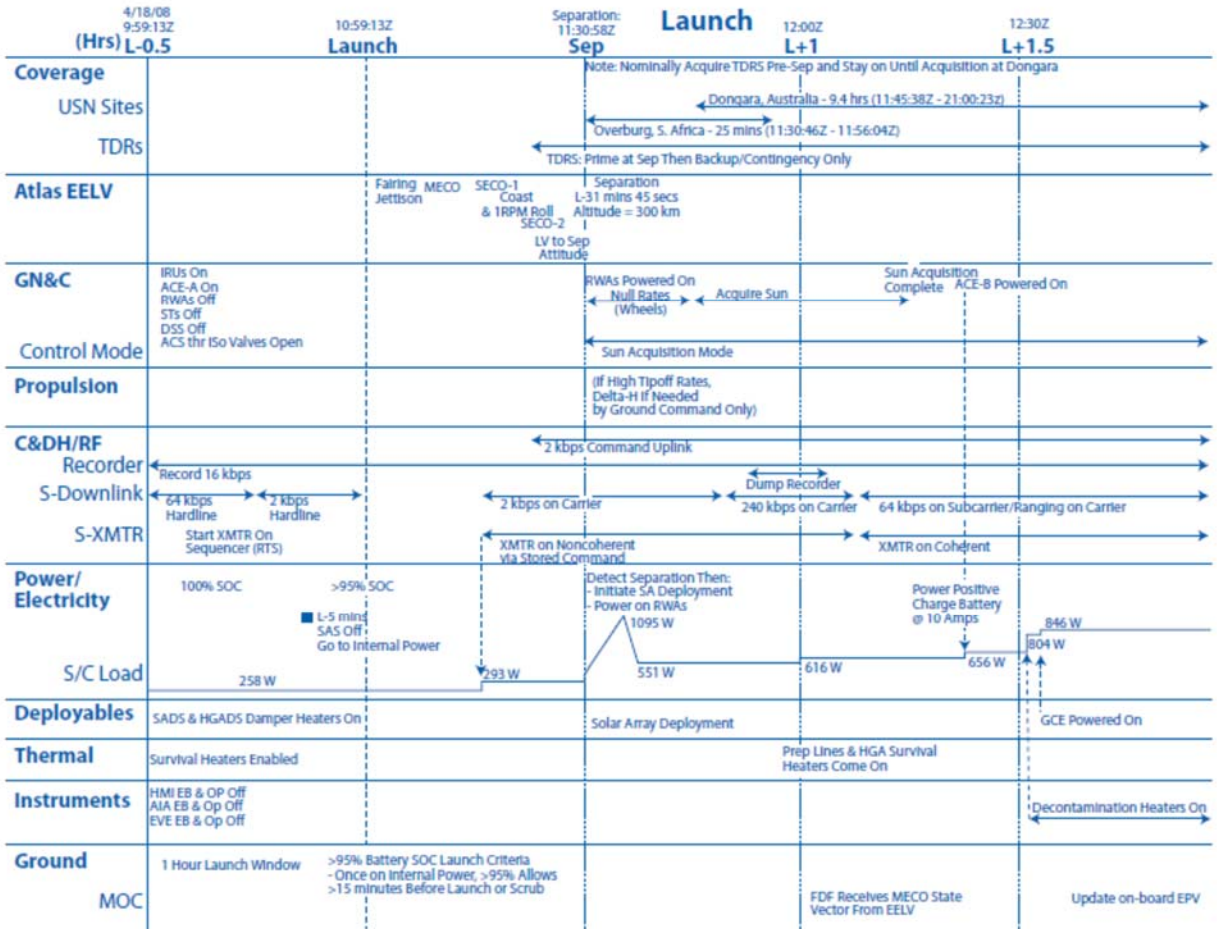


Figure 4.1‑6 Example of a More Detailed, Integrated Timeline Later in the Life Cycle for a Science Mission

An important part of the ConOps is defining the operational phases, which will span project Phases D, E, and F. The operational phases provide a time-sequenced structure for defining the configuration changes and operational activities that need to be carried out to meet the goals of the mission. For each of the operational phases, facilities, equipment, and critical events should also be included. Table 4.1-2 identifies some common examples of operational phases for a NASA mission.

Table 4.1-2 Typical Operational Phases (E and F) for a NASA Mission

Operational Phase	Description
Integration and test operations (Phase D)	**Project Integration and Test:** During the latter period of project integration and test, the system is tested by performing operational simulations during functional and environmental testing. The simulations typically exercise the end-to-end command and data system to provide a complete verification of system functionality and performance against simulated project operational scenarios.
	Launch Integration: The launch integration phase may repeat integration and test operational and functional verification in the launch-integrated configuration.
Launch operations (Phase D)	**Launch:** Launch operation occurs during the launch countdown, launch ascent, and orbit injection. Critical event telemetry is an important driver during this phase.
	Deployment: Following orbit injection, spacecraft deployment operations reconfigure the spacecraft to its orbital configuration. Typically, critical events covering solar array, antenna, and other deployments and orbit trim maneuvers occur during this phase.
	On-Orbit Checkout: On-orbit checkout is used to verify that all systems are healthy. This is followed by on-orbit alignment, calibration, and parameterization of the flight systems to prepare for science operations
Science operations (Phase E)	The majority of the operational lifetime is used to perform science operations.
Safe-hold operations (Phase E)	As a result of on-board fault management or by ground command, the spacecraft may transition to a safe-hold mode. This mode is designed to maintain the spacecraft in a power positive, thermally stable state until the fault is resolved and science operations can resume.
Anomaly resolution and maintenance operations (Phase E)	Anomaly resolution and maintenance operations occur throughout the mission. They may require resources beyond established operational resources.
Disposal operations (Phase F)	Disposal operations occur at the end of project life. These operations are used to either provide a controlled reentry of the spacecraft or a repositioning of the spacecraft to a disposal orbit. In the latter case, the dissipation of stored fuel and electrical energy is required.

4.1.2.2 Space Asset Protection

Current trends in technology proliferation, ease of accessibility to space, the globalization of space programs, and the commercialization of space systems and services has led to a fundamental change in the space environment. This fundamental change has led to a congested, contested, and competitive space environment, which increases the likelihood that U.S. space systems and the infrastructure and ground systems may be vulnerable to multiple types of threats. The reality is that there are many existing capabilities to deny, disrupt, or physically destroy NASA's space systems and the ground facilities that control them. Due to the reliance the United States has on space systems, the latest national space policies addressed in Presidential Policy Directives PPD-4 (2010) and PPD-21 (2013) require the protection of all critical space systems and supporting infrastructure.

Space asset protection is a critical systems engineering function. This concept is illustrated in Figure 4.1-7. The approach is shown in the Systems Engineering Analysis section (bottom row of the diagram) and is based on the fundamental concept: threat times susceptibility = vulnerability. This concept is based on the seminal work developed through the aircraft survivability discipline that is identified in *The Fundamentals of Aircraft Combat Survivability Analysis and Design* by Robert Ball but is applicable to any system, including spacecraft and infrastructure systems. Factors such as size, structure, concept of operations, communication links, and other subsystems create inherent weaknesses. When these inherent design features are matched with potential threats, the system's vulnerabilities become apparent. This is a fundamental concept in developing the design of a space system through its architecture. The difference in Figure 4.1-7 and for the NASA Space Asset Protection Program is that the full spectrum of threats, including intentional malicious threats, is considered in developing NASA missions.

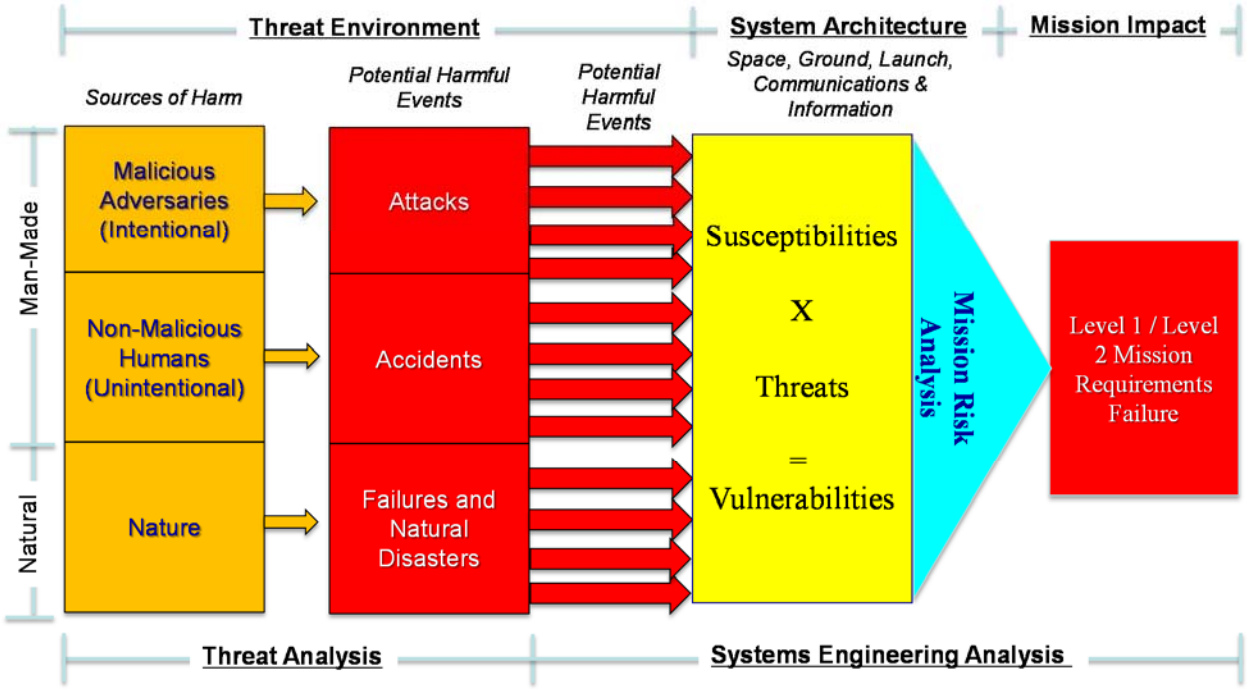


Figure 4.1-7 Space Architecture Security Environment

These threats are similar to other technical risks and are what makes space asset protection a fundamental building block for systems engineering. Space Situational Awareness (SSA) is an essential element of space asset protection that provides an in-depth knowledge and understanding of the threats posed to U.S. space systems by adversaries and the environment and is crucial in developing and employing protection measures. SSA includes collision avoidance and space weather.

NASA began working to protect its space vehicles and critical infrastructure prior to the recent national space policy statements as a good steward of the nation's resources. NASA processes are being updated to reflect current direction and agency portfolio dynamics. For example, NASA has implemented space asset protection requirements to programs and projects through

NPR 7120.5. The program and project are responsible for documenting the implementation of these requirements through the normal program and project planning process via the Program/Project Protection Plan (PPP). The first step in developing a PPP is to extract the viable threats, which are commonly referred to as protection threats, from a civil space system's threat summary and categorize them according to the NASA Risk Matrix Standard Scale. "Protection threats" are defined as any natural or man-made event, accident, or system with the ability to exploit a susceptibility of any part of a space system resulting in the potential damage, degradation, destruction, or denial of service to the mission.

The next step is to determine the vulnerabilities in a space system by fusing space system susceptibilities with protection threats, again where threat times susceptibility = vulnerability and then recommend protection strategies and countermeasures to alleviate the risks posed by the unmitigated vulnerabilities. This process is iterated during the systems engineering process between issuance of the baselined PPP and prior to each succeeding KDP. The desired end result is enhanced space system survivability. See Figure 4.1-8.

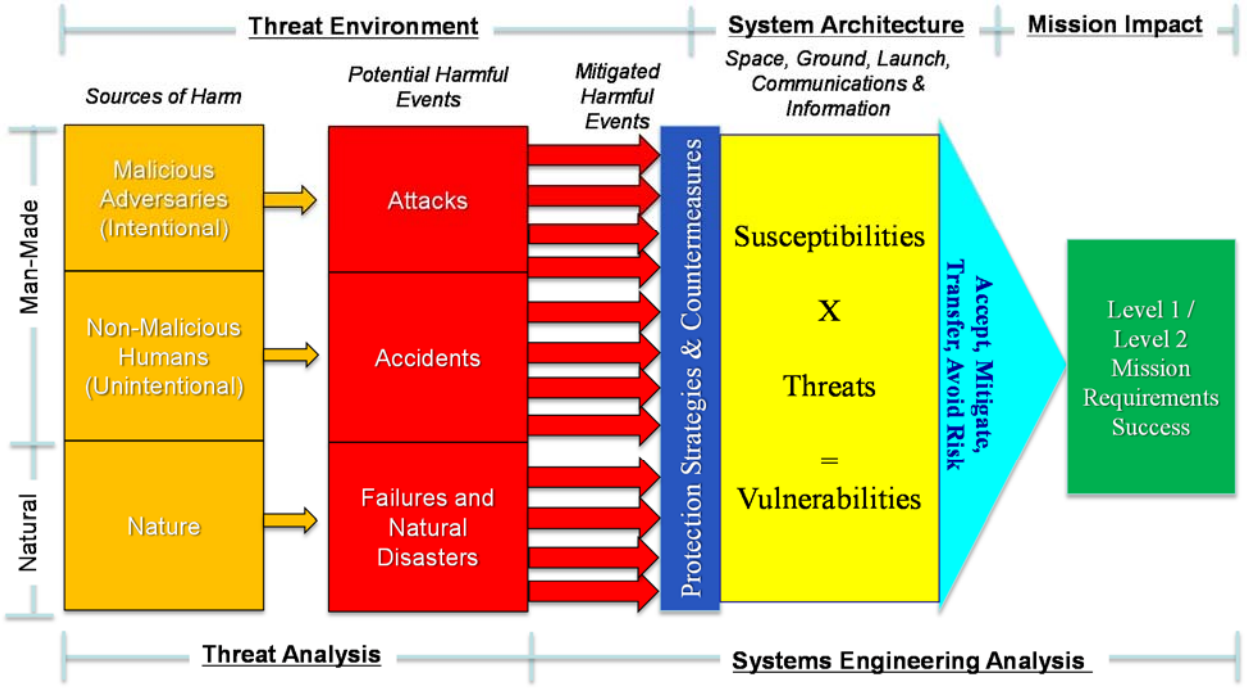


Figure 4.1-8 Security Environment with Protection Strategies and Countermeasures Considered

Protection of NASA's critical assets is being integrated as a systems engineering function through each project's chief engineer or systems engineer and the Space Asset Protection Program SMEs. In the future, it is expected that project chief engineers and Mission Systems Engineers (MSEs) will take on a greater role in drafting protection plans. Once a project's spacecraft is launched and operational, a systems engineer from the flight operations team will take on protection responsibilities.

4.1.2.3 Identifying Stakeholders throughout Phases

The Stakeholder Expectations Definition Process is used the most during Pre-Phase A and Phase A when the concepts and requirements are being developed. But this process is also useful during later phases as more stakeholders join the project.

In Phases B and C, as the design develops, more stakeholders may be identified including contractors who are hired to design and implement the system and subsystem personnel. The project team should revisit the Stakeholder Expectations Definition Process with these new stakeholders to determine if changes are needed to the baselined products, especially as the project team processes change requests that affect the requirements.

In Phase D, more stakeholders join the project, including assembly, integration, test, and operations personnel. The Stakeholder Expectations Definition Process can focus on documenting expectations for operational procedures, training of operating personnel and crew members, and logistics.

In Phase E, the project team may transition to an operations team, and the Stakeholder Expectation Definition Process is used again to revisit the expectations of operations personnel. It is also used for iterative development used in upgrades of the system.

In Phase F, new stakeholders may arrive to close out the project including archivists and tear-down personnel. The Stakeholder Expectations Definition Process is used again with these stakeholders.

4.2 Technical Requirements Definition

The Technical Requirements Definition Process transforms the stakeholder expectations into a definition of the problem and then into a complete set of validated technical requirements expressed as "shall" statements that can be used for defining a design solution for the Product Breakdown Structure (PBS) and related enabling products. The process of requirements definition is a recursive and iterative one that develops the stakeholders' requirements, product requirements, and lower level product/component requirements. The requirements should enable the description of all inputs, outputs, and required relationships between inputs and outputs, including constraints, and system interactions with operators, maintainers, and other systems. The requirements documents organize and communicate requirements to the customer and other stakeholders and the technical community.

> It is important to note that the team must not rely solely on the requirements received to design and build the system. Communication and iteration with the relevant stakeholders are essential to ensure a mutual understanding of each requirement. Otherwise, the designers run the risk of misunderstanding and implementing an unwanted solution to a different interpretation of the requirements. This iterative stakeholder communication is a critically important part of project validation. Always confirm that the right products and results are being developed.

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

4.2.1 Process Description

Figure 4.2-1 provides a typical flow diagram for the Technical Requirements Definition Process and identifies typical inputs, outputs, and activities to consider in addressing technical requirements definition.

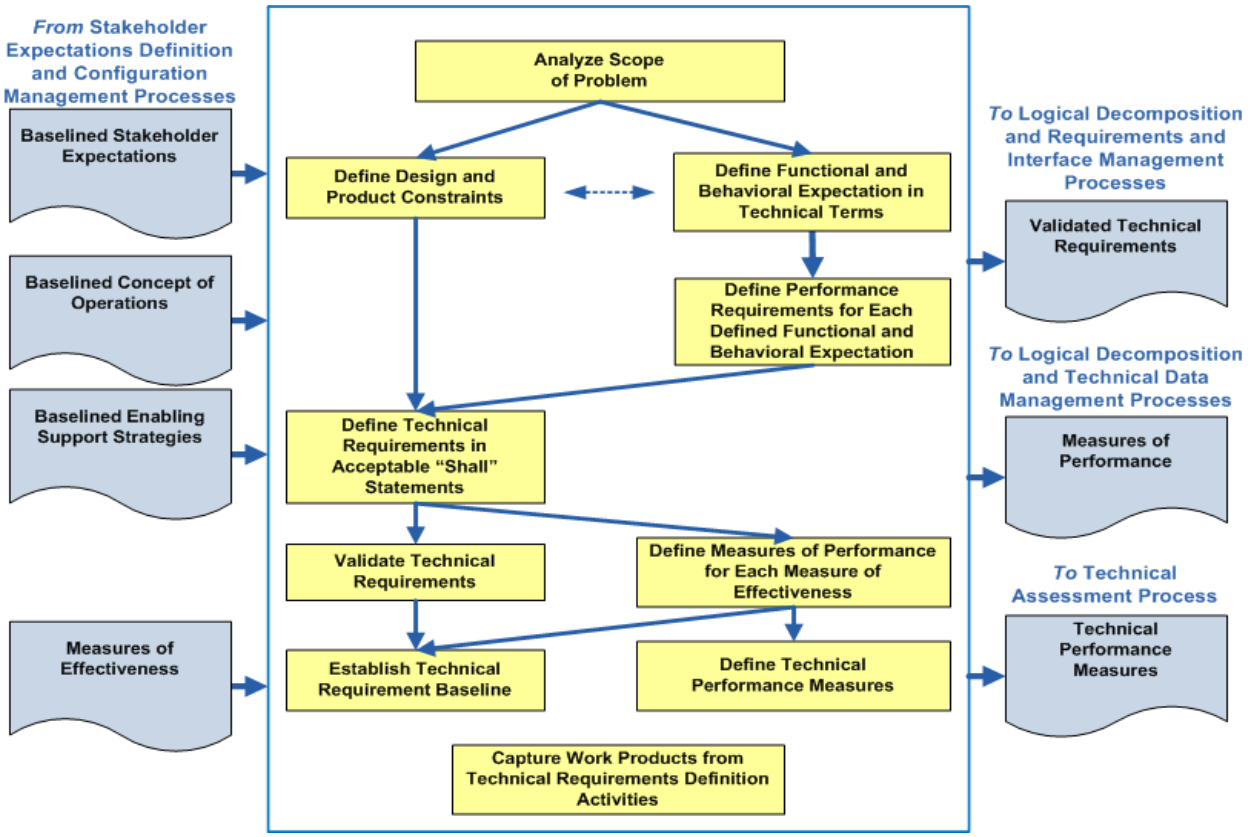


Figure 4.2-1 Technical Requirements Definition Process

4.2.1.1 Inputs

Typical inputs needed for the requirements process include the following:

- **Baselined Stakeholder Expectations:** This is the agreed-to set of stakeholder expectations (e.g., needs, goals, objectives, assumptions, constraints, external interfaces) for the product(s) of this product layer.
- **Baselined Concept of Operations:** This describes how the system will be operated during the life-cycle phases to meet stakeholder expectations. It describes the system characteristics from an operational perspective and helps facilitate an understanding of the system goals, objectives, and constraints. It includes scenarios, use cases, and/or Design Reference Missions (DRMs) as appropriate for the project. It may be in the form of a document, graphics, videos, models, and/or simulations.
- **Baselined Enabling Support Strategies:** These describe the enabling products that were identified in the Stakeholder Expectations Definition Process as needed to develop, test, produce, operate, or dispose of the end product. They also include descriptions of how the end product will be supported throughout the life cycle.
- **Measures of Effectiveness:** These MOEs were identified during the Stakeholder Expectations Definition Process as measures that the stakeholders deemed necessary to meet in order for the project to be considered a success (i.e., to meet success criteria).

Other inputs that might be useful in determining the technical requirements:

- **Human/Systems Function Allocation:** This describes the interaction of the hardware and software systems with all personnel and their supporting infrastructure. When human operators are a critical total-system component, the roles and responsibilities of the humans-in-the-system should be clearly understood. This should include all human/system interactions required for a mission including assembly, ground operations, logistics, in-flight and ground maintenance, in-flight operations, etc.

4.2.1.2 Process Activities

4.2.1.2.1 Define Constraints, Functional and Behavioral Expectations

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

- Defining constraints that the design needs to adhere to or that limit how the system will be used. The constraints typically cannot be changed based on tradeoff analyses.
- Identifying those elements that are already under design control and cannot be changed. This helps establish those areas where further trades will be made to narrow potential design solutions.
- Identifying external and enabling systems with which the system should interact and establishing physical and functional interfaces (e.g., mechanical, electrical, thermal, human, etc.).
- Defining functional and behavioral expectations for the range of anticipated uses of the system as identified in the ConOps. The ConOps describes how the system will be operated and the possible use-case scenarios.

4.2.1.2.2 Define Requirements

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

> Note: Requirements can be generated from nonobvious stakeholders and may not directly support the current mission and its objectives, but instead provide an opportunity to gain additional benefits or information that can support the Agency or the Nation. Early in the process, the systems engineer can help identify potential areas where the system can be used to collect unique information that is not directly related to the primary mission. Often outside groups are not aware of the system goals and capabilities until it is almost too late in the process.

4.2.1.2.3 Define Requirements in Acceptable Statements

Finally, the requirements should be defined in acceptable “shall” statements, which are complete sentences with a single “shall” per statement. Rationale for the requirement should also be captured to ensure the reason and context of the requirement is understood. The Key Driving

Requirements (KDRs) should be identified. These are requirements that can have a large impact on cost or schedule when implemented. A KDR can have any priority or criticality. Knowing the impact a KDR has on the design allows better management of requirements.

See appendix C for guidance and a checklist on how to write good requirements and appendix E for validating requirements. A well-written requirements document provides several specific benefits to both the stakeholders and the technical team as shown in Table 4.2-1.

Table 4.2-1 Benefits of Well-Written Requirements

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

4.2.1.2.4 Validate Technical Requirements

An important part of requirements definition is the validation of the requirements against the stakeholder expectations, the mission objectives and constraints, the concept of operations, and the mission success criteria. Validating requirements can be broken into five steps:

1. **Are the Requirements Written Correctly?** Identify and correct requirements "shall" statement format errors and editorial errors.
2. **Are the Requirements Technically Correct?** A few trained reviewers from the technical team identify and remove as many technical errors as possible before having all the relevant stakeholders review the requirements. The reviewers should check that the requirement statements (a) have bidirectional traceability to the baselined stakeholder expectations; (b)

were formed using valid assumptions; and (c) are essential to and consistent with designing and realizing the appropriate product solution form that will satisfy the applicable product life-cycle phase success criteria.

3. **Do the Requirements Satisfy Stakeholders?** All relevant stakeholder groups identify and remove defects.
4. **Are the Requirements Feasible?** All requirements should make technical sense and be possible to achieve.
5. **Are the Requirements Verifiable?** All requirements should be stated in a fashion and with enough information that it will be possible to verify the requirement after the end product is implemented.
6. **Are the Requirements Redundant or Over-specified?** All requirements should be unique (not redundant to other requirements) and necessary to meet the required functions, performance, or behaviors.

Requirements validation results are often a deciding factor in whether to proceed with the next process of Logical Decomposition or Design Solution Definition. The project team should be prepared to: (1) demonstrate that the project requirements are complete and understandable; (2) demonstrate that evaluation criteria are consistent with requirements and the operations and logistics concepts; (3) confirm that requirements and MOEs are consistent with stakeholder needs; (4) demonstrate that operations and architecture concepts support mission needs, goals, objectives, assumptions, guidelines, and constraints; and (5) demonstrate that the process for managing change in requirements is established, documented in the project information repository, and communicated to stakeholders.

4.2.1.2.5 Define MOPs and TPMs

Measures of Performance (MOPs) define the performance characteristics that the system should exhibit when fielded and operated in its intended environment. MOPs are derived from the MOEs but are stated in more technical terms from the supplier's point of view. Typically, multiple MOPs, which are quantitative and measurable, are needed to satisfy a MOE, which can be qualitative. From a verification and acceptance point of view, MOPs reflect the system characteristics deemed necessary to achieve the MOEs.

Technical Performance Measures (TPMs) are physical or functional characteristics of the system associated with or established from the MOPs that are deemed critical or key to mission success. The TPMs are monitored during implementation by comparing the current actual achievement or best estimate of the parameters with the values that were anticipated for the current time and projected for future dates. They are used to confirm progress and identify deficiencies that might jeopardize meeting a critical system requirement or put the project at cost or schedule risk.

For additional information on MOPs and TPMs, their relationship to each other and MOEs, and examples of each, see Section 6.7.2.6.2 of this guide.

4.2.1.2.6 Establish Technical Requirement Baseline

Once the technical requirements are identified and validated to be good (clear, correct, complete, and achievable) requirements, and agreement has been gained by the customer and key

stakeholders, they are baselined and placed under configuration control. Typically, a System Requirements Review (SRR) is held to allow comments on any needed changes and to gain agreement on the set of requirements so that it may be subsequently baselined. For additional information on the SRR, see Section 6.7.

4.2.1.2.7 Capture Work Products

The work products generated during the above activities should be captured along with key decisions that were made, any supporting decision rationale and assumptions, and lessons learned in performing these activities**.**

4.2.1.3 Outputs

- **Validated Technical Requirements:** This is the approved set of requirements that represents a complete description of the problem to be solved and requirements that have been validated and approved by the customer and stakeholders. Examples of documents that capture the requirements are a System Requirements Document (SRD), Project Requirements Document (PRD), Interface Requirements Document (IRD), and a Software Requirements Specification (SRS).

- **Measures of Performance:** These are the identified quantitative measures that, when met by the design solution, help ensure that one or more MOEs will be satisfied. There may be two or more MOPs for each MOE. See Section 6.7.2.6.2 for further details.
- **Technical Performance Measures:** These are the set of performance measures that are monitored and trended by comparing the current actual achievement of the parameters with that expected or required at the time. TPMs are used to confirm progress and identify deficiencies. See Section 6.7.2.6.2 for further details.

4.2.2 Technical Requirements Definition Guidance

4.2.2.1 Types of Requirements

A complete set of project requirements includes those that are decomposed and allocated down to design elements through the PBS and those that cut across product boundaries. Requirements allocated to the PBS can be functional requirements (what functions need to be performed), performance requirements (how well these functions should be performed), and interface requirements (product to product interaction requirements). Crosscutting requirements include environmental, safety, human factors, and those that originate from the "-ilities" and from Design and Construction (D&C) standards. Figure 4.2-2 is a general overview on the flow of requirements, what they are called, and who is responsible (owns) for approving waivers.

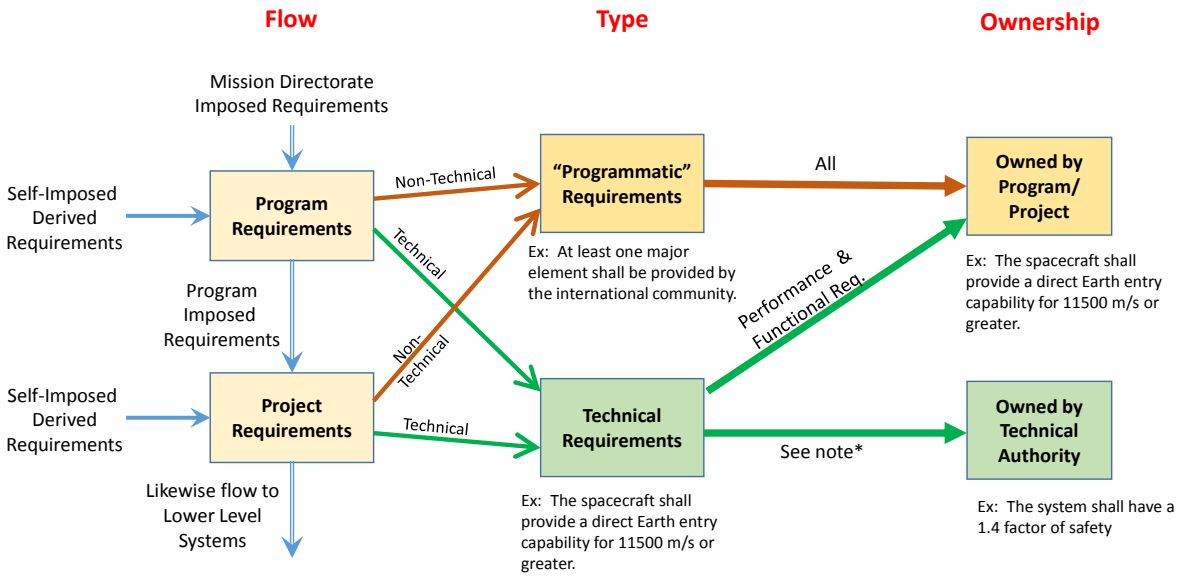


Figure 4.2-2 Flow, Type and Ownership of Requirements

Functional, performance, and interface requirements are very important but do not constitute the entire set of technical requirements necessary for project success. The space segment design elements should also survive and continue to perform in the project environment. These environmental drivers include radiation, thermal, acoustic, mechanical loads, contamination, radio frequency and others. In addition, reliability requirements drive fault management through design choices in design robustness, failure tolerance, and redundancy. Safety requirements drive design choices in providing diverse functional redundancy. Figure 4.2-3 shows the organization of types of requirements described in this chapter.

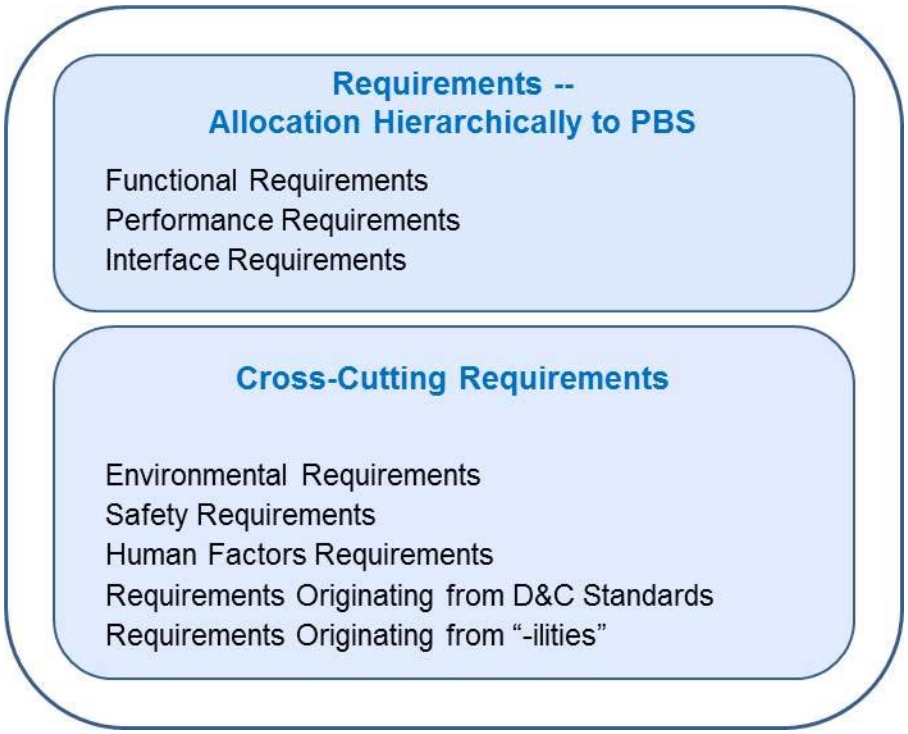


Figure 4.2-3 Types of Requirements

4.2.2.2 Product Breakdown Structure Requirements

4.2.2.2.1 Functional Requirements

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

- Functional requirements define what functions need to be performed to accomplish the objectives.
- Performance requirements define how well the system needs to perform the functions.

It is helpful to walk through the ConOps and scenarios asking the following types of questions: what functions need to be performed; where, how often, by whom, and under what operational

and environmental conditions do they need to be performed, etc. Thinking through this process often reveals additional functional requirements.

4.2.2.2.2 Performance Requirements

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

Example of Functional and Performance Requirements

Initial Function Statement

The Thrust Vector Controller (TVC) shall provide vehicle control about the pitch and yaw axes.

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

Functional Requirements with Associated Performance Requirements

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

Wherever possible, the performance requirements are defined in terms of (1) a threshold value (the minimum acceptable value needed for the system to carry out its mission) and (2) the baseline level of performance desired. Going below the threshold value would require a descope of the project. Sometimes additional functionality of a design over the threshold value can be had at little or no additional cost. When this occurs, the customer and project team may agree to make the new functionality part of the baseline requirements. Thus, specifying performance in terms of thresholds and baseline requirements provides the system designers with trade space in which to investigate alternative designs.

Be careful not to make performance requirements too restrictive. For example, for a system that must be able to run on rechargeable batteries, if the performance requirements specify that the time to recharge should be less than 3 hours when a 12-hour recharge time would be sufficient, potential design solutions are eliminated. In the same sense, if the performance requirements specify that a weight must be within ±0.5 kg, when ±2.5 kg is sufficient, metrology cost will increase without adding value to the product.

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

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

4.2.2.2.3 Interface Requirements

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

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

As the Technical Requirements Definition Process continues, the interface diagram should be revisited and the documented interface requirements refined to include newly identified interfaces information for requirements both internal and external. More information regarding interfaces can be found in Section 6.3.

4.2.2.3 Crosscutting Requirements

A subset of non-functional requirements is applied across the systems rather than down through the Product Breakdown Structure (PBS). Examples are provided in this section for environmental requirements, safety requirements, human factors engineering requirements, and reliability requirements. These are representative of types of crosscutting requirements, but there can be many more domains and disciplines providing requirements depending on the scope and nature of the program or project.

Each design element should survive and continue to perform in the project environment. Environmental requirements include limits for radiation, thermal, acoustic, mechanical loads, contamination, radio frequency, and others. Safety requirements drive design choices in providing diverse functional redundancy. Human factors engineering requirements ensure that human capabilities and limitations are considered for proper system operations. Other non-functional requirements, sometimes called the “-ilities”, may also affect design choices. These may include producibility, reliability, maintainability, availability, upgradeability, and others. Design and Construction (D&C) standards often flow to crosscutting requirements.

4.2.2.3.1 Environmental Requirements

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

environments, develop design guidance, and establish a margin philosophy against the expected environments.

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

External and internal environment concerns that should be addressed include acceleration, vibration, shock, static loads, acoustic, thermal, contamination, crew-induced loads, total dose radiation/radiation effects, Single-Event Effects (SEEs), surface and internal charging, orbital debris, atmospheric (atomic oxygen) control and quality, attitude control system disturbance (atmospheric drag, gravity gradient, and solar pressure), magnetic, pressure gradient during launch, microbial growth, and radio frequency exposure on the ground and on orbit.

The requirements structure should address the specialty engineering disciplines that apply to the mission environments across project elements. These discipline areas levy requirements on system elements regarding Electromagnetic Interference, Electromagnetic Compatibility (EMI/EMC), grounding, radiation and other shielding, contamination protection, human factors and environmental health requirements, and reliability.

4.2.2.3.2 Safety Requirements

NASA uses the term "safety" broadly to include human (public and workforce), environmental, and asset safety. There are two types of safety requirements: deterministic and risk-informed. A "deterministic safety requirement" is the qualitative or quantitative definition of a threshold of action or performance that should be met by a mission-related design item, system, or activity for that item, system, or activity to be acceptably safe. Examples of deterministic safety requirements are incorporation of safety devices (e.g., building physical hardware stops into the system to prevent the hydraulic lift/arm from extending past allowed safety height and length limits); limits on the range of values a system input variable is allowed to take on; and limit checks on input commands to ensure they are within specified safety limits or constraints for that mode or state of the system (e.g., only allowing the command to retract the landing gear if the airplane is in the airborne state). Human errors can cause safety hazards and safety assessments should include consideration of hazards and corrective actions. For those components identified as "safety critical," the corresponding fault management requirements include functional redundancy or failure tolerance to allow the system to meet its requirements in the presence of one or more failures or to take the system to a safe state with reduced functionality (e.g., one fault tolerant computer processors, safe-state backup processor); detection and automatic system shutdown if specified values (e.g., temperature) exceed prescribed safety limits; use of only a subset that is approved for safety-critical software of a particular computer language; caution or warning devices; and safety procedures depending on the mission or payload risk classification. A "risk-informed safety requirement" is a requirement that has been established, at least in part, on the basis of the consideration of safety-related TPMs and their associated uncertainty. An example of a risk-informed safety requirement is the Probability of Loss of Crew (P(LOC)) not exceeding a certain value "p" with a certain confidence level. These requirements might also be the established human tolerance limits for environmental, mechanical, or electrical hazards. Meeting safety requirements involves identification and elimination of hazards, reducing the

likelihood of the accidents associated with hazards, or reducing the impact from the hazard associated with these accidents to within acceptable levels. (For additional information concerning safety, see, for example, NPR 8705.2, Human-Rating Requirements for Space Systems, NPR 8715.3, NASA General Safety Program Requirements, and *NASA-STD-8719.13, Software Safety Standard*.)

4.2.2.3.3 Human Factors Engineering Requirements

In aeronautics, space flight, robotics missions, and other NASA endeavors, the human, as operator and as maintainer, is a critical component of the mission and system design. Human capabilities and limitations should enter into designs in the same way that the properties of materials and characteristics of electronic components do. For human space flight, many Human Factors (HF) requirements flow from *NASA-STD-3001, NASA Space Flight Human System Standards* and are explained further in the companion handbook *NASA-SP-2010-3407, Human Integration Design Handbook*.

HF-related goals and constraints are included in the overall plans for the system during requirements definition. HF-related issues, design risks, and tradeoffs pertinent to each human-system component are documented as part of the project's requirements so they are adequately addressed during the design phase.

By including as stakeholders not only those who are specifying the system to be built but also those who will be utilizing the system when it is put into operation, requirements are both generated from the top down (what the system is intended to accomplish) and from the bottom up (how the system is anticipated to function).

All the hardware and software requirements should consider the role of the human in the system and the type of tasks the human is expected to perform. The difference between a passive passenger and an active operator will drive major design decisions. The number of crewmembers will drive subsequent decisions about habitable volume and storage and about crew time available for operations and maintenance.

Appropriate system design defines the environmental conditions in which the system will operate to support humans and any factors that impact the human users. The requirements may need to specify acceptable atmospheric conditions, including temperature, pressure, composition, and humidity, or address acceptable ranges of acoustic noise, vibration, acceleration, and gravitational forces. The requirements may also indicate when the use of protective clothing is required or how to accommodate adverse or emergency conditions outside the range of normal operations.

Appropriate system design requires not only consideration of environmental factors, such as the physical environment or available technologies, but also consideration of the human components; e.g., physical and cognitive abilities. For example, issues with fatigue and autonomy that may be associated with aeronautics and space travel can heavily impact human performance and should also be considered in design requirements.

4.2.2.3.4 Reliability Requirements

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

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

Reliability requirements ensure that the system (and subsystems, e.g., software and hardware) can perform in the predicted environments and conditions as expected throughout the mission and that the system has the ability to withstand certain numbers and types of faults, errors, or failures (e.g., withstand vibration, predicted data rates, command and/or data errors, single-event upsets, and temperature variances to specified limits).

Environments can include ground (transportation and handling), launch, on-orbit (Earth or other), planetary, reentry, and landing, or they might be for software within certain modes or states of operation. Reliability can be affected by human errors as well as failures in the engineered systems (mechanical, electrical, hydraulic, etc.). Reliability should consider the potential for human error (in coordination with Human Systems Integration (HSI) – see Section 7.9) and then validate assumptions using test subjects. Reliability addresses design and verification requirements to meet the requested level of operation as well as fault and/or failure tolerance for all expected environments and conditions. Reliability requirements provide inputs to design choices for fault/failure prevention, detection, isolation, and recovery functions, and relevant operator/crew notifications.

4.2.2.4 Requirements Decomposition, Allocation, and Validation

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

The traceability of requirements to the lowest level ensures that each requirement is necessary to meet the stakeholder expectations. Requirements that are not allocated to lower levels or are not implemented at a lower level can result in a design that does not meet objectives. Conversely, lower level requirements that are not traceable to higher level requirements can result in an overdesign that is not justified. This hierarchical flowdown is illustrated in Figure 4.2-4.

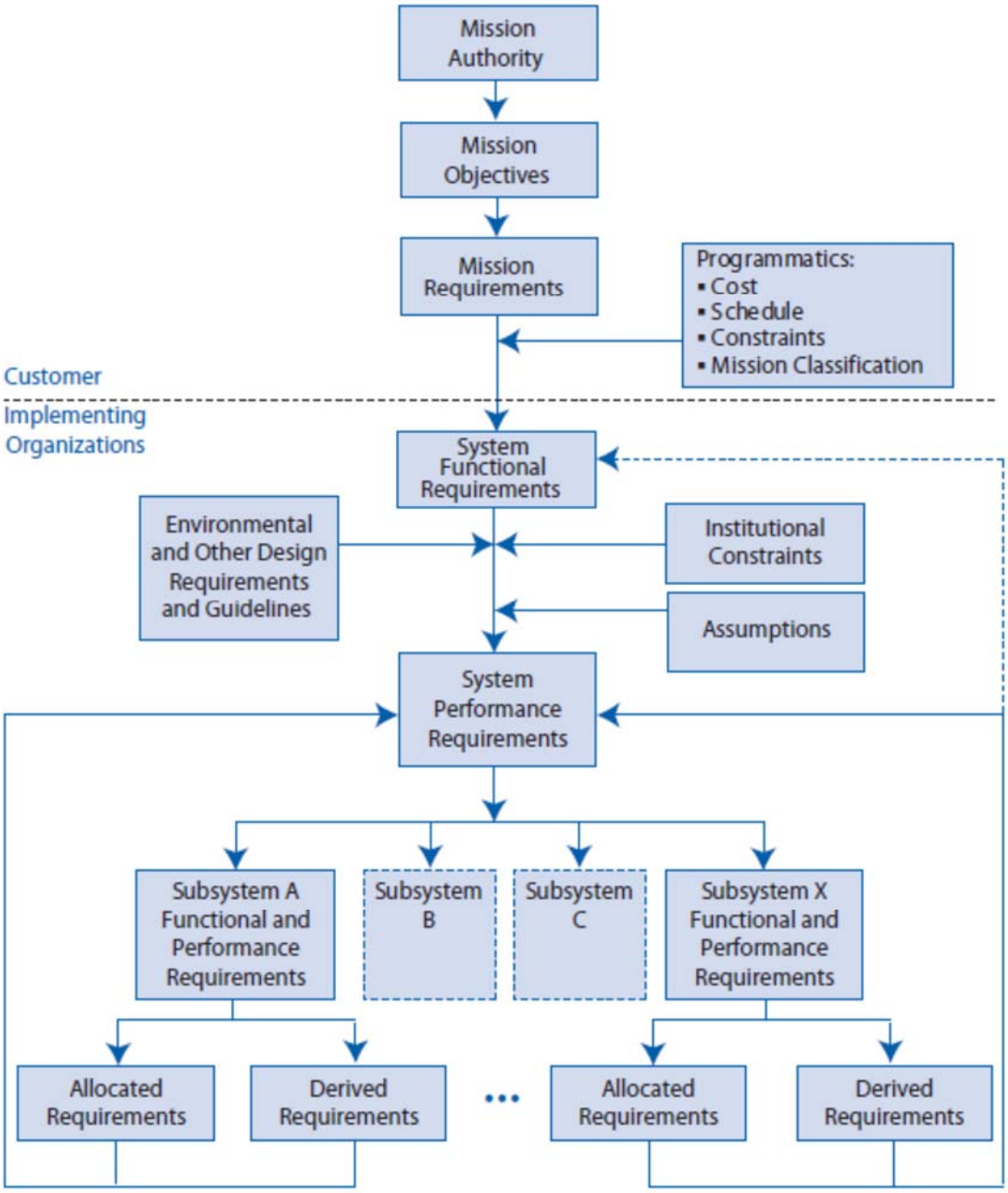


Figure 4.2-4 The Flowdown of Requirements

Figure 4.2-5 is an example of how science pointing requirements are successively decomposed and allocated from the top down for a typical science mission. It is important to understand and document the relationship between requirements. This will reduce the possibility of misinterpretation and the possibility of an unsatisfactory design and associated cost or schedule increase.

Throughout Phases A and B, changes in requirements and constraints will occur. It is imperative that all changes be thoroughly evaluated to determine the impacts on both higher and lower hierarchical levels. All changes should be subjected to a review and approval cycle as part of a

formal change control process to maintain traceability and to ensure the impacts of any changes are fully assessed for all parts of the system. A more formal change control process is required if the mission is very large and involves more than one Center or crosses other jurisdictional or organizational boundaries.

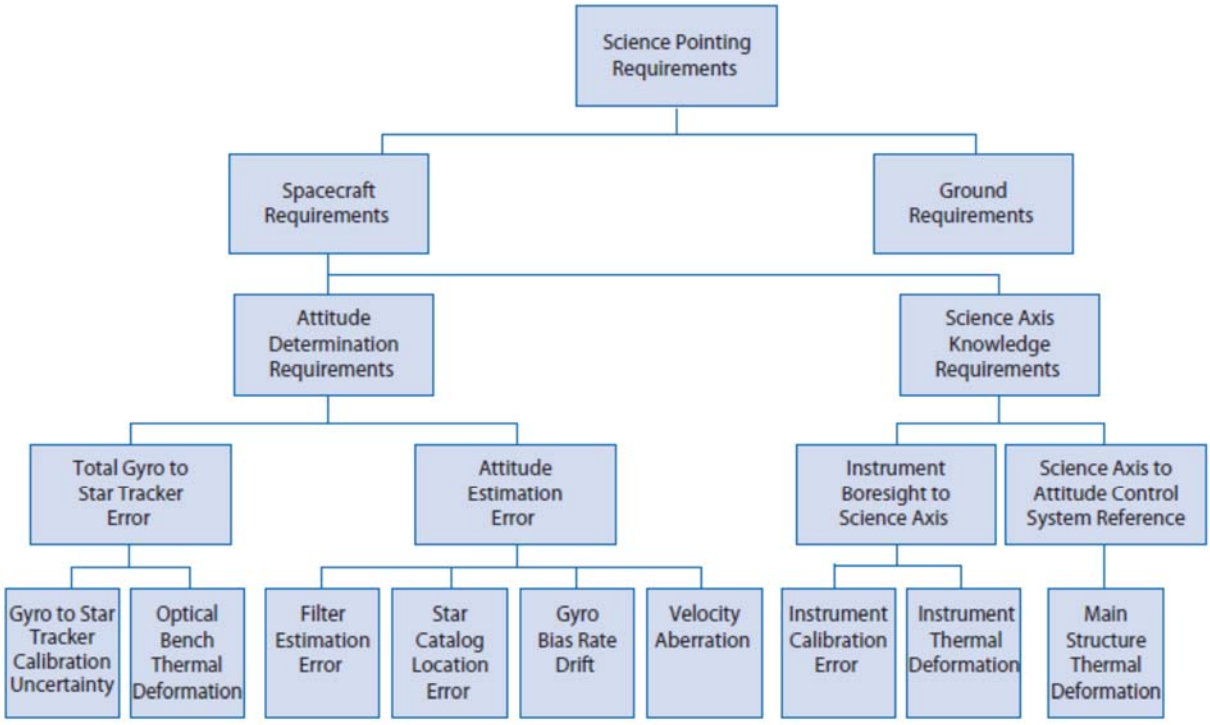


Figure 4.2-5 Allocation and Flowdown of Science Pointing Requirements

4.2.2.5 Capturing Requirements and the Requirements Database

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

The method of verification should also be thought through and captured for each requirement at the time it is developed. Some programs/projects capture those methodologies in the form of verification requirements. The verification method includes test, inspection, analysis, and demonstration. Any new or derived requirements that are uncovered during determination of the verification method need to be documented. An example is requiring an additional test port to give visibility to an internal signal during integration and test. If a requirement cannot be verified, then either it should not be a requirement or the requirement statement needs to be rewritten. For example, the requirement to "minimize noise" is vague and cannot be verified. If the requirement is restated as "the noise level of the component X shall remain under Y decibels," then it is clearly verifiable. Examples of the types of metadata are provided in Table 4.2-2. A lack of this information may result in NASA and the contractor having different expectations with respect to successfully verifying the requirement.

Table 4.2-2 Requirements Metadata

Item	Function
Requirement ID	Provides a unique numbering system for sorting and tracking.
Rationale	Provides additional information to help clarify the intent of the requirements at the time they were written. (See “Rationale” box below on what should be captured.)
Traced from	Captures the bidirectional traceability between parent requirements and lower level (derived) requirements and the relationships between requirements.
Owner	Person or group responsible for writing, managing, and/or approving changes to this requirement.
Verification method	Captures the method of verification (test, inspection, analysis, demonstration) and should be determined as the requirements are developed.
Verification lead	Person or group assigned responsibility for verifying the requirement.
Verification level	Specifies the level in the hierarchy at which the requirements will be verified (e.g., system, subsystem, element).

The requirements database is an extremely useful tool for capturing the requirements and the associated metadata and for showing the bidirectional traceability between requirements. The database evolves over time and could be used for tracking status information related to requirements such as To Be Determined (TBD)/To Be Resolved (TBR) status, resolution date, and verification status. Each project should decide what metadata will be captured. The database is usually in a central location that is made available to the entire project team. (See appendix D for a sample requirements verification matrix.)

Rationale

The rationale should be kept up to date and include the following information:

- **Reason for the Requirement:** Often the reason for the requirement is not obvious, and it may be lost if not recorded as the requirement is being documented. The reason may point to a constraint or concept of operations. If there is a clear parent requirement or trade study that explains the reason, then it should be referenced.
- **Document Assumptions:** If a requirement was written assuming the completion of a technology development program or a successful technology mission, the assumption should be documented.
- **Document Relationships:** The relationships with the product’s expected operations (e.g., expectations about how stakeholders will use a product) should be documented. This may be done with a link to the ConOps.
- **Document Design Constraints:** Constraints imposed by the results from decisions made as the design evolves should be documented. If the requirement states a method of implementation, the rationale should state why the decision was made to limit the solution to this one method of

4.2.2.6 Technical Standards

4.2.2.6.1 Importance of Standards Application

Standards provide a proven basis for establishing common technical requirements across a program or project to avoid incompatibilities and ensure that at least minimum requirements are

met. When used effectively, common standards can also lower implementation cost as well as costs for inspection, common supplies, etc. Standards are based on lessons learned and best practices. Typically, standards (and specifications) are used throughout the product life cycle to establish design requirements and margins, materials and process specifications, test methods, and interface specifications (e.g., D&C standards). Standards are not self-invoking and need to be called out or used as requirements (and guidelines) for design, fabrication, verification, validation, acceptance, operations, and maintenance.

4.2.2.6.2 Selection of Standards

NASA policy for technical standards is provided in NPR 7120.10, Technical Standards for NASA Programs and Projects, which addresses selection, tailoring, application, and control of standards. In general, the order of authority among standards for NASA programs and projects is as follows:

1. Standards mandated by law (e.g., environmental standards),
2. National or international voluntary consensus standards recognized by industry,
3. Other Government standards,
4. NASA technical standards.

NASA may also designate mandatory or "core" standards that are to be applied to all programs where technically applicable. Waivers to designated core standards need to be justified and approved at the Agency level unless otherwise delegated. Standards are owned by the appropriate Technical Authority.

4.3 Logical Decomposition

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

The Logical Decomposition Process is used to:

- Improve understanding of the defined technical requirements and the relationships among the requirements (e.g., functional, performance, behavioral, and temporal etc.), and
- Decompose the parent requirements into a set of logical decomposition models and their associated sets of derived technical requirements for input to the Design Solution Definition Process.

4.3.1 Process Description

Figure 4.3-1 provides a typical flow diagram for the Logical Decomposition Process and identifies typical inputs, outputs, and activities to consider in addressing logical decomposition.

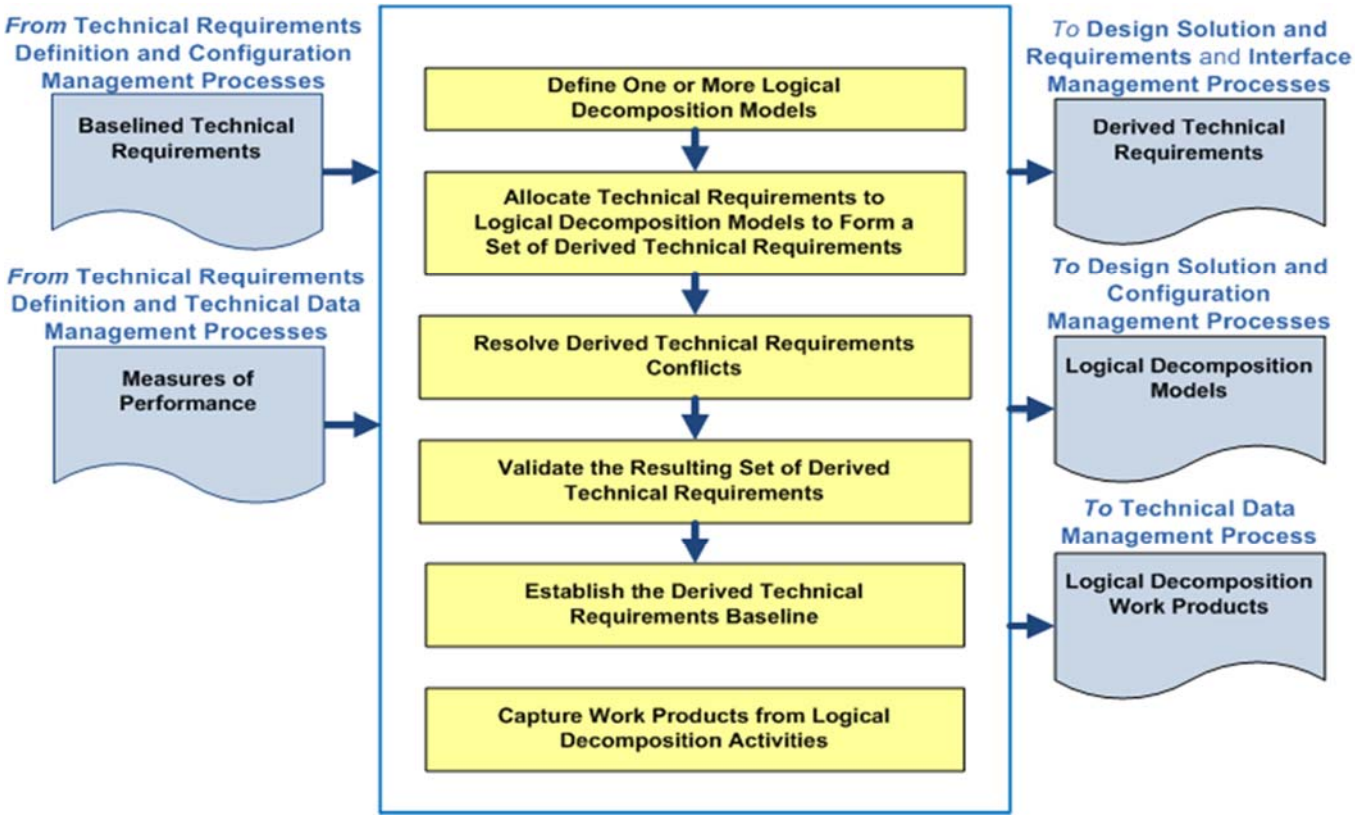


Figure 4.3-1 Logical Decomposition Process

4.3.1.1 Inputs

Typical inputs needed for the Logical Decomposition Process include the following:

- **Technical Requirements:** A validated set of requirements that represent a description of the problem to be solved, have been established by functional and performance analysis, and have been approved by the customer and other stakeholders. Examples of documents that capture the requirements are an SRD, PRD, and IRD.
- **Technical Measures:** An established set of measures based on the expectations and requirements that will be tracked and assessed to determine overall system or product effectiveness and customer satisfaction. These measures are MOEs, MOPs, and a special subset of these called TPMs. See Section 6.7.2.6.2 for further details.

4.3.1.2 Process Activities

4.3.1.2.1 Define One or More Logical Decomposition Models

The key first step in the Logical Decomposition Process is establishing the system architecture model. The system architecture activity defines the underlying structure and relationships of hardware, software, humans-in-the-loop, support personnel, communications, operations, etc., that provide for the implementation of Agency, mission directorate, program, project, and subsequent levels of the requirements. System architecture activities drive the partitioning of system elements and requirements to lower level functions and requirements to the point that design work can be accomplished. Interfaces and relationships between partitioned subsystems and elements are defined as well.

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

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

Focusing on the project's ends, top-level (or parent) requirements, and constraints, the system architect should develop at least one, but preferably multiple, concept architectures capable of achieving program objectives. Each architecture concept involves specification of the functional elements (what the pieces do), their relationships to each other (interface definition), and the ConOps, i.e., how the various segments, subsystems, elements, personnel, units, etc., will operate as a system when distributed by location and environment from the start of operations to the end of the mission.

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

In the early stages of development, multiple concepts are generated. Cost and schedule constraints will ultimately limit how long a program or project can maintain multiple architectural concepts. For all NASA programs, architecture design is completed during the Formulation Phase. For most NASA projects (and tightly coupled programs), the baselining of a single architecture happens during Phase A. Architectural changes at higher levels occasionally occur as decomposition to lower levels produces complexity in design, cost, or schedule that necessitates such changes. However, as noted in Figure 2.5-3, the later in the development process that changes occur, the more expensive they become.

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

4.3.1.2.2 Allocate Technical Requirements, Resolve Conflicts, and Baseline

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

Three key steps in performing functional analysis are:

1. Translate top-level requirements into functions that should be performed to accomplish the requirements.
2. Decompose and allocate the functions to lower levels of the product breakdown structure.
3. Identify and describe functional and subsystem interfaces.

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

The process is recursive and iterative and continues until all desired levels of the architecture/system have been analyzed, defined, and baselined. There will almost certainly be

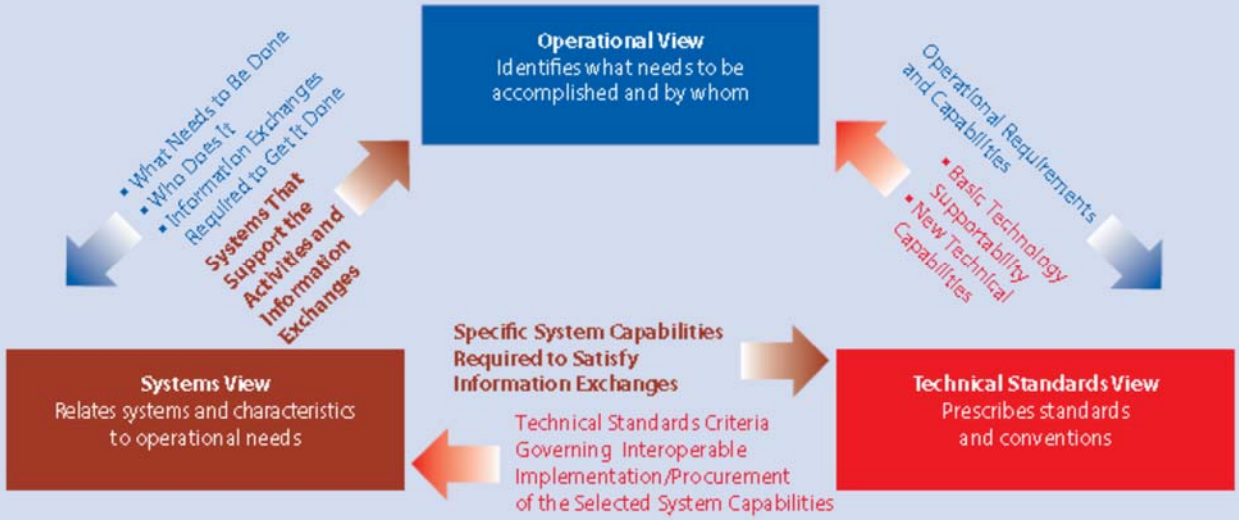
alternative ways to decompose functions. For example, there may be several ways to communicate with the crew: Radio Frequency (RF), laser, Internet, etc. Therefore, the outcome is highly dependent on the creativity, skills, and experience of the engineers doing the analysis. As the analysis proceeds to lower levels of the architecture and system, and the system is better understood, the systems engineer should keep an open mind and a willingness to go back and change previously established architecture and system requirements. These changes will then have to be decomposed down through the architecture and subfunctions again with the recursive process continuing until the system is fully defined with all of the requirements understood and known to be viable, verifiable, and internally consistent. Only at that point should the system architecture and requirements be baselined.

4.3.1.2.3 Capture Work Products

The other work products generated during the Logical Decomposition Process should be captured along with key decisions made, supporting decision rationale and assumptions, and lessons learned in performing the activities.

DOD Architecture Framework

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



An architecture can be understood as "the structure of components, their relationships, and the principles and guidelines governing their design and evolution over time."* To describe an architecture, the DODAF defines several views: operational, systems, and technical standards. In addition, a dictionary and summary information are also required.

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

*Definition based on Institute of Electrical and Electronics Engineers (IEEE) STD 610.12. Source: DOD, *DOD Architecture Framework*.

4.3.1.3 Outputs

Typical outputs of the Logical Decomposition Process include the following:

- **Logical Decomposition Models:** These models define the relationship of the requirements and functions and their behaviors. They include the system architecture models that define the underlying structure and relationship of the elements of the system (e.g., hardware, software, humans-in-the-loop, support personnel, communications, operations, etc.) and the basis for the partitioning of requirements into lower levels to the point that design work can be accomplished.

- **Derived Technical requirements:** These are requirements that arise from the definitions of the selected architecture that were not explicitly stated in the baselined requirements that served as an input to this process. Both the baselined and derived requirements are allocated to the system architecture and functions.
- **Logical Decomposition Work Products:** These are the other products generated by the activities of this process.

4.3.2 Logical Decomposition Guidance

4.3.2.1 Product Breakdown Structure

The decompositions represented by the PBS and the Work Breakdown Structure (WBS) form important perspectives on the desired product system. The WBS is a hierarchical breakdown of the work necessary to complete the project. See Section 6.1.2.1 for further information on WBS development. The WBS contains the PBS, which is the hierarchical breakdown of the products such as hardware items, software items, and information items (documents, databases, etc.). The PBS is used during the Logical Decomposition Process and the functional analysis processes. The PBS should be carried down to the lowest level for which there is a cognizant engineer or manager. Figure 6.1-4 is an example of a PBS.

4.3.2.2 Functional Analysis Techniques

Although there are many techniques available to perform functional analysis, some of the more popular are (1) Functional Flow Block Diagrams (FFBDs) to depict task sequences and relationships, (2) N2 diagrams (or N x N interaction matrix) to identify interactions or interfaces between major factors from a systems perspective, and (3) TimeLine Analyses (TLAs) to depict the time sequence of time-critical functions. Refer to appendix F of details of these techniques.

4.4 Design Solution Definition

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

4.4.1 Process Description

Figure 4.4-1 provides a typical flow diagram for the Design Solution Definition Process and identifies typical inputs, outputs, and activities to consider in addressing design solution definition.

4.4.1.1 Inputs

There are several fundamental inputs needed to initiate the Design Solution Definition Process:

- **Technical Requirements:** These are the customer and stakeholder needs that have been translated into a complete set of validated requirements for the system, including all interface requirements.
- **Logical Decomposition Models:** Requirements are analyzed and decomposed by one or more different methods (e.g., function, time, behavior, data flow, states, modes, system architecture, etc.) in order to gain a more comprehensive understanding of their interaction and behaviors. (See the definition of a model in appendix B.)

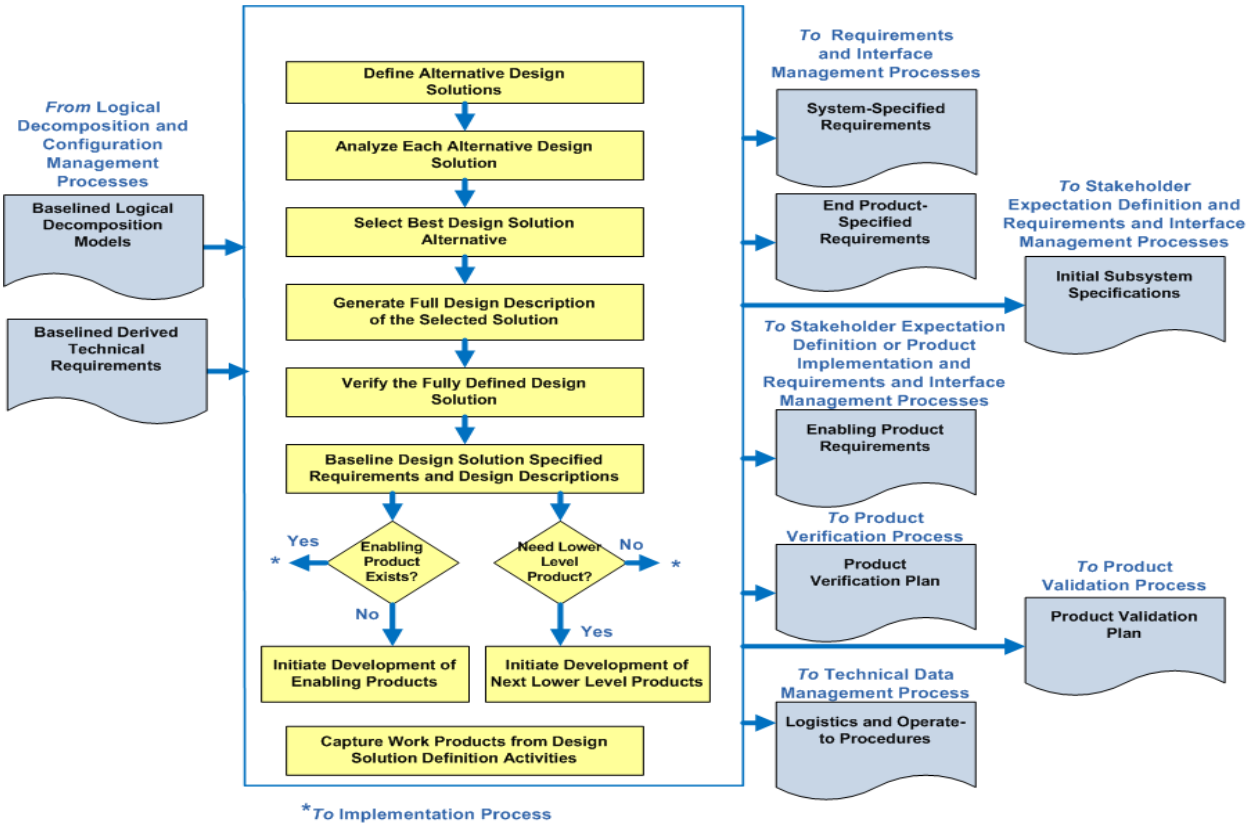


Figure 4.4-1 Design Solution Definition Process

4.4.1.2 Process Activities

4.4.1.2.1 Define Alternative Design Solutions

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

To obtain assessments that are crisp enough to facilitate good decisions, it is often necessary to delve more deeply into the space of possible designs than has yet been done, as illustrated in Figure 4.4-2. It should be realized, however, that this illustration represents neither the project life cycle, which encompasses the system development process from inception through disposal, nor the product development process by which the system design is developed and implemented.

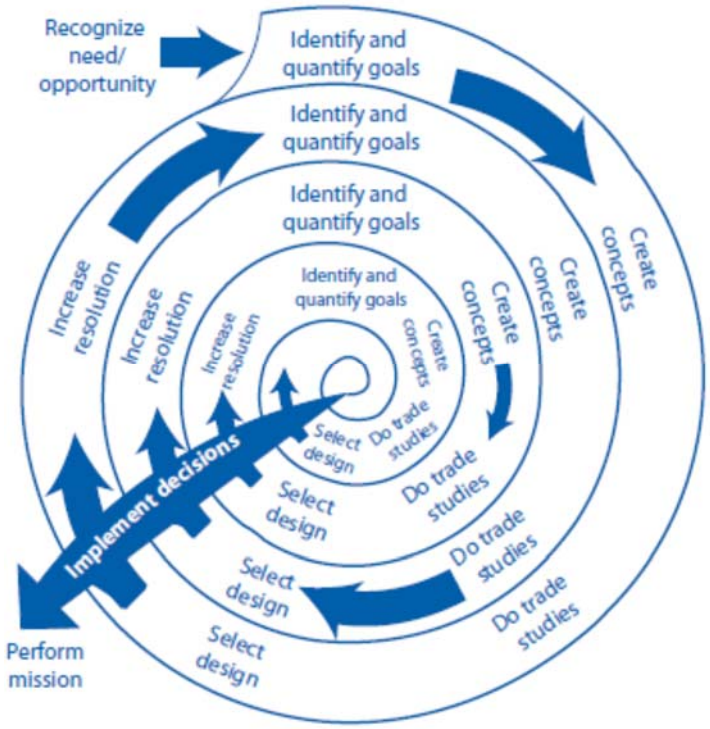


Figure 4.4‑2 The Doctrine of Successive Refinement

Each “create concepts” step in Figure 4.4-2 involves a recursive and iterative design loop driven by the set of stakeholder expectations where a strawman architecture/design, the associated ConOps, and the derived requirements are developed and programmatic constraints such as cost and schedule are considered. These three products should be consistent with each other and will require iterations and design decisions to achieve this consistency. This recursive and iterative design loop is illustrated in Figure 4.0-1.

Each “create concepts” step in Figure 4.4-2 also involves an assessment of potential capabilities offered by the continually changing state of technology and potential pitfalls captured through experience-based review of prior program/project lessons learned data. It is imperative that there be a continual interaction between the technology development process, crosscutting processes such as human systems integration, and the design process to ensure that the design reflects the realities of the available technology and that overreliance on immature technology is avoided. Additionally, the state of any technology that is considered enabling should be properly monitored, and care should be taken when assessing the impact of this technology on the concept performance. This interaction is facilitated through a periodic assessment of the design with respect to the maturity of the technology required to implement the design. (See Section 4.4.2.1 for a more detailed discussion of technology assessment.) These technology elements usually exist at a lower level in the PBS. Although the process of design concept development by the integration of lower level elements is a part of the systems engineering process, there is always a danger that the top-down process cannot keep up with the bottom-up process. Therefore, system architecture issues need to be resolved early so that the system can be modeled with sufficient realism to do reliable trade studies.

As the system is realized, its particulars become clearer—but also harder to change. See the rising "Cost to Change Design Direction" in Figure 2.5-3. The purpose of systems engineering is to make sure that the Design Solution Definition Process happens in a way that leads to the most functional, safe, and cost-effective final system while working within any given schedule boundaries. The basic idea is that before those decisions that are hard to undo are made, the alternatives should be carefully and iteratively assessed, particularly with respect both to the maturity of the required technology and to stakeholder expectations for efficient, effective operations.

4.4.1.2.2 Create Alternative Design Concepts

Once it is understood what the system is to accomplish, it is possible to devise a variety of ways that those goals can be met. Sometimes, that comes about as a consequence of considering alternative functional allocations and integrating available subsystem design options, all of which can have technologies at varying degrees of maturity. Ideally, as wide a range of plausible alternatives as is consistent with the design organization's charter should be defined, keeping in mind the current stage in the process of successive refinement. When the bottom-up process is operating, a problem for the systems engineer is that the designers tend to become fond of the designs they create, so they lose their objectivity; the systems engineer should stay an "outsider" so that there is more objectivity. This is particularly true in the assessment of the technological maturity of the subsystems and components required for implementation. There is a tendency on the part of technology developers and project management to overestimate the maturity and applicability of a technology that is required to implement a design. This is especially true of "heritage" equipment. The result is that critical aspects of systems engineering are often overlooked.

On the first turn of the successive refinement in Figure 4.4-2, the subject is often general approaches or strategies, sometimes architectural concepts. On the next, it is likely to be functional design, then detailed design, and so on. The reason for avoiding a premature focus on a single design is to permit discovery of the truly best design. Part of the systems engineer's job is to ensure that the design concepts to be compared take into account all interface requirements. Characteristic questions include: "Did you include the cabling?" or "Did you consider how the maintainers can repair the system? When possible, each design concept should be described in terms of controllable design parameters so that each represents as wide a class of designs as is reasonable. In doing so, the systems engineer should keep in mind that the potentials for change may include organizational structure, personnel constraints, schedules, procedures, and any of the other things that make up a system. When possible, constraints should also be described by parameters.

4.4.1.2.3 Analyze Each Alternative Design Solution

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

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

The trade study process should be done openly and inclusively. While quantitative techniques and rules are used, subjectivity also plays a significant role. To make the process work effectively, participants should have open minds, and individuals with different skills—systems engineers, design engineers, crosscutting specialty discipline and domain engineers, program analysts, system end users, decision scientists, maintainers, operators, and project managers—should cooperate. The right quantitative methods and selection criteria should be used. Trade study assumptions, models, and results should be documented as part of the project archives. The participants should remain focused on the functional requirements, including those for enabling products. For an in-depth discussion of the trade study process, see Section 6.8. The ability to perform these studies is enhanced by the development of system models that relate the design parameters to those assessments, but it does not depend upon them.

The technical team should consider a broad range of concepts when developing the system model. The model should define the roles of crew, operators, maintainers, logistics, hardware, and software in the system. It should identify the critical technologies required to implement the mission and should consider the entire life cycle from fabrication to disposal. Evaluation criteria for selecting concepts should be established. Cost is always a limiting factor. However, other criteria, such as time to develop and certify a unit, risk, and reliability, also are critical. This stage cannot be accomplished without addressing the roles of operators and maintainers. These contribute significantly to life-cycle costs and to the system reliability. Reliability analysis should be performed based upon estimates of component failure rates for hardware and an understanding of the consequences of these failures. If probabilistic risk assessment models are applied, it may be necessary to include occurrence rates or probabilities for software faults or human error events. These models should include hazard analyses and controls implemented through fault management. Section 7.7 defines fault management approaches to improve system reliability in more detail. Assessments of the maturity of the required technology should be done and a technology development plan developed.

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

Whether system models are used or not, the design concepts are developed, modified, reassessed, and compared against competing alternatives in a closed-loop process that seeks the best choices for further development. System and subsystem sizes are often determined during the trade studies. The end result is the determination of bounds on the relative cost-effectiveness of the

design alternatives, measured in terms of the quantified system goals. (Only bounds, rather than final values, are possible because determination of the final details of the design is intentionally deferred.) Increasing detail associated with the continually improving resolution reduces the spread between upper and lower bounds as the process proceeds.

4.4.1.2.4 Select the Best Design Solution Alternative

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

The Decision Analysis Process, as described in Section 6.8, should be used to make an evaluation of the alternative design concepts and to recommend the "best" design solution.

When it is possible, it is usually well worth the trouble to develop a mathematical expression, called an "objective function," that expresses the values of combinations of possible outcomes as a single measure of cost-effectiveness, as illustrated in Figure 4.4-3, even if both cost and effectiveness should be described by more than one measure.

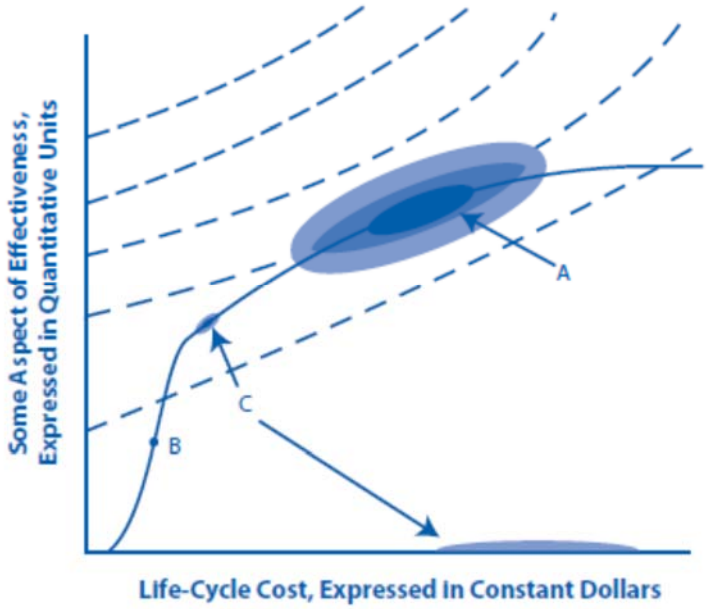


Figure 4.4-3 A Quantitative Objective Function, Dependent on Life-Cycle Cost and All Aspects of Effectiveness

Note: The different shaded areas indicate different levels of uncertainty. Dashed lines represent constant values of objective function (cost-effectiveness). Higher values of cost-effectiveness are achieved by moving toward upper left. A, B, and C are design concepts with different risk patterns.

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

can be compared in terms of their value. Risks associated with design concepts can cause these evaluations to be somewhat nebulous because they are uncertain and are best described by probability distributions.

In Figure 4.4-3, the risks are relatively high for design concept A. There is little risk in either effectiveness or cost for concept B, while the risk of an expensive failure is high for concept C, as is shown by the cloud of probability near the x axis with a high cost and essentially no effectiveness. Schedule factors may affect the effectiveness and cost values and the risk distributions.

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

Technical reviews of the data and analyses, including technology maturity assessments, are an important part of the decision support packages prepared for the technical team. The decisions that are made are generally entered into the configuration management system as changes to (or elaborations of) the system baseline. The supporting trade studies are archived for future use. An essential feature of the systems engineering process is that trade studies are performed before decisions are made. They can then be baselined with much more confidence.

4.4.1.2.5 Increase the Resolution of the Design

The successive refinement process of Figure 4.4-2 illustrates a continuing refinement of the system design. At each level of decomposition, the baselined derived (and allocated) requirements become the set of high-level requirements for the decomposed elements, and the process begins again. One might ask, “When do we stop refining the design?” The answer is that the design effort proceeds to a depth that is sufficient to meet several needs: the design should penetrate sufficiently to allow analytical validation of the design to the requirements and ConOps; it should also have sufficient depth to support cost and operations modeling and to convince a review team of a feasible design with performance, cost, and risk margins.

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

It is reasonable to expect the system to be defined with better resolution as time passes. This tendency is formalized at some point (in Phase B) by defining a baseline system definition.

Usually, the goals, objectives, and constraints are baselined as the requirements portion of the baseline. The entire baseline is then placed under configuration control in an attempt to ensure that any subsequent changes are indeed justified and affordable.

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

4.4.1.2.6 Fully Describe the Design Solution

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

The scope and content of the full design description should be appropriate for the product life-cycle phase, the phase success criteria, and the product position in the PBS (system structure). Depending on these factors, the form of the design solution definition could be simply a simulation model or a paper study report. The technical data package evolves from phase to phase, starting with conceptual sketches or models and ending with complete drawings, parts list, and other details needed for product implementation or product integration. Typical output definitions from the Design Solution Definition Process are shown in Figure 4.4-1 and are described in Section 4.4.1.3.

4.4.1.2.7 Verify the Design Solution

Once an acceptable design solution has been selected from among the various alternative designs and documented in a technical data package, the design solution should next be verified against the system requirements and constraints. A method to achieve this verification is by means of a peer review to evaluate the resulting design solution definition. Guidelines for conducting a peer review are discussed in Section 6.7.2.4.5.

In addition, peer reviews play a significant role as a detailed technical component of higher level technical and programmatic reviews. For example, the peer review of a component battery design can go into much more technical detail on the battery than the integrated power subsystem review. Peer reviews can cover the components of a subsystem down to the level appropriate for verifying the design against the requirements. Concerns raised at the peer review might have implications on the power subsystem design and verification and therefore should be reported at the next higher level review of the power subsystem.

The verification should show that the design solution definition:

- Is realizable within the constraints imposed on the technical effort;
- Has specified requirements that are stated in acceptable statements and have bidirectional traceability with the technical requirements and stakeholder expectations; and

- Has decisions and assumptions made in forming the solution consistent with its set of technical requirements and identified system product and service constraints.

This design solution verification is in contrast to the verification of the end product described in the end product verification plan which is part of the technical data package. That verification occurs in a later life-cycle phase and is a result of the Product Verification Process (see section 5.3) applied to the realization of the design solution as an end product.

4.4.1.2.8 Validate the Design Solution

The validation of the design solution is a recursive and iterative process as shown in Figure 4.0-1. Each alternative design concept is validated against the set of stakeholder expectations. The stakeholder expectations drive the iterative design loop in which a strawman architecture/design, the ConOps, and the derived requirements are developed. These three products should be consistent with each other and will require iterations and design decisions to achieve this consistency. Once consistency is achieved, functional analyses allow the study team to validate the design against the stakeholder expectations. A simplified validation asks the questions: Does the system work as expected? How does the system respond to failures, faults, and anomalies? Is the system affordable? If the answer to any of these questions is no, then changes to the design or stakeholder expectations will be required, and the process is started over again. This process continues until the system—architecture, ConOps, and requirements—meets the stakeholder expectations.

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

4.4.1.2.9 Identify Enabling Products

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

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

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

Special test fixtures or special mechanical handling devices are examples of enabling products that would have to be created by the project. Because of long development times as well as oversubscribed facilities, it is important to identify enabling products and secure the commitments for them as early in the design phase as possible.

4.4.1.2.10 Baseline the Design Solution

As shown earlier in Figure 4.0-1, once the selected system design solution meets the stakeholder expectations, the study team baselines the products and prepares for the next life-cycle phase. Because of the recursive nature of successive refinement, intermediate levels of decomposition are often validated and baselined as part of the process. In the next level of decomposition, the baselined requirements become the set of high-level requirements for the decomposed elements, and the process begins again.

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

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

4.4.1.3 Outputs

Outputs of the Design Solution Definition Process are the specifications and plans that are passed on to the product realization processes. They contain the design-to, build-to, train-to, and code-to documentation that complies with the approved baseline for the system.

As mentioned earlier, the scope and content of the full design description should be appropriate for the product life-cycle phase, the phase success criteria, and the product position in the PBS.

Outputs of the Design Solution Definition Process include the following:

- **The System Specification:** The system specification contains the functional baseline for the system that is the result of the Design Solution Definition Process. The system design specification provides sufficient guidance, constraints, and system requirements for the design engineers to begin developing the design.
- **The System External Interface Specifications:** The system external interface specifications describe the functional baseline for the behavior and characteristics of all physical interfaces

that the system has with the external world. These include all structural, thermal, electrical, and signal interfaces, as well as the human-system interfaces.

- **The End-Product Specifications:** The end-product specifications contain the detailed build-to and code-to requirements for the end product. They are detailed, exact statements of design particulars, such as statements prescribing materials, dimensions, and quality of work to build, install, or manufacture the end product.
- **The End-Product Interface Specifications:** The end-product interface specifications contain the detailed build-to and code-to requirements for the behavior and characteristics of all logical and physical interfaces that the end product has with external elements, including the human-system interfaces.
- **Initial Subsystem Specifications:** The end-product subsystem initial specifications provide detailed information on subsystems if they are required.
- **Enabling Product Requirements:** The requirements for associated supporting enabling products provide details of all enabling products. Enabling products are the life-cycle support products, infrastructures, personnel, logistics, and services that facilitate the progression and use of the operational end product through its life cycle. They are viewed as part of the system since the end product and its enabling products are interdependent.
- **Product Verification Plan:** The end-product verification plan (generated through the Technical Planning Process) provides the content and depth of detail necessary to provide full visibility of all verification activities for the end product. Depending on the scope of the end product, the plan encompasses qualification, acceptance, prelaunch, operational, and disposal verification activities for flight hardware and software.
- **Product Validation Plan:** The end-product validation plan (generated through the Technical Planning Process) provides the content and depth of detail necessary to provide full visibility of all activities to validate the end product against the baselined stakeholder expectations. The plan identifies the type of validation, the validation procedures, and the validation environment that are appropriate to confirm that the realized end product conforms to stakeholder expectations.
- **Logistics and Operate-to Procedures:** The applicable logistics and operate-to procedures for the system describe such things as handling, transportation, maintenance, long-term storage, and operational considerations for the particular design solution.

Other outputs may include:

- **Human Systems Integration Plan**: The system HSI Plan should be updated to indicate the numbers, skills, and development (i.e., training) required for humans throughout the full life-cycle deployment and operations of the system.

4.4.2 Design Solution Definition Guidance

4.4.2.1 Technology Assessment

As mentioned in the process description (section 4.4.1), the creation of alternative design solutions involves assessment of potential capabilities offered by the continually changing state

of technology. A continual interaction between the technology development process and the design process ensures that the design reflects the realities of the available technology. This interaction is facilitated through periodic assessment of the design with respect to the maturity of the technology required to implement the design.

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

If requirements are defined without fully understanding the resources required to accomplish needed technology developments, then the program/project is at risk. Technology assessment should be done iteratively until requirements and available resources are aligned within an acceptable risk posture. Technology development plays a far greater role in the life cycle of a program/project than has been traditionally considered, and it is the role of the systems engineer to develop an understanding of the extent of program/project impacts—maximizing benefits and minimizing adverse effects. Traditionally, from a program/project perspective, technology development has been associated with the development and incorporation of any "new" technology necessary to meet requirements. However, a frequently overlooked area is that associated with the modification of "heritage" systems incorporated into different architectures and operating in different environments from the ones for which they were designed. If the required modifications and/ or operating environments fall outside the realm of experience, then these too should be considered technology development.

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

4.4.2.2 Human Capability Assessment

The requisite human components of complex systems (operators, maintainers, etc.) should—like hardware/software technologies—be assessed for appropriate expectations during systems engineering process execution. If too much is expected or assumed of human elements, they can be prone to fail just as inappropriate technology subsystems can fail to perform, thereby lowering the total hardware/software/human total system performance. For example, very high flight turnaround rates—up to 40 missions per year—were projected for the Space Shuttle System, but these rates were never achievable (9 flights/year maximum) since the system was not designed for quick turnaround times, ground crew factors, and efficient ground maintenance, test, and checkout operations. For some subsystems, the frequency or scope of the turnaround work did not match initial estimates. Inclusion of an HSI Plan in the NASA systems engineering process is

intended to lead program/project managers and systems engineers to be realistic about what functions they allocate to human elements and to assess early what the expectations are for human system elements throughout the program/project life cycle in order to avoid surprises in the operational era for which greater-than-assumed human capital should be engaged as an operational workout to make the system functional. Early consideration of HSI is intended to avoid an under-estimate of life-cycle maintenance costs, including repairs and replacement of parts and the design and build of ground support facilities.

4.4.2.3 Integrating Engineering Specialties into the Systems Engineering Process

As part of the technical effort, specialty engineers, in cooperation with systems engineering and subsystem designers, often perform tasks that are common and cut across disciplines, such as manufacturability, security, safety, operability, and affordability. Some crosscutting disciplines apply specialized analytical techniques to create information needed by the project manager and systems engineer. They also help define and write the concept of operations and system requirements in their areas of expertise, and they review data packages, Engineering Change Requests (ECRs), test results, and documentation for major project reviews. The project manager and/or systems engineer needs to ensure that the information and products so generated add value to the project commensurate with their cost. Specialty engineering technical efforts should be well integrated into the project. The roles and responsibilities of specialty engineering disciplines should be summarized in the SEMP.

The specialty engineering disciplines identified and described in this guide are safety and reliability, fault management, Quality Assurance (QA), Integrated Logistics Support (ILS), maintainability, producibility, and human factors. Integrating these domain experts requires organization, skill, and time, which can be documented as plans in the SEMP. An overview of these specialty engineering disciplines is provided to give systems engineers a brief introduction. This introduction is not intended to be a comprehensive handbook or implementation plan for any of these discipline specialties. Not all of these disciplines may be applicable to all projects.

4.4.2.3.1 Safety and Reliability

Overview and Purpose

A safe and reliable system ensures mission success by functioning properly over its intended life. It has a low and acceptable probability of failure that is achieved through simplicity, proper design for nominal and off-nominal activities, and proper application of reliable parts and materials. In addition to long life, a reliable system is robust and fault tolerant, meaning that it can continue to perform its intended function in the presence of failure as well as variations in its operating parameters and environments. An effective and efficient system integrates well among its hardware, software, and human elements to achieve mission objectives.

Safety and Reliability in the System Design Process

A focus on safety and reliability throughout the mission life cycle is essential for ensuring mission success. The fidelity to which safety and reliability are designed and built into the system depends on the information needed and the type of mission. For human-rated systems, safety and reliability is the primary objective throughout the design process. For science missions, safety and reliability should be commensurate with the funding and level of risk a

program or project is willing to accept. Regardless of the type of mission, safety and reliability considerations should be an integral part of the system design processes.

To realize the maximum benefit from reliability analysis, it is essential to integrate the risk and reliability and fault management analysts within the design teams. The importance of this cannot be overstated. In too many cases, these analysts are only engaged in analyzing a design after the design has been formulated. As a result, safety and reliability features are added on or outsourced rather than designed in. This results in unrealistic analysis that is not focused on risk drivers and does not provide value to the design.

Risk and reliability analyses evolve to answer key questions about design trades as the design matures. Reliability analyses utilize information about the system, identify sources of risk and risk drivers, and provide an important input for decision-making. *NASA-STD-8729.1, Planning, Developing, and Maintaining an Effective Reliability and Maintainability (R&M) Program,* outlines engineering activities that should be tailored for each specific project. The concept is to choose an effective set of reliability and maintainability engineering activities to ensure that the systems designed, built, and deployed will operate successfully for the required mission life cycle.

In the early phases of a project, risk and reliability analyses help designers understand the interrelationships of the concept of operations, requirements, system architectures, human/system function allocations, constraints, and resources, and uncover key relationships and drivers so they can be properly considered. The analyst should help designers go beyond the requirements to understand implicit dependencies that emerge as the design concept matures. It is unrealistic to assume that design requirements will correctly capture all risk and reliability issues and "force" a reliable design. The systems engineer should develop a system strategy mapped to the PBS or function on how to allocate and coordinate reliability, fault tolerance, and recovery between systems both horizontally and vertically within the architecture to meet the total mission requirements. System impacts of designs should play a key role in the design. Making designers aware of the impacts of their decisions on overall mission reliability is key.

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

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

Analysis Techniques and Methods

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

- Event sequence diagrams/event trees are models that describe the sequence of events and responses to off-nominal conditions that can occur during a mission.

- Failure Modes and Effects Analyses (FMEAs) are bottom-up analyses that identify the types of failures that can occur within a system and identify the causes, effects, and mitigating strategies that can be employed to control the effects of the failures.
- Qualitative top-down logic models identify how failures within a system can combine to cause an undesired event.
- Quantitative logic models (probabilistic risk assessment) extend the qualitative models to include the likelihood of failure. These models involve developing failure criteria based on system physics and system success criteria, and employing statistical techniques to estimate the likelihood of failure along with uncertainty.
- Reliability block diagrams are diagrams of the elements to evaluate the reliability of a system to provide a function.
- Preliminary Hazard Analysis (PHA) is performed early based on the functions performed during the mission. Preliminary hazard analysis is a "what if" process that considers the potential hazard, initiating event scenarios, effects, and potential corrective measures and controls. The objective is to determine if the hazard can be eliminated, and if not, how it can be controlled.
- Hazard analysis evaluates the completed design. Hazard analysis is a "what if" process that considers the potential hazard, initiating event, effects, and potential corrective measures and controls. The objective is to determine if the hazard can be eliminated, and if not, how it can be controlled.
- Human reliability analysis is a method to understand how human failures can lead to system failure and estimate the likelihood of those failures.
- Probabilistic structural analysis provides a way to combine uncertainties in materials and loads to evaluate the failure of a structural element.
- Sparing/logistics models provide a means to estimate the interactions of systems in time. These models include ground-processing simulations and mission campaign simulations.

Limitations on Reliability Analysis

The engineering design team should understand that reliability is expressed as the probability of mission success. Probability is a mathematical measure expressing the likelihood of occurrence of a specific event. Therefore, probability estimates should be based on engineering and historical data, and any stated probabilities should include some measure of the uncertainty surrounding that estimate.

Uncertainty expresses the degree of belief analysts have in their estimates. Uncertainty decreases as the quality of data and understanding of the system improve. The initial estimates of failure rates or failure probability might be based on comparison to similar equipment, historical data (heritage), failure rate data from handbooks, or expert elicitation.

In summary:

- Reliability estimates express probability of success.
- Uncertainty should be included with reliability estimates.

- Reliability estimates combined with FMEAs provide additional and valuable information to aid in the decision-making process.

4.4.2.3.2 Fault Management

Fault Management (FM), as described in Section 7.7, comprises the capabilities of the system that preserve the system's ability to function as intended. FM addresses:

- Unexpected and unintended conditions;
- Risk mitigation through monitoring of critical components;
- Detection and location of faults;
- Prediction of future performance degradation; and
- Actions to ensure the safety of the system and the operators during testing and operational phases.

To be effective, FM activities need to begin as early as the conceptual design stage and need to have a system-level perspective, as opposed to local perspective, because a system's design is not complete until potential failures are addressed. Comprehensive FM relies on the cooperative design and operation of separately deployed system elements to achieve overall reliability, availability, and safety objectives. Like all other system elements, FM is constrained by programmatic and operational resources. Thus, FM practitioners are challenged to identify, evaluate, and balance risks to these objectives against the cost and risk of additional FM functionality.

Role of Fault Management Engineer

A Fault Management Systems Engineer (FMSE) is responsible for engineering the set of system functions and elements that maintain desired system behavior in off-nominal situations. The FMSE works closely with systems engineers, safety and mission assurance engineers, subsystem engineers (who are sometimes themselves the fault management engineers for their specific subsystem), and production engineers in assessing potential targets (subsystems / components) for implementing fault management functions. Part of the task is determining the level of fault management that is needed for each of these targets and allocating FM requirements to respective subsystems. To provide the largest possible benefit (e.g., reduced cost and risk, more robust functionality, increased resiliency), it is imperative that the FM requirements, conceptual design, and architecture are developed in unison with the system design. The FMSE performs FM design at the system level, allocates subsystem requirements, and oversees the design and implementation of FM capabilities within all the allocated areas of the system. Design and corresponding requirements for software and hardware must be assessed and negotiated through a risk-effectiveness analysis. As part of that analysis, the FMSE strives to understand the designs and implementations of allocated subsystem FM requirements to do the following:

- Identify and articulate resiliency and recovery properties commensurate with the overall mission posture;
- Search for potential hazardous interactions between subsystem and system designs;

- Understand the ramifications of the subsystem implementations on the system-level FM design; and
- Develop verification and validation plans for the FM capabilities of these systems.

As a part of a project's SE team, the FMSE leverages visibility into the nominal functionality of the entire system to identify and plan appropriate responses to off-nominal behaviors. FM engineering utilizes the analyses of Safety and Mission Assurance (SMA) analysts, which include reliability, availability, maintainability, FMEAs, Probabilistic Risk Assessments (PRAs), and hazards/system safety. FM engineers utilize off-nominal physical analyses from subsystem engineering, as well as creating their own integrated analyses drawing from all of these sources to analyze trades-offs at various levels and across multiple subsystems. The FMSE is most often part of the flight systems engineering team, but depending on project structure, could function at a project systems engineering level and/or a subsystem engineering level. Therefore, a project's organizational structure and delegation of roles/responsibilities/authority has to integrate FM engineering and allow trades to be clearly communicated and resolved across subsystems and engineering disciplines. FM engineers need to maintain a constant awareness of the global nature of engineering decisions that can affect FM as well as FM decisions that can affect overall system complexity and operations. See *NASA-HDBK-1002 Fault Management Handbook* for additional details.

4.4.2.3.3 Quality Assurance

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

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

The Systems Engineer's Relationship to QA

As with reliability, producibility, and other characteristics, quality should be designed as an integral part of any system. It is important that the systems engineer understands SMA's safeguarding role in the broad context of total risk and supports the quality role explicitly and vigorously. All of this is easier if the SMA quality function is actively included and if quality is designed in with buy-in by all roles, starting at concept development. This will help mitigate conflicts between design and quality requirements, which can take on the effect of "tolerance stacking."

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

While there is more leeway in small, less costly robotic projects, rigid adherence to procedural requirements is typically necessary in space flight projects that include high-risk, low-volume manufacturing. In the absence of large samples and long production runs, compliance with these written procedures is a strong step toward ensuring process and, thereby, product consistency. To address this, NASA requires QA programs to be designed to mitigate risks associated with noncompliance with those requirements.

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

4.4.2.3.4 Integrated Logistics Support

The objective of ILS activities within the systems engineering process is to ensure that the product system is supported during development (Phase D) and operations (Phase E) in a cost-effective manner. ILS is particularly important to projects that are reusable or serviceable. Projects whose primary product does not evolve over its operations phase typically only apply ILS to parts of the project (for example, the ground system) or to some of the elements (for example, transportation). As an ILS element, fault management enables improved planning and operations by facilitating in-time repair and maintenance, which not only saves time and money, but also may prevent delays. Fault management and reliability analysis consistent with the sparing philosophy helps determine the spares procurement and approach. ILS is primarily accomplished by early, concurrent consideration of supportability characteristics; performing trade studies on alternative system and ILS concepts; quantifying resource requirements for each ILS element using best practices; and acquiring the support items associated with each ILS element. During operations, ILS activities support the system while seeking improvements in cost-effectiveness by conducting analyses in response to actual operational conditions. These

analyses continually reshape the ILS system and its resource requirements. Neglecting ILS or poor ILS decisions invariably have adverse effects on the life-cycle cost of the resultant system. Table 4.4-1 summarizes the ILS disciplines.

Table 4.4-1 ILS Technical Disciplines

Technical Discipline	Definition
Maintenance support planning	Ongoing and iterative planning, organization, and management activities necessary to ensure that the logistics requirements for any given program are properly coordinated and implemented. Maintenance and logistics planning can be informed by fault management approaches (section 7.7).
Design interface	The interaction and relationship of logistics with the systems engineering process to ensure that supportability influences the definition and design of the system so as to reduce life cycle cost
Technical data and technical publications	The recorded scientific, engineering, technical, and cost information used to define, produce, test, evaluate, modify, deliver, support, and operate the system
Training and training support	Encompasses all personnel, equipment, facilities, data/documentation, and associated resources necessary for the training of operational and maintenance personnel
Supply support	Actions required to provide all the necessary material to ensure the system's supportability and usability objectives are met
Test and support equipment	All tools, condition-monitoring equipment, diagnostic and checkout equipment, special test equipment, metrology and calibration equipment, maintenance fixtures and stands, and special handling equipment required to support operational maintenance functions
Packaging, handling, storage, and transportation	All materials, equipment, special provisions, containers (reusable and disposable), and supplies necessary to support the packaging, safety and preservation, storage, handling, and transportation of the prime mission-related elements of the system, including personnel, spare and repair parts, test and support equipment, technical data computer resources, and mobile facilities
Personnel	Involves identification and acquisition of personnel with skills and grades required to operate and maintain a system over its lifetime
Logistics facilities	All special facilities that are unique and are required to support logistics activities, including storage buildings and warehouses and maintenance facilities at all levels
Computer resources support	All computers, associated software, connecting components, networks, and interfaces necessary to support the day-to-day flow of information for all logistics functions

Source: Blanchard, System Engineering Management.

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

4.4.2.3.5 Maintainability

Maintainability is defined as the measure of the ability of an item to be retained in or restored to specified conditions when maintenance is performed by personnel having specified skill levels, using prescribed procedures and resources, at each prescribed level of maintenance. It is the

inherent characteristics of a design or installation that contribute to the ease, economy, safety, and accuracy with which maintenance actions can be performed.

Role of the Maintainability Engineer

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

4.4.2.3.6 Producibility

Producibility is a system characteristic associated with the ease and economy with which a completed design can be transformed (i.e., fabricated, manufactured, or coded) into a hardware and/or software realization. While major NASA systems tend to be produced in small quantities, a particular producibility feature can be critical to a system's cost-effectiveness, as experience with the shuttle's thermal tiles has shown. Factors that influence the producibility of a design include the choice of materials, simplicity of design, flexibility in production alternatives, tight tolerance requirements, and clarity and simplicity of the technical data package.

Role of the Production Engineer

The production engineer supports the systems engineering process (as a part of the multidisciplinary product development team) by taking an active role in implementing specific design features to enhance producibility and by performing the production engineering analyses needed by the project. These tasks and analyses include:

- Performing the manufacturing/fabrication portion of the system risk management program. This is accomplished by conducting a rigorous production risk assessment and by planning effective risk mitigation actions.
- Identifying system design features that enhance producibility. Efforts usually focus on design simplification, fabrication tolerances, and avoidance of hazardous materials.
- Conducting producibility trade studies to determine the most cost-effective fabrication/manufacturing process.
- Assessing production feasibility within project constraints. This may include assessing contractor and principal subcontractor production experience and capability, new fabrication technology, special tooling, and production personnel training requirements.
- Identifying long-lead items and critical materials.
- Estimating production costs as a part of life-cycle cost management.
- Supporting technology readiness assessments.
- Developing production schedules.

- Developing approaches and plans to validate fabrication/manufacturing processes.

The results of these tasks and production engineering analyses are documented in the manufacturing plan with a level of detail appropriate to the phase of the project. The production engineer also participates in and contributes to major project reviews (primarily PDR and Critical Design Review (CDR)) on the above items, and to special interim reviews such as the PRR.

Prototypes

The prototype unit demonstrates form, fit, and function at a scale deemed to be representative of the final product operating in its operational environment.

Experience has shown that prototype systems can be effective in enabling efficient producibility and maintainability even when building only a single flight system. Prototypes are built early in the life cycle and they are made as close to the flight item in form, fit, and function as is feasible at that stage of the development.

The prototype is used to "wring out" the design solution so that experience gained from the prototype can be fed back into design changes that will improve the manufacture, integration, and maintainability of a single flight item or the production run of several flight items.

Prototypes are often challenged by projects to save cost. This often leads the project to accept an increased risk in the development phase of the life cycle. It is important for the systems engineer to understand the utility of the prototype and the mitigated risks in order to justify program cost and schedule.

Fortunately, advancements in computer-aided design and manufacturing have mitigated that risk somewhat by enabling the designer to visualize the design and "walk through" integration and maintenance sequences to uncover problems before they become a costly reality. This includes human interaction assessments for assembly and maintenance actions.

4.4.2.3.7 Human Factors Engineering

Overview and Purpose

Human Factors Engineering (HFE) is the discipline that studies human-system interfaces and interactions and provides requirements, standards, and guidelines to ensure the entire system can function as designed with effective accommodation of the human component.

HFE focuses on those aspects where people interface with the system. It considers all personnel who should interact with the system, not just the operator; deals with organizational systems as well as hardware and software; and examines all types of interactions. The role of the HFE specialist is to advocate for the human component and to ensure that the design of hardware, software, tasks, and environment is compatible with the sensory, perceptual, cognitive, and physical attributes of humans interacting with the system.

Role of the Human Factors Engineer

It is necessary to include HFE on the team throughout all the systems engineering common technical processes so that they can create and execute the specific HFE analysis techniques and tests customized to the specific process or project. Not only do the HFE specialists help in the

development of the end items, but they also ensure that the verification test and completeness techniques are appropriate for humans to undertake. Participation early in the process is especially important. Entering the system design process early ensures that human systems requirements are "designed in" rather than corrected later.

Human Factors engineers facilitate human-centered design (HCD) processes as a part of HSI's implementation. For human space flight, HCD is mandated by *NASA-STD-3001, NASA Space Flight Human System Standard* and includes ConOps and scenario development, task analyses, function allocation between humans and systems, allocation of roles and responsibilities among humans, iterative conceptual design and prototyping, empirical testing such as human-in-the-loop and model-based assessment of human-system performance, and in situ monitoring of human-system performance during operations.

It is very likely that of all the HSI domains mentioned in Section 7.9.1 of this document, that HFE will provide the largest contribution to the SEMP and HSI Plan. HFE processes are traditionally designed to coordinate and integrate well with systems engineering, iterative conceptual development and evaluation, verification, validation, and operational assessments. The cost-benefits of HFE—like all of HSI—are greatest when HFE/HSI contributes to system design from the earliest workings of Pre-Phase A development.

4.4.2.3.8 Human Factors Engineering in the System Design Process

Humans are initially "integrated" into systems through analysis of the overall mission. Mission functions are allocated to humans early in Stakeholder Expectations Definition Process (section 4.1) and, as appropriate, to the system architecture, technical capabilities, cost factors, and crew capabilities. Once functions are allocated, human factors analysts work with system designers to ensure that human operators (ground support and crew), trainers, and maintainers are provided with the equipment, tools, and interfaces needed to perform their assigned tasks safely and effectively.

Figure 4.4-4 provides a reference of *human factors process phases* to be aware of in planning, analyzing, designing, testing, operating, and maintaining systems.

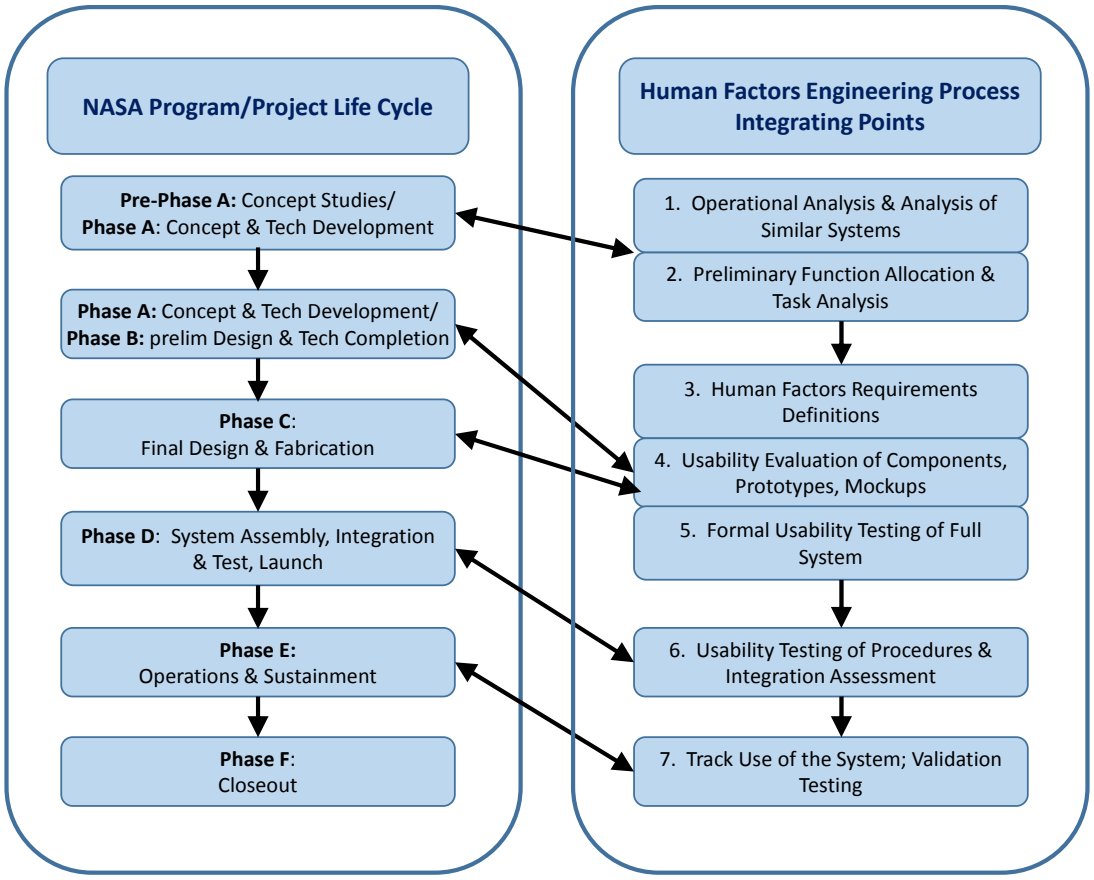


Figure 4.4-4 HF Engineering Process and Its Links to the NASA Program/Project Life Cycle

HFE methods are used to understand user needs, prototype design alternatives, analyze systems, provide data about human performance, make predictions about human-system performance, and evaluate whether the human-machine system performance meets design criteria. These methods are appropriate to all phases of system design with increasing specificity and detail as development progresses. HFE principles must be tailored to fit the design phase. The methods include:

- **Task Analysis:** Produces a detailed description of the things a person should do in a system to accomplish a task, with emphasis on requirements for information presentation, decisions to be made, task times, operator actions, and environmental conditions.
- **Timeline Analysis:** Durations of tasks are identified in task analyses, and the times at which these tasks occur are plotted in graphs, which also show the task sequences. The purpose is to identify requirements for simultaneous incompatible activities and activities that take longer than is available. Timelines for a given task can describe the activities of multiple operators or crewmembers.
- **Modeling and Simulation:** Models or mockups are used to make predictions about system performance, compare configurations, evaluate procedures, and evaluate alternatives. Simulations can be as simple as positioning a graphical human model with realistic

anthropometric dimensions with a graphical model of an operator station, or they can be complex stochastic models capturing aspects such as decision points and error opportunities.

- **Usability Testing:** Based on a task analysis and preliminary design, realistic tasks are carried out in a controlled environment with monitoring and recording equipment. Objective measures such as performance time and number of errors are evaluated; subjective ratings are collected as well (e.g., questionnaires, rating scales). The outputs systematically describe strengths and weaknesses of candidate design solutions.
- **Workload Assessment:** Measurement on a standardized scale such as the NASA Task Load Index (NASA-TLX) or the Cooper-Harper rating scales of the amount and type of workload. It assesses operator and crew task loading, which determines the ability of a human to perform the required tasks in the desired time with the desired accuracy.
- **Human Error and Human Reliability Assessment:** Includes both top-down (fault tree analyses) and bottom-up (human factors process failure modes and effects analysis) analyses. The goal is to promote human reliability by creating a system that can tolerate and recover from human errors. Such a system should also support the human role in adding reliability to the system.

5.0 Product Realization

This chapter describes the activities in the product realization processes listed in Figure 2.1-1. The chapter is separated into sections corresponding to steps 5 through 9 listed in Figure 2.1-1. The processes within each step are discussed in terms of the inputs, the activities, and the outputs. Additional guidance is provided using examples that are relevant to NASA projects.

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

This effort starts with the technical team taking the output from the system design processes and using the appropriate crosscutting functions, such as data and configuration management, and technical assessments to make, buy, or reuse subsystems. Once these subsystems are realized, they should be integrated to the appropriate level as designated by the appropriate interface requirements. These products are then verified through the Technical Assessment Process to ensure that they are consistent with the technical data package and that "the product was built right." Once consistency is achieved, the technical team validates the products against the stakeholder expectations to ensure that "the right product was built." Upon successful completion of validation, the products are transitioned to the next level of the system. Figure 5.0-1 illustrates these processes.

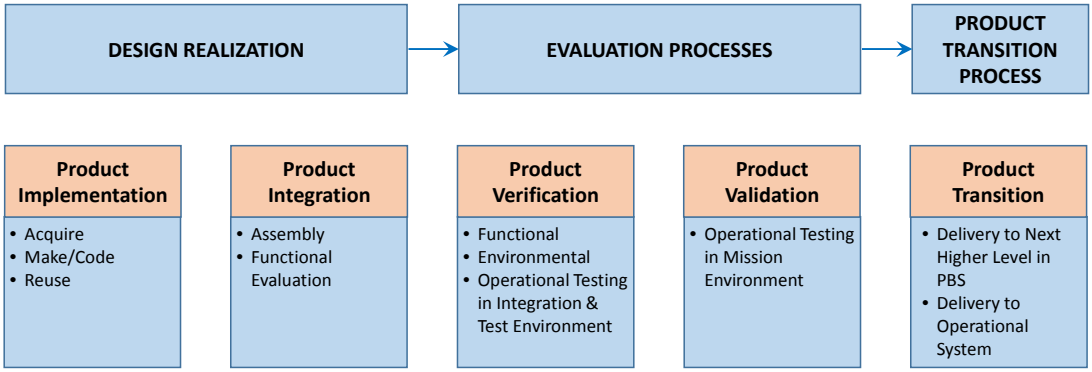


Figure 5.0-1 Product Realization

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

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

Product Realization Keys

- Define and execute production activities.
- Generate and manage requirements for off-the-shelf hardware/software products as for all other products.
- Understand the differences between verification testing and validation testing.
- Consider all customer, stakeholder, technical, programmatic, and safety requirements when evaluating the input necessary to achieve a successful product transition.
- Analyze for any potential incompatibilities with interfaces as early as possible.
- Completely understand and analyze all test data for trends and anomalies.
- Understand the limitations of the testing and any assumptions that are made.
- Ensure that a reused product meets the verification and validation required for the relevant system in which it is to be used, as opposed to relying on the original verification and validation it met for the system of its original use. Then ensure that it meets the same verification and validation as a purchased product or a built product. The “pedigree” of a reused product in its original application should not be relied upon in a different system, subsystem, or application.

5.1 Product Implementation

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

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

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

5.1.1 Process Description

Figure 5.1-1 provides a typical flow diagram for the Product Implementation Process and identifies typical inputs, outputs, and activities to consider in addressing product implementation.

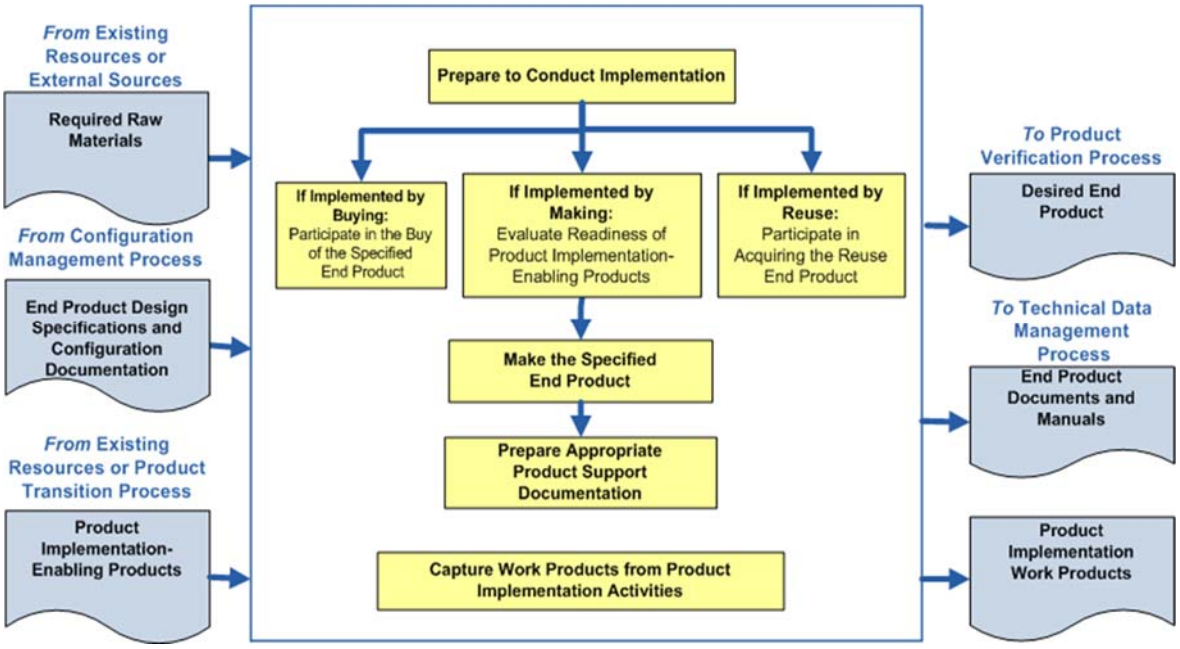


Figure 5.1-1 Product Implementation Process

5.1.1.1 Inputs

Inputs to the Product Implementation Process depend primarily on the decision about whether the end product will be purchased, developed from scratch, or formed by reusing part or all of products from other projects. Typical inputs are shown in Figure 5.1-1.

- **Inputs If Purchasing the End Product:** If the decision was made to purchase part or all of the products for this project, the end product design specifications are obtained from the configuration management system as well as other applicable documents.
- **Inputs If Making/Coding the End Product:** For end products that will be made/coded by the technical team, the inputs will be the configuration-controlled design specifications, manufacturing plans, manufacturing processes, manufacturing procedures, and raw materials as provided to or purchased by the project.
- **Inputs Needed If Reusing an End Product:** For end products that will reuse part or all of products generated by other projects, the inputs may be the documentation associated with the product as well as the product itself. Care should be taken to ensure that these products will indeed meet the specifications and environments for this project. These would have been factors involved in the Decision Analysis Process to determine the make/buy/reuse decision.
- **Enabling Products:** These would be any enabling products necessary to make, code, purchase, or reuse the product (e.g., drilling fixtures, production facilities, production lines, software development facilities, software test facilities, system integration and test facilities).

5.1.1.2 Process Activities

Implementing the product can take one of three forms:

1. Purchase/buy
2. Make/code
3. Reuse

These three forms will be discussed in the following subsections. Figure 5.1-1 shows what kind of inputs, outputs, and activities are performed during product implementation regardless of where in the product hierarchy or life cycle it is. These activities include preparing to conduct the implementation, purchasing/making/reusing the product, and capturing the product implementation work product. In some cases, implementing a product may have aspects of more than one of these forms (such as a build-to-print). In those cases, the appropriate aspects of the applicable forms are used.

5.1.1.2.1 Prepare to Conduct Implementation

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

The documentation, specifications, and other inputs also need to be reviewed to ensure they are ready and at an appropriate level of detail to adequately complete the type of implementation form being employed and for the product life-cycle phase. For example, if the “make”

implementation form is being employed, the design specifications need to be reviewed to ensure they are at a design-to level that allows the product to be developed. If the product is to be bought as a pure Commercial Off-the-Shelf (COTS) item, the specifications need to be checked to make sure they adequately describe the vendor characteristics to narrow to a single make/model of their product line.

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

5.1.1.2.2 Purchase, Make, or Reuse the Product

Purchase the Product

In the first case, the end product is to be purchased from a commercial or other vendor. (If the end product to be purchased is a major contracted effort, see Section 7.1). Design/purchase specifications will have been generated during requirements development and provided as inputs. The technical team needs to review these specifications and ensure they are in a form adequate for the contract or purchase order. This may include the generation of contracts, Statements of Work (SOWs), requests for proposals, purchase orders, or other purchasing mechanisms. For major end products purchased from a vendor, the responsibilities of the Government and contractor team should be documented in the SEMP and Integration Plan. This will define, for example, whether NASA expects the vendor to provide a fully verified and validated product or whether the NASA technical team will be performing those duties. The team needs to work with the acquisition team to ensure the accuracy of the contract SOW or purchase order and to ensure that adequate documentation, certificates of compliance, or other specific needs are requested from the vendor. See Section 7.1 for more details.

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

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

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

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

been captured, the product may be ready to enter the Product Transition Process to be delivered to the next higher level or to its final end user. For products that the technical team will verify and validate, the product will be ready for verification after ensuring that all work products for this phase have been captured.

Make/Code the Product

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

The product is then made or coded in accordance with the specified requirements, configuration documentation, and applicable standards. Software development must be consistent with NPR 7150.2, NASA Software Engineering Requirements. Throughout this process, the technical team should work with the quality organization to review, inspect, and discuss progress and status within the team and with higher levels of management as appropriate. Progress should be documented within the technical schedules. Peer reviews, audits, unit testing, code inspections, simulation checkout, and other techniques may be used to ensure the made or coded product is ready for the verification process. Some production and coding can also be separately contracted. This is sometimes pursued as a cost control feature providing motivation for the design contractor to keep the operations costs low and not roll costs into the operations phase of a long-term contract. This is also valuable when the design contractor is not well suited for long-term continuing production operations. Small projects and activities often use small manufacturing shops to fabricate the system or major portions and small software companies to code their software. In these cases, the production and software engineers may specify some portion of the hardware production or software coding and request the remaining portions, including as-built documentation, from the manufacturing or software provider. The specified portions are contained as part of the contract statement of work in these cases. The level of process control and information provided to or from the vendor is dependent on the criticality of the systems obtained. As production proceeds and components are produced, there is a need to establish a method (Material Review Boards (MRBs) are typically used for large projects) to review any nonconformance to specifications and disposition whether the components can be accepted, reworked, or scrapped and remade.

Reuse

If the strategy is to reuse a product that already exists, extreme care should be taken to ensure that the product is truly applicable to this project and for the intended uses and the environment in which it will be used. This should have been a major factor used in the decision strategy to make / buy / reuse. If the new environment is more extreme, requalification is needed for the component or system. Design factors of safety, margins, and other required design and construction standards should also be assessed. If the program/project requires higher factor of safety or margins, the component may not be useable or a waiver may have to be approved.

The documentation available (e.g., as-built documentation, user's guides, operations manuals, discrepancy reports, waivers and deviations) from the reuse product should be reviewed by the technical team so that they can become completely familiar with the product and ensure it will meet the requirements in the intended environment. Any supporting manuals, drawings, or other documentation available should also be gathered.

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

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

A reused product often needs to undergo the same verification and validation as a purchased product or a built product. Relying on prior verification and validation should only be considered if the product's verification and validation documentation meets or exceeds the verification, validation, and documentation requirements of the current project and the documentation demonstrates that the product was verified and validated against equivalent requirements (including environments) and expectations. The savings gained from reuse is not necessarily from reduced acceptance-level testing of the flight products, but possibly elimination of the need to fully requalify the item (if all elements are the same, including the environment and operation), elimination of the need to specify all of the internal requirements such as printed circuit board specifications or material requirements, reduced internal data products, or the confidence that the item will pass acceptance test and will not require rework.

5.1.1.2.3 Capture Work Products

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

5.1.1.3 Outputs

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

information such as as-built specifications or stakeholder expectations); certificates of compliance; or other vendor documentation.

- **Product Implementation Work Products:** Any additional work products providing reports, records, lesson learned, assumptions, updated CM products, and other outcomes of these activities.

The process is complete when the following activities have been accomplished:

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

5.1.2 Product Implementation Guidance

5.1.2.1 Buying Off-the-Shelf Products

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

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

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

- Maintenance support and relevance of maintenance actions for other customers of the same product line;
- Heritage of the product;
- Critical or noncritical application;
- Amount of modification required and who performs it;
- Whether sufficient documentation is available;
- Proprietary, usage, ownership, warranty, and licensing rights;
- Future support for the product from the vendor/provider;
- Any additional validation of the product needed by the project; and

- Agreement on disclosure of defects discovered by the community of users of the product.

5.1.2.2 Heritage

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

The focus of a "heritage review" is to confirm the applicability of the component for the current application. Assessments should be made regarding not only technical interfaces (hardware and software) and performance, but also the environments to which the unit has been previously qualified, including space environment, aeronautic environment, electromagnetic compatibility, radiation, and contamination. The compatibility of the design with parts quality requirements should also be assessed. All instances of noncompliance should be identified, documented, and addressed either by modification to bring the component into compliance or formal waivers / deviations for accepted deficiencies. This heritage review is commonly held soon after contract award.

When reviewing a product's applicability, it is important to consider the nature of the application. A "catastrophic" application is one where a failure could cause loss of life or vehicle. A "critical" application is one where failure could cause loss of mission. For use in these applications, several additional precautions should be taken including ensuring that the product will not be used near the boundaries of its performance or environmental envelopes. Extra scrutiny by experts should be applied during Preliminary Design Reviews (PDRs) and Critical Design Reviews (CDRs) to ensure the appropriateness of its use. Tabletop peer reviews are often extremely valuable at this stage.

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

For additional information and suggested test and analysis requirements for OTS products, see the NEN and the V&V Community of Practice and G-1182006e *AIAA Guide for Managing the Use of Commercial Off the Shelf (COTS) Software Components for Mission-Critical Systems*.

5.2 Product Integration

Product integration is a key activity of the systems engineer. Product integration is the engineering of the subsystem interactions and their interactions with the system environments (both natural and induced). Also in this process, lower-level products are assembled into higher-level products and checked to make sure that the integrated product functions properly and that there are no adverse emergent behaviors. This integration begins during concept definition and continues throughout the system life cycle. Integration involves several activities focused on the interactions of the subsystems and environments. These include system analysis to define and understand the interactions, development testing including qualification testing, and integration with external systems (e.g., launch operations centers, space vehicles, mission operations centers, flight control centers, and aircraft) and objects (i.e., planetary bodies or structures). To accomplish this integration, the systems engineer is active in integrating the different discipline and design teams to ensure system and environmental interactions are being properly balanced by the differing design teams. The result of a well-integrated and balanced system is an elegant design and operation.

Integration begins with concept development, ensuring that the system concept has all necessary functions and major elements and that the induced and natural environment domains in which the system is expected to operate are all identified. Integration continues during requirements development, ensuring that all system and environmental requirements are compatible and that the system has a proper balance of functional utility to produce a robust and efficient system. Interfaces are defined in this phase and are the pathway of system interactions. Interfaces include mechanical (i.e., structure, loads), fluids, thermal, electrical, data, logical (i.e., algorithms and software), and human. These interfaces may include support for assembly, maintenance, and testing functions in addition to the system main performance functions. The interactions that occur through all of these interfaces can be subtle and complex, leading to both intended and unintended consequences. All of these interactions need to be engineered to produce an elegant and balanced system.

Integration during the design phase continues the engineering of these interactions and requires constant analysis and management of the subsystem functions and the subsystem interactions between themselves and with their environments. Analysis of the system interactions and managing the balance of the system is the central function of the systems engineer during the design process. The system needs to create and maintain a balance between the subsystems, optimizing the system performance over any one subsystem to achieve an elegant and efficient design. The design phase often involves development testing at the component, assembly, or system level. This is a key source of data on system interactions, and the developmental test program should be structured to include subsystem interactions, human-in-the-loop evaluations, and environmental interaction test data as appropriate.

Integration continues during the operations phase, bringing together the system hardware, software, and human operators to perform the mission. The interactions between these three integrated natures of the system need to be managed throughout development and into operations for mission success. The systems engineer, program manager, and the operations team (including the flight crew from crewed missions) need to work together to perform this management. The systems engineer is not only cognizant of these operations team interactions, but is also involved

in the design responses and updates to changes in mission parameters and unintended consequences (through fault management).

Finally, integration or deintegration occurs during system closeout (i.e., decommissioning and disposal). The system capabilities to support de-integration and/or disposal need to be engineered into the system from the concept definition phase. The closeout phase involves the safe disposal of flight assets consistent with U.S. policy and law and international treaties. This disposal can involve the safe reentry and recovery or impact in the ocean, impact on the moon, or solar trajectory. This can also involve the disassembly or repurposing of terrestrial equipment used in manufacturing, assembly, launch, and flight operations. Dispositioning of recovered flight assets also occurs during this phase. Capture of system data and archiving for use in future analysis also occurs. In all of these activities, the systems engineer is involved in ensuring a smooth and logical disassembly of the system and associated program assets.

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

Product integration involves many activities that need to be planned early in the program or project in order to effectively and timely accomplish the integration. Some integration activities (such as system tests) can require many years of work and costs that need to be identified and approved through the budget cycles. An integration plan should be developed and documented to capture this planning. Small projects and activities may be able to include this as part of their SEMP. Some activities may have their integration plans captured under the integration plan of the sponsoring flight program or R&T program. Larger programs and projects need to have a separate integration plan to clearly lay out the complex analysis and tests that need to occur. An example outline for a separate integration plan is provided in appendix H.

During project closeout, a separate closeout plan should be produced describing the decommissioning and disposal of program assets. (For example, see National Space Transportation System (NSTS) 60576, *Space Shuttle Program, Transition Management Plan*). For smaller projects and activities, particularly with short life cycles (i.e., short mission durations), the closeout plans may be contained in the SEMP.

5.2.1 Process Description

Figure 5.2-1 provides a typical flow diagram for the Product Integration Process and identifies typical inputs, outputs, and activities to consider in addressing product integration. The activities of the Product Integration Process are truncated to indicate the action and object of the action.

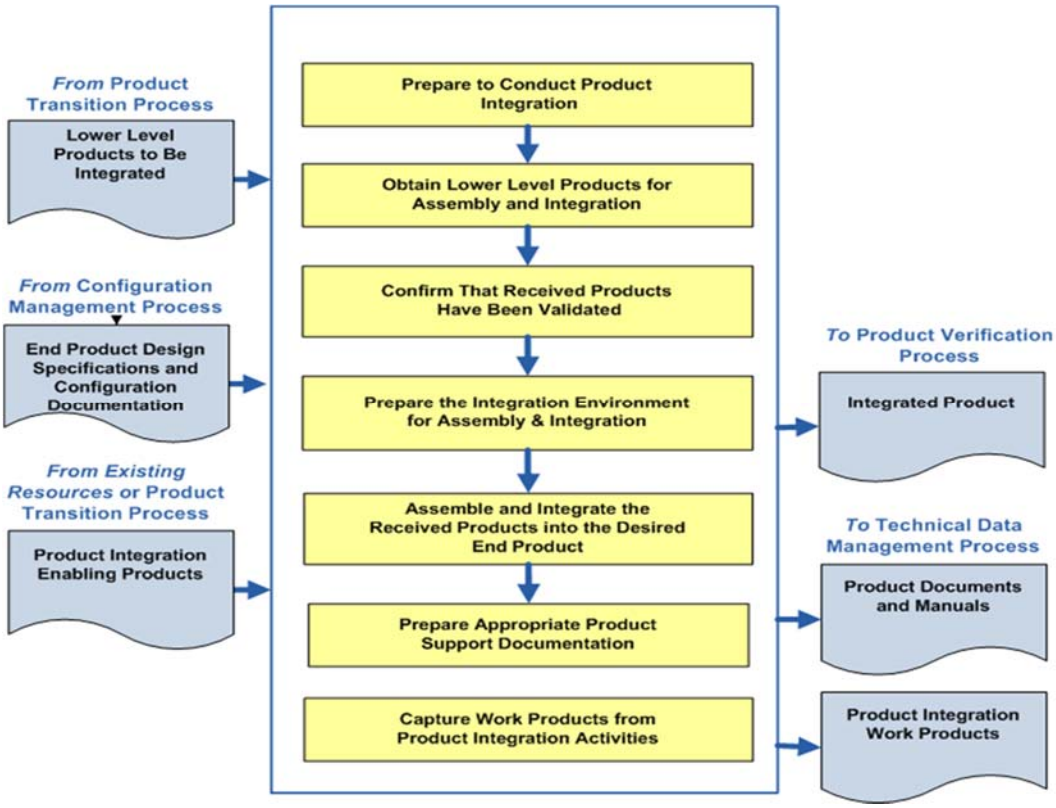


Figure 5.2-1 Product Integration Process

5.2.1.1 Inputs

- **Lower-level products to be integrated:** These are the products developed in the previous lower-level tier in the product hierarchy. These products will be integrated / assembled to generate the product for this product layer.
- **End product design specifications and configuration documentation:** These are the specifications, Interface Control Documents (ICDs), drawings, integration plan, procedures or other documentation or models needed to perform the integration including documentation for each of the lower-level products to be integrated.
- **Product integration-enabling products:** These would include any enabling products, such as holding fixtures, necessary to successfully integrate the lower-level products to create the end product for this product layer.

5.2.1.2 Process Activities

This subsection addresses the approach to the implementation of the Product Integration Process, including the activities required to support the process. The basic tasks that need to be established involve the management of internal and external interactions of the various levels of products and operator tasks to support product integration and are as follows:

5.2.1.2.1 Prepare to Conduct Product Integration

Prepare to conduct product integration by (1) reviewing the product integration strategy/plan (see Section 6.1.2.4.4), generating detailed planning for the integration, and developing integration sequences and procedures; and (2) determining whether the product configuration documentation is adequate to conduct the type of product integration applicable for the product life-cycle phase, location of the product in the system structure, and management phase success criteria.

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

The optimal sequence of assembly is built from the bottom up as components become sub-elements, elements, and subsystems, each of which should be checked prior to fitting it into the next higher assembly. The sequence will encompass any effort needed to establish and equip the assembly facilities; e.g., raised floor, hoists, jigs, test equipment, input/output, and power connections. Once established, the sequence should be periodically reviewed to ensure that variations in production and delivery schedules have not had an adverse impact on the sequence or compromised the factors on which earlier decisions were made.

5.2.1.2.2 Obtain Lower-Level Products for Assembly and Integration

Each of the lower-level products that is needed for assembly and integration is obtained from the transitioning lower-level product owners or a storage facility as appropriate. Received products should be inspected to ensure no damages occurred during the transitioning process.

5.2.1.2.3 Confirm That Received Products Have Been Validated

Confirm that the received products that are to be assembled and integrated have been validated to demonstrate that the individual products satisfy the agreed-to set of stakeholder expectations, including interface requirements. This validation can be conducted by the receiving organization or by the providing organization if fully documented or witnessed by the receiving representative.

5.2.1.2.4 Prepare the Integration Environment for Assembly and Integration

Prepare the integration environment in which assembly and integration will take place, including evaluating the readiness of the product integration-enabling products and the assigned workforce. These enabling products may include facilities, equipment jigs, tooling, and assembly/production lines. The integration environment includes test equipment, simulators, models, storage areas, and recording devices.

5.2.1.2.5 Assemble and Integrate the Received Products into the Desired End Product

Assemble and integrate the received products into the desired end product in accordance with the specified requirements, configuration documentation, interface requirements, applicable standards, and integration sequencing and procedures. This activity includes managing, evaluating, and controlling physical, functional, and data interfaces among the products being integrated.

Functional testing of the assembled or integrated unit is conducted to ensure that assembly is ready to enter verification testing and ready to be integrated into the next level. Typically, all or key representative functions are checked to ensure that the assembled system is functioning as expected. Formal product verification and validation will be performed in the next process.

5.2.1.2.6 Prepare Appropriate Product Support Documentation

Prepare appropriate product support documentation, such as special procedures for performing product verification and product validation. Drawings or accurate models of the assembled system are developed and confirmed to be representative of the assembled system.

5.2.1.2.7 Capture Product Integration Work Products

Capture work products and related information generated while performing the Product Integration Process activities. These work products include system models, system analysis data and assessment reports, derived requirements, the procedures that were used in the assembly, decisions made and supporting rationale, assumptions that were made, identified anomalies and associated corrective actions, lessons learned in performing the assembly, and updated product configuration and support documentation.

5.2.1.3 Outputs

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

- **Integrated product(s)** with all system interactions identified and properly balanced.
- **Documentation and manuals** including system analysis models, data, and reports supporting flight-readiness rationale and available for future analysis during the operation of the system in the mission-execution phase.
- **Work products**, including reports, records, and non-deliverable outcomes of product integration activities (to support the Technical Data Management Process); integration strategy document; assembly/check area drawings; system/component documentation sequences and rationale for selected assemblies; interface management documentation; personnel requirements; special handling requirements; system documentation; shipping schedules; test equipment and drivers' requirements; emulator requirements; and identification of limitations for both hardware and software.

5.2.2 Product Integration Guidance

5.2.2.1 Product Integration Strategy

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

The optimal sequence of assembly is built from the bottom up as components become subelements, elements, and subsystems, each of which should be checked prior to fitting it into the next higher assembly. The sequence will encompass any effort needed to establish and equip the assembly facilities; e.g., raised floor, hoists, jigs, test equipment, input/output, and power connections. Once established, the sequence should be periodically reviewed to ensure that variations in production and delivery schedules have not had an adverse impact on the sequence or compromised the factors on which earlier decisions were made.

See Section 7.1 for a discussion of integration issues that arise when different components of a complex program are developed, acquired, and/or integrated under differing contract mechanisms.

5.2.2.2 Relationship to Product Implementation

As previously described, product implementation is where the plans, designs, analysis, requirements development, and drawings are realized into actual products.

Product integration focuses on the planning and analysis necessary to produce a complete system design for implementation. Product integration evaluates the system interactions within itself and with the environment by identifying system interfaces, establishing the system environments, identifying organizational relationship interactions, defining the key system analysis to be conducted, the test strategy, and the assembly and integration plans. Product integration also provides the closeout plan identifying key activities and system features (derived requirements) necessary to enable decommissioning and/or disposal of the system.

System analysis of the various system configurations and design options is performed to select design options as discussed in Section 4.4, Design Solution Definition. System analysis focuses on the uncertainty and sensitivities of the integrated design configuration to ensure the system will perform as intended. As design decisions and configuration down-selections are made, derived technical requirements are produced. Testing is defined to assess the design options and to anchor the system analysis models to reduce the uncertainties and determine product sensitivities to various effects including the natural and induced environments. The system analysis and planning during product integration also consider assembly of components, subassemblies, assemblies, subsystems, and systems into a final integrated product. This includes human system integration activities to ensure manufacturing, operations, and maintenance activities can be performed by human technicians and operators in a coherent, safe, and efficient

manner. Organizational interactions are also considered, including the relationships with those organizations responsible for product implementation (e.g., manufacturing).

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

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

5.2.2.3 Product Integration Support

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

The following is a list of typical products that support the integration of products and interfaces and that should be addressed by the project in the overall approach to product integration: requirements documents; requirements reviews; design reviews; design drawings and specifications; integration and test plans; hardware configuration control documentation; quality assurance records; interface control requirements/documents; ConOps documents; verification requirement documents; verification reports/analysis; NASA, military, and industry standards; best practices; and lessons learned.

5.2.2.4 Product Integration of the Design Solution

This subsection addresses the more specific implementation of product integration related to the selected design solution.

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

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

5.2.2.5 System Analysis

There are many different system analyses that will need to be conducted in order to ensure the system interactions are fully identified and managed. The specific analysis conducted will depend on the specific system being developed and operated. Typical analyses that show an integrated view of the system include: loads, controllability/stability, thermal, power quality, data bandwidth, flight measurements, mass margin, system energy, etc. In addition, manufacturability, maintainability, and testability analysis should be conducted as part of the requirements development cycle to identify features that need to be included in the system design.

5.2.2.5.1 Compatibility Analysis

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

5.2.2.6 Interface System Integration

Integration of the elements of the system should be performed in accordance with the established integration plan. This ensures that the integration of the system elements into larger or more complex assemblies is conducted in accordance with the planned strategy. Software integration typically occurs in a software integration facility. Once the software has been integrated into a complete load, integration and testing occurs with the flight avionics hardware. This is typically the first form of system integration and provides testing of the software control and interaction with the avionics hardware, including the system control and response algorithms. Once the integration is complete, the flight software is loaded onto the flight system for final checkout prior to launch and/or operation. Hardware integration occurs in the manufacturing plants, launch

centers, or laboratories. This integration occurs in phases beginning with hardware system tests and scaling to the full system assembly. Smaller projects and activities will produce an Engineering Development Unit (EDU) to test the full system (hardware, software, human) integration and interactions. Larger programs will typically do this at the element level and do final system integration at the launch center, flight facility, or onorbit. Simulations are often built to support human operator training, test software interactions (as part of the software integration facility) where full-scale EDUs are not practical or cost-effective.

5.3 Product Verification

The Product Verification Process is the first of the verification and validation processes conducted on an end product. As used in the context of the systems engineering common technical processes, a product is one provided by either the Product Implementation Process or the Product Integration Process in a form suitable for meeting applicable life-cycle phase success criteria. Realization is the act of implementing, integrating, verifying, validating, and transitioning the end product for use at the next level up of the system structure or to the customer. At this point, the end product can be referred to as a "realized product" or "realized end product."

Product verification proves that an end product (whether built, coded, bought, or reused) for any element within the system structure conforms to its requirements or specifications. Such specifications and other design description documentation establish the configuration baseline of that product, which may have to be modified at a later time. Without a verified baseline and appropriate configuration controls, such later modifications could be costly or cause major performance problems.

From a process perspective, product verification and validation may be similar in nature, but the objectives are fundamentally different. A customer is interested in whether the end product provided will do what the customer intended within the environment of use. Examination of this condition is validation. Simply put, the Product Verification Process answers the critical question, "Was the end product realized right?" The Product Validation Process addresses the equally critical question, "Was the right end product realized?" When cost effective and warranted by analysis, the expense of validation testing alone can be mitigated by combining tests to perform verification and validation simultaneously.

The outcome of the Product Verification Process is confirmation that the end product, whether achieved by implementation or integration, conforms to its specified requirements, i.e., verification of the end product. This subsection discusses the process activities, inputs, outcomes, and potential product deficiencies.

Differences between Verification and Validation Testing

Testing is a detailed evaluation method of both verification and validation

Verification Testing

Verification testing relates back to the approved requirements set (such as an SRD) and can be performed at different stages in the product life cycle. Verification tests are the official "for the record" testing performed on a system or element to show that it meets its allocated requirements or specifications including physical and functional interfaces. Verification tests use instrumentation and measurements and are generally accomplished by engineers, technicians, or operator-maintainer test personnel in a controlled environment to facilitate failure analysis.

Validation Testing

Validation relates back to the ConOps document. Validation testing is conducted under realistic conditions (or simulated conditions) on any end product to determine the effectiveness and suitability of the product for use in mission operations by typical users and to evaluate the results of such tests. It ensures that the system is operating as expected when placed in a realistic environment.

Differences between Verification, Qualification, Acceptance and Certification

Verification

Verification is a formal process, using the method of test, analysis, inspection or demonstration, to confirm that a system and its associated hardware and software components satisfy all specified requirements. The Verification program is performed once regardless of how many flight units may be generated (as long as the design doesn't change).

Qualification

Qualification activities are performed to ensure that the flight unit design will meet functional and performance requirements in anticipated environmental conditions. A subset of the verification program is performed at the extremes of the environmental envelope and will ensure the design will operate properly with the expected margins. Qualification is performed once regardless of how many flight units may be generated (as long as the design doesn't change).

Acceptance

A smaller subset of the verification program is selected as criteria for the acceptance program. The selected Acceptance activities are performed on each of the flight units as they are manufactured and readied for flight/use. An Acceptance Data Package is prepared for each of the flight units and shipped with the unit. The acceptance test/analysis criteria are selected to show that the manufacturing/workmanship of the unit conforms to the design that was previously verified/qualified. Acceptance testing is performed for each flight unit produced.

Certification

Certification is the audit process by which the body of evidence that results from the verification activities and other activities are provided to the appropriate certifying authority to indicate the design is certified for flight/use. The Certification activity is performed once regardless of how many flight units may be generated.

5.3.1 Process Description

Figure 5.3-1, taken from NPR 7123.1, provides a typical flow diagram for the Product Verification Process and identifies typical inputs, outputs, and activities to consider in addressing product verification.

5.3.1.1 Inputs

Key inputs to the process are:

- **The product to be verified:** This product will have been transitioned from either the Product Implementation Process or the Product Integration Process. The product will likely have been through at least a functional test to ensure it was assembled correctly. Any supporting documentation should be supplied with the product.
- **Verification plan:** This plan will have been developed under the Technical Planning Process and baselined before entering this verification.
- **Specified requirements baseline:** These are the requirements that have been identified to be verified for this product. Acceptance criteria should have been identified for each requirement to be verified.
- **Enabling products:** Any other products needed to perform the Product Verification Process. This may include test fixtures and support equipment.

Additional work products such as the ConOps, mission needs and goals, interface control drawings, testing standards and policies, and Agency standards and policies may also be needed to put verification activities into context.

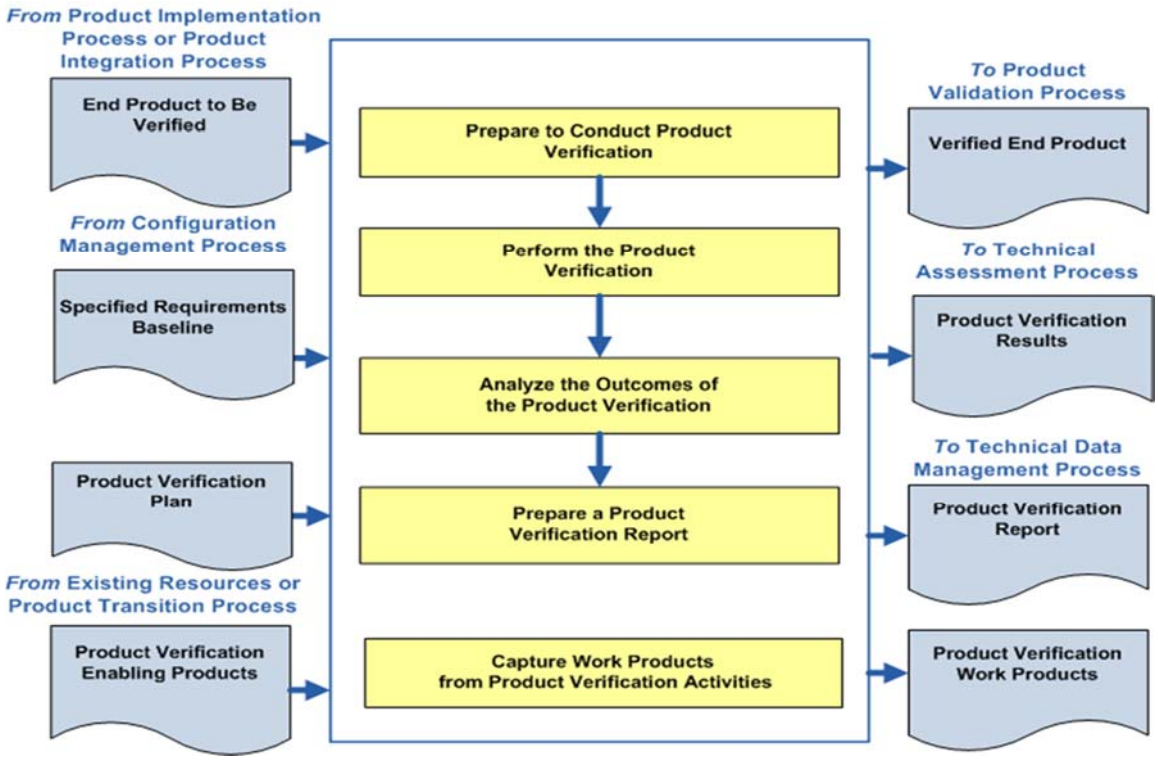


Figure 5.3-1 Product Verification Process

5.3.1.2 Process Activities

There are five major activities in the Product Verification Process: (1) prepare to conduct product verification; (2) perform verification; (3) analyze verification results; (4) preparing a product verification report; and (5) capture work products generated during the verification activities.

Product Verification is often performed by the developer that produced the end product with participation of the end user and customer. Quality Assurance (QA) personnel are also critical in the verification planning and execution activities.

5.3.1.2.1 Product Verification Preparation

In preparation for verification, the verification plan and the specified requirements are collected, reviewed, and confirmed. The product to be verified is obtained (output from the Product Implementation Process or the Product Integration Process) along with any enabling products, such as those representing external interfacing products and support resources (including personnel) that are necessary for verification. Procedures capturing detailed step-by-step activities and based on the verification type and methods are finalized and approved. Development of procedures typically begins during the design phase of the project life cycle and matures as the design is matured. The verification environment is considered as part of procedure development. Operational scenarios are assessed to explore all possible verification activities to be performed. The final element is preparation of the verification environment; e.g., facilities, equipment, tools, measuring devices, and climatic conditions.

When operator or other user interaction is involved, it is important to ensure that humans are properly represented in the verification activities. This includes physical size, skills, knowledge, training, clothing, special gear, and tools. Note: Testing that includes representatives of the human in the system is often referred to as "human-in-the-loop" testing.

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

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

Peer reviews are additional reviews that may be conducted formally or informally to ensure readiness for verification (as well as the results of the verification process). Guidelines for conducting a peer review are discussed in section 6.7.2.4.5.

Table 5.3-1 provides an example of the type of information that may be included in a verification procedure and a verification report:

Table 5.3-1 Example Information in Verification Procedures and Reports

Verification Procedure	Verification Report
Nomenclature and identification of the test article or material;	Verification objectives and the degree to which they were met;
Identification of test configuration and any differences from flight operational configuration;	Description of verification activity including deviations from nominal results (discrepancies);
Identification of objectives and criteria established for the verification by the applicable requirements specification;	Test configuration and differences from the flight operational configuration;
Characteristics and design criteria to be inspected, demonstrated, or tested, including values with tolerances for acceptance or rejection;	Specific result of each activity and each procedure, including the location or link to verification data/artifacts;
Description, in sequence, of steps, operations, and observations to be taken;	Specific result of each analysis including those associated with test-data analysis;
Identification of computer software required;	Test performance data tables, graphs, illustrations, and pictures;
Identification of measuring, test, and recording equipment to be used, specifying range, accuracy, and type;	Summary of nonconformance/discrepancy reports, including dispositions with approved

	corrective actions and planned retest activity if available;
Provision for recording equipment calibration or software version data;	Conclusions and recommendations relative to the success of verification activity;
Credentials showing that required computer test programs/support equipment and software have been verified prior to use with flight operational hardware;	Status of Government-Supplied Equipment (GSE) and other enabling support equipment as affected by test;
Any special instructions for operating data recording equipment or other automated test equipment as applicable;	Copy of the as-run procedure (may include redlines); and
Layouts, schematics, or diagrams showing identification, location, and interconnection of test equipment, test articles, and measuring points and any other associated design or configuration work products;	Authentication of test results and authorization of acceptability.
Identification of hazardous situations or operations;	
Precautions and safety instructions to ensure safety of personnel and prevent degradation of test articles and measuring equipment;	
Environmental and/or other conditions to be maintained with tolerances;	
Constraints on inspection or testing;	
Provision or instructions for the recording of verification results and other artifacts;	
Special instructions for instances of nonconformance and anomalous occurrences or results; and	
Specifications for facility, equipment maintenance, housekeeping, quality inspection, and safety and handling requirements before, during, and after the total verification activity.	

Outcomes of verification preparation include the following:

- The verification plan, approved procedures, and an appropriate baseline set of specified requirements and supporting configuration documentation is available and on hand;
- Articles/models to be verified and verification-enabling products are on hand, assembled, and integrated with the verification environment according to verification plans and schedules;
- The resources (funding, facilities, and people including appropriately skilled operators) needed to conduct the verification are available according to the verification plans and schedules; and

- The verification environment is evaluated for adequacy, completeness, readiness, and integration.

5.3.1.2.2 Perform Product Verification

The actual act of verifying the end product is performed as spelled out in the plans and procedures, and conformance is established with each specified product requirement. The verification lead should ensure that the procedures were followed and performed as planned, the verification-enabling products and instrumentation were calibrated correctly, and the data were collected and recorded for required verification measures.

When a "discrepancy" is observed (i.e., any variance, lack of agreement, or contradiction with the required or expected outcome, configuration, or result), verification activities should stop and a discrepancy report should be generated. The activities and events leading up to the discrepancy should be analyzed to determine if a nonconforming product exists or there is an issue with the verification procedure or conduct. The Decision Analysis Process should be used to make decisions with respect to needed changes in the verification plans, environment, and/or procedures.

Outcomes of performing product verification include the following:

- A verified product is established with supporting confirmation that the product (in the appropriate form for the life-cycle phase) complies with its specified requirements, and if it does not, a nonconformance report delineating the variance is available.
- A determination is made as to whether the appropriate results were collected and evaluated to show completion of verification objectives throughout their performance envelope.
- A determination is made that the verification product was appropriately integrated with the enabling products and verification environment.

5.3.1.2.3 Analyze Product Verification Results and Report

As the verification activities are completed, the results are collected and analyzed. The data are analyzed for quality, integrity, correctness, consistency, and validity. Any verification discrepancies (anomalies, variations, and out-of-compliance conditions) are identified and reviewed to determine if there is a nonconforming product not resulting from poor verification conduct, procedure, or conditions. If possible, this analysis is performed while the test/analysis configuration is still intact. This allows a quick turnaround in case the data indicates that a correction to the test or analysis run needs to be performed again.

Discrepancies and nonconforming products should be recorded and reported for follow-up action and closure. Verification results should be recorded in a requirements compliance or verification matrix or other method developed during the Technical Requirements Definition Process to trace compliance for each product requirement. Waivers needed as a result of verification to request relief from or modify a requirement are identified.

Note: Nonconformance and discrepancy reports may be directly linked with the Technical Risk Management Process. Depending on the nature of the nonconformance, approval through such bodies as a Material Review Board or Configuration Control Board (which typically includes risk management participation) may be required.

System design and product realization process activities may be required to resolve product nonconformance. If the mitigation of the nonconformance results in a change to the product, the verification may need to be planned and performed again.

Outcomes of analyzing the verification results include the following:

- Product nonconformance (not compliant with product requirement) is identified.
- Appropriate replanning, redefinition of requirements, redesign, implementation/integration, modification, and reverification have been accomplished for resolution of the nonconforming product.
- Appropriate facility modifications, procedure corrections, enabling product modification, and reverification have been performed for non-product-related discrepancies.
- Waivers for nonconforming products are accepted.
- Discrepancy and nonconformance reports including corrective actions have been generated as needed.
- The verification report is completed.

Reengineering

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

Reengineering could require the reapplication of the system design processes (Stakeholder Expectations Definition Process, Technical Requirements Definition Process, Logical Decomposition Process, and Design Solution Definition Process).

5.3.1.2.4 Capture Product Verification Work Products

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

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

Outcomes of capturing verification work products include the following:

- Verification of work products is recorded, e.g., method of verification, procedures, environments, outcomes, decisions, assumptions, corrective actions, and lessons learned.
- Variations, anomalies, and out-of-compliance conditions have been identified and documented, including the actions taken to resolve them.
- Proof that the realized end product did or did not satisfy the specified requirements is documented.
- The verification report is developed, including:
  - Recorded test/verification results/data;
  - Version of the set of specified requirements used;
  - Version of the product verified;
  - Version or standard for tools, data, and equipment used;
  - Results of each verification including pass or fail declarations; and
  - Discrepancies.

5.3.1.3 Outputs

Key outputs from the process are:

- **Verified product ready for validation:** After the product is verified, it will next pass through the Product Validation Process.
- **Product verification results**: Results from executed procedures are passed to technical assessment.
- **Product verification report(s):** A report shows the results of the verification activities. It includes the requirement that was to be verified and its bidirectional traceability, the verification method used, and reference to any special equipment, conditions, or procedures used. It also includes the results of the verification, any anomalies, variations or out-of-compliance results noted and associated corrective actions taken.
- **Product verification work products:** These include discrepancy and nonconformance reports with identified correction actions; updates to requirements compliance documentation; changes needed to the procedures, equipment or environment; configuration drawings; calibrations; operator certifications; and other records.

Criteria for completing verification of the product include: (1) documented objective evidence of compliance with requirements or waiver and (2) closure of all discrepancy and nonconformance reports.

5.3.2 Product Verification Guidance

5.3.2.1 Verification Approach

A verification approach should be adapted (tailored) to the project it supports. The project manager and systems engineer should work with the verification lead engineer to develop a verification approach and plan the activities. Many factors need to be considered in developing this approach and the subsequent verification program. These factors include:

- Project type, especially for flight projects. Verification activities and timing depend on the following:
  - The type of flight article involved (e.g., an experiment, payload, or launch vehicle).
  - For missions required to follow NPR 7120.5, NASA Space Flight Program and Project Management Requirements, NASA payload classification (NPR 8705.4, Risk Classification for NASA Payloads) guidelines are intended to serve as a starting point for establishing the formality of verification approaches that can be adapted to the needs of a specific project based on the "A-D" payload classification. Further flexibility is imparted to projects following NPR 7120.8, NASA Research and Technology Program and Project Management Requirements.
  - Project cost and schedule implications. Verification activities can be significant drivers of a project's cost and schedule, and these implications should be considered early in the development of the verification plan. Trade studies should be performed early in the life cycle to support decisions about verification methods and types and the selection of facility capabilities and locations. For example, a trade study might be made to decide between performing a test at a centralized facility or at several decentralized locations.
  - Risk management should be considered in the development of the verification approach. Qualitative risk assessments and quantitative risk analyses (e.g., a Failure Mode and Effects Analysis (FMEA)) often identify new concerns that can be mitigated by additional verifications, thus increasing the extent of verification activities. Other risk assessments contribute to trade studies that determine the preferred methods of verification to be used and when those methods should be performed. For example, a trade might be made between performing a model test versus determining model characteristics by a less costly but less revealing analysis. The project manager/systems engineer should determine what risks are acceptable in terms of the project's cost and schedule.
- Availability of verification facilities/sites and transportation assets to move an article from one location to another (when needed). This requires coordination with the Integrated Logistics Support (ILS) engineer.
- Availability of appropriately trained users for interaction with systems having human interfaces.
- Acquisition strategy; i.e., in-house development or system contract. A NASA field center can often shape a contractor's verification process through the project's SOW.
- Degree of design heritage and hardware/software reuse.

5.3.2.2 Verification in the Life Cycle

The method of verification completed will be a function of the life-cycle phase and the position of the product within the system structure. The end product should be verified and validated before it is transitioned to the next level up as part of the bottom-up realization process. See Figure 5.3-2. One element of an end product may be going through verification while another element is going through validation. While illustrated as separate processes in Figure 5.3-2, there

can be considerable overlap between verification and validation events when implemented. A verification configuration may also lend itself to performing a validation activity.

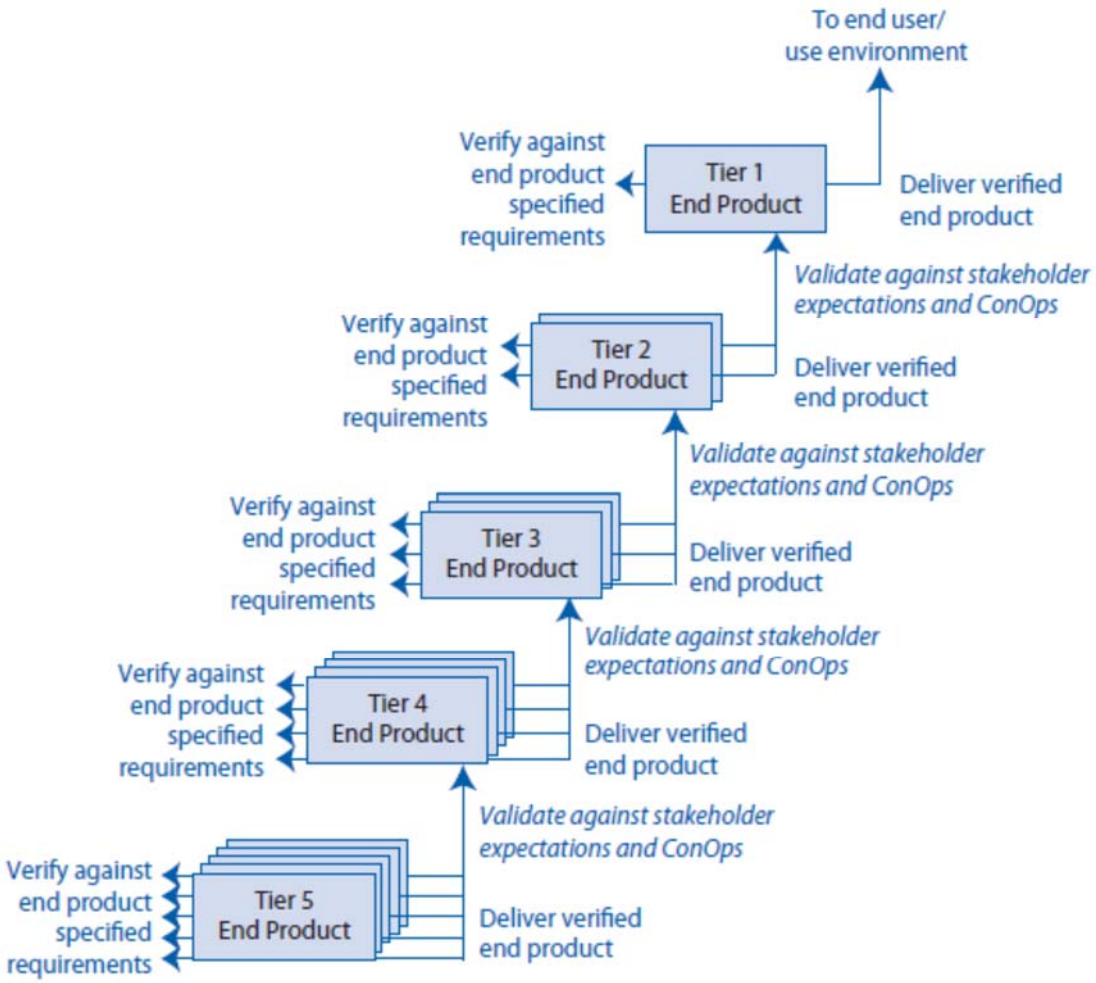


Figure 5.3-2 Bottom-up Product Realization Process

5.3.2.2.1 Quality Assurance in Verification

Even with the best of available designs, hardware fabrication, software coding, and testing, projects are subject to the vagaries of nature and human beings. The systems engineer needs to have some confidence that the system actually produced and delivered is in accordance with its requirements. The role of Quality Assurance (QA) is to provide an independent assessment to the project manager/systems engineer of the products produced and processes used during the project life cycle. The QA engineer typically acts as the systems engineer's eyes and ears in this context.

The QA engineer typically monitors fabrication, assembly, integration, verification, and validation activities and the resolution and closeout of nonconformance and discrepancy reports; verifies that the physical configuration of the system conforms to the build-to (or code-to) documentation; and collects and maintains QA data for subsequent failure analyses. The QA

engineer also participates in major reviews (primarily SRR, PDR, CDR, and MRR/FRR/PRR) on issues of design, parts, materials, workmanship, fabrication, and verification processes, and other characteristics that could degrade end product quality.

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

5.3.2.2.2 Qualification

Qualification activities are performed to ensure that the flight unit design will meet functional and performance requirements in anticipated environmental conditions. These activities begin after the baselining of flight hardware (and software) design and include analyses and testing. Qualification establishes the soundness of the design, manufacturing processes, and assembly processes, and demonstrates the design margin. Qualification tests generally subject the unit under test to worst-case loads and environmental requirements (maximum expected flight/operations levels, including the maximum number of cycles that can be accumulated during acceptance testing) plus a defined level of margin. Care should be exercised not to set test levels so that unrealistic failure modes are created. However, sometimes the qualification unit is tested to failure to determine performance limits. For additional information on qualification and environmental testing (except for radiation testing), see *MIL-STD-1540 Product Verification Requirements for Launch, Upper Stage, and Space Vehicles* or a Center standard such as *GSFC-STD-7000 General Environmental Verification Standard*.

Many performance requirements are verified and analyses and models are validated and updated as test data are acquired. Some of the verification activities are performed to ensure environmental compliance for vibration/acoustic, pressure limits, leak rates, thermal vacuum, thermal cycling, Electromagnetic Interference and Electromagnetic Compatibility (EMI/EMC), radiation, high-and low-voltage limits, and lifetime/cycling. Safety requirements, defined by hazard analysis reports, may also be satisfied by qualification testing.

Qualification usually occurs at the component or subsystem level to test facility limitations or to detect problems early in the realization process, but it could occur at the system level as well. If a project decides against building dedicated qualification hardware and uses the flight unit itself for qualification purposes, the final end product is referred to as “protoflight.” Here, the test parameters being used are typically less than those of qualification levels or durations but higher than those of acceptance levels (i.e., the margins are reduced).

Qualification verifies the soundness of the design. Test levels are typically set with some margin above expected flight/operations levels, including the maximum number of cycles that can be accumulated during acceptance testing. These margins are set to address design safety margins in general, and care should be exercised not to set test levels so that unrealistic failure modes are created.

5.3.2.2.3 Acceptance

Acceptance activities provide the assurance that the flight unit (hardware and software) is in compliance with all requirements and is ready for integration at the next level of the architecture. For the highest level (instrument or payload), it is ready for shipment to the launch site. Acceptance assures that the delivered product does not have latent material deficiencies and workmanship defects (i.e., quality of work), and that proper manufacturing processes and procedures were used. Environmental test levels are set to induce failures arising from defects in parts, materials, and workmanship. As such, test levels are those anticipated during flight / operations with no additional margin or levels established to stress hardware or drive out defects, whichever is greater. The acceptance begins with the acceptance of each individual component or piece part for assembly into the fully integrated flight unit, continuing through the System Acceptance Review (SAR). (See Section 6.7.2.1.) For additional information on qualification and environmental testing, see *MIL-STD-1540 Product Verification Requirements for Launch, Upper Stage, and Space Vehicles*.

Some verification procedures cannot be performed after a flight unit has been assembled and integrated, especially a large unit, due to inaccessibility or other practicality constraints. When this occurs, these verification procedures are performed during fabrication and integration, and are known as "in-process" inspections or tests, sometimes referred to as Government Mandatory Inspection Points (GMIP) in a procedure. In this case, acceptance begins with in-process verification and continues through unit-level verification.

Acceptance normally begins at the component level and continues to the mission system level, ending with all systems operating simultaneously. When the actual flight unit is unavailable, or its use is inappropriate for a specific test, simulators may be used to verify interfaces before mating flight hardware. Where appropriate, verification data can be used to validate and update analyses and models.

Discrepancies occurring during acceptance are documented on the appropriate reporting system with discrepancy/nonconformance reports, and a proposed resolution shou*ld be defined and approved by the project's Material Review Board or equivalent body before* acceptance continues. Disposition may require a collaborative effort of the systems engineer and the design, test and other organization. The disposition and approval process is typically captured in the project's quality assurance plan.

5.3.2.2.4 Deployment

The deployment verification begins with the arrival of the flight article (payload) at the launch site or other designated operational facility and concludes with delivery to and check-out at the site of operation. Deployment includes evaluation of in-flight performance of the flight article to determine whether the project is ready to conduct the mission (begin operations). The flight article is processed and integrated with the launch vehicle or other carrier, or the flight article could be part of the launch vehicle. Verification activities performed during this phase of the life cycle start with "checkout" or inspection to ensure that no visible damage or environmental overexposure to the flight article has occurred during shipment before applying power to the flight article. This is followed by verification activities to ensure that the end product continues to function properly before installation into or integration with the carrier.

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

5.3.2.2.5 Operation and Disposal

Operational verification after deployment when the product is at its site of operation provides the assurance that the system functions properly in the actual deployed environment. These verifications are performed through system activation and operation. Payloads or vehicles that are assembled on-orbit should have each interface verified and should function properly during end-to-end testing, which verifies both the on-orbit element and the ground element or the mission system. Mechanical interfaces that provide fluid and gas flow should be verified to ensure no leakage occurs and that pressures and flow rates are within specification. Environmental systems should be verified. This may serve as acceptance of the integrated system. The system activation or on-orbit checkout is done to ensure the deployed system is ready to be transitioned into formal operations.

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

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

5.3.2.3 Verification Procedures

End product verification procedures provide step-by-step instructions for performing a given verification activity. The procedure to be used is written, reviewed, and approved in accordance with the verification plan and submitted to the Test Readiness Review (TRR) for each verification activity. (See Test Readiness Review discussion in Section 6.7.2.1.) Procedures are also used to verify the acceptance of facilities, electrical and mechanical ground support equipment, and special test equipment (enabling products).

The information generally contained in a procedure is as follows, but it may vary according to the activity and test article:

- Nomenclature and identification of the test article or material;
- Identification of test configuration and any differences from flight operational configuration;

- Identification of objectives and criteria established for the verification by the applicable requirements specification;
- Characteristics and design criteria to be inspected, demonstrated, or tested, including values with tolerances for acceptance or rejection;
- Description, in sequence, of steps, operations, and observations to be taken;
- Identification of computer software required;
- Identification of measuring, test, and recording equipment to be used, specifying range, accuracy, and type;
- Provision for recording equipment calibration or software version data;
- Credentials showing that required computer test programs/support equipment and software have been verified prior to use with flight operational hardware;
- Any special instructions for operating data recording equipment or other automated test equipment as applicable;
- Layouts, schematics, or diagrams showing identification, location, and interconnection of test equipment, test articles, and measuring points and any other associated design or configuration work products;
- Identification of hazardous situations or operations;
- Precautions and safety instructions to ensure safety of personnel and prevent degradation of test articles and measuring equipment;
- Environmental and/or other conditions to be maintained with tolerances;
- Constraints on inspection or testing;
- Provision or instructions for the recording of verification results and other artifacts;
- Special instructions for instances of nonconformance and anomalous occurrences or results; and
- Specifications for facility, equipment maintenance, housekeeping, quality inspection, and safety and handling requirements before, during, and after the total verification activity.

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

5.3.2.4 Verification Reports

A verification report should be provided for each analysis and, at a minimum, for each verification activity conducted on qualification and flight units, including functional testing, environmental testing, deployment, and end-to-end compatibility testing. Reports may be provided for developmental verification as a design record. Verification reports may be needed for each individual test activity, such as functional testing, acoustic testing, vibration testing, and

thermal vacuum/thermal balance testing. Multiple activities can be included in a single report if they are conducted jointly/concurrently or sequentially over a short time period. The verification plan should establish the need and use of reports. Verification reports should be completed within a few weeks following a verification activity and should provide evidence of compliance with the end product requirements for which it was conducted.

The verification report should include the following as appropriate:

- Verification objectives and the degree to which they were met;
- Description of verification activity including deviations from nominal results (discrepancies);
- Test configuration and differences from the flight operational configuration;
- Specific result of each activity and each procedure, including the location or link to verification data/artifacts;
- Specific result of each analysis including those associated with test-data analysis;
- Test performance data tables, graphs, illustrations, and pictures;
- Summary of nonconformance/discrepancy reports, including dispositions with approved corrective actions and planned retest activity if available;
- Conclusions and recommendations relative to the success of verification activity;
- Status of Government-Supplied Equipment (GSE) and other enabling support equipment as affected by test;
- Copy of the as-run procedure (may include redlines); and
- Authentication of test results and authorization of acceptability.

5.3.2.5 End-to-End System Testing

The purpose of end-to-end testing is to demonstrate interface compatibility and desired total functionality among different elements of a mission system, between systems (the system of interest and external enabling systems), and within a system as a whole. It can involve real or representative input and operational scenarios. End-to-end tests performed on the integrated ground and flight assets include all elements of the flight article (payload or vehicle), its control, stimulation, communications, and data processing to demonstrate that the entire integrated mission system is operating in a manner to fulfill all mission requirements and objectives. End-to-end tests may be performed as part of investigative engineering tests, verification testing, or validation testing. These are some of the most important tests for the systems engineers to participate in or to lead. They review the overall compatibility of the various systems and demonstrate compliance with system-level requirements and whether the system behaves as expected by the stakeholders.

Note: It is important to understand that over the lifetime of a system, requirements may change or component obsolescence may make a design solution too difficult to produce from either a cost or technical standpoint. In these instances, it is critical to employ the systems engineering design processes at a lower level to ensure the modified design provides a proper design solution. An evaluation should be made to determine the magnitude of the change required, and the process should be tailored to address the issues appropriately. A modified qualification, verification, and validation process may be required to baseline a new design solution, consistent with the intent previously described for those processes. The acceptance testing will also need to be updated as necessary to verify that the new product has been manufactured and coded in compliance with the revised baselined design.

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

Figure 5.3-3 presents an example of an end-to-end data flow for a scientific satellite mission. Each arrow in the diagram represents one or more data or control flows between two hardware, software, subsystem, or system configuration items. End-to-end testing verifies that the data flows throughout the multisystem environment are correct, that the system provides the required functionality, and that the outputs at the eventual end points correspond to expected results. Since the test environment is as close an approximation as possible to the operational environment, system performance requirements testing is also included. This figure is not intended to show the full extent of end-to-end testing. Each system shown would need to be broken down into a further level of granularity for completeness.

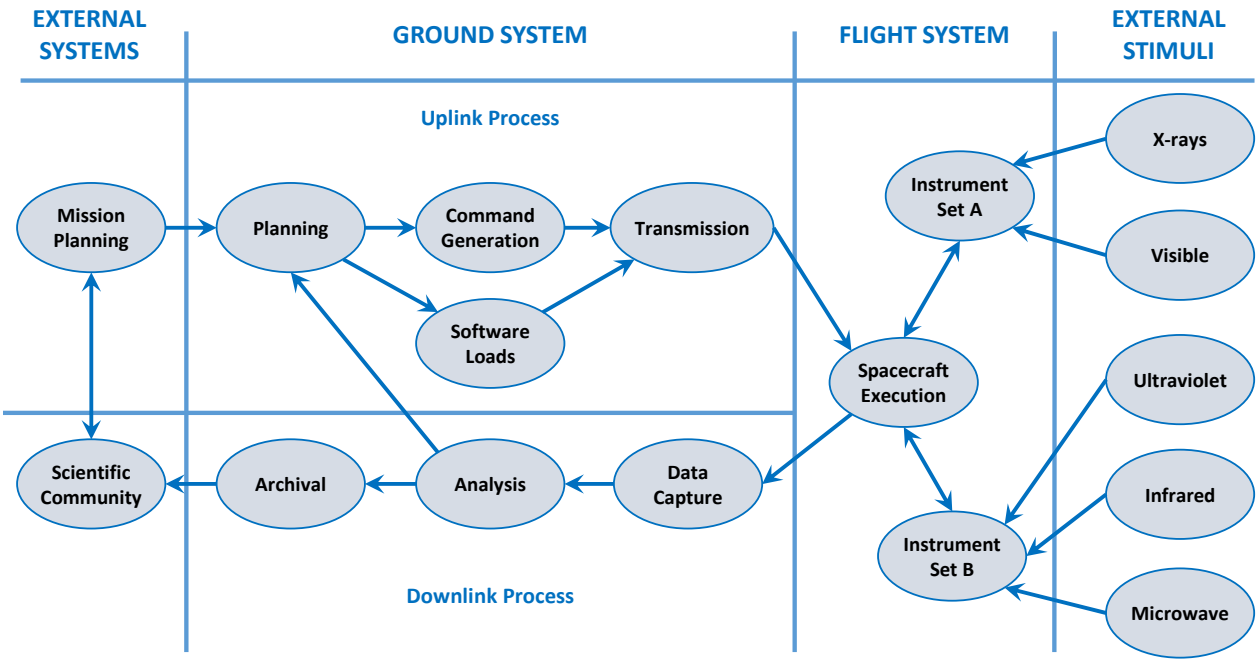


Figure 5.3-3 Example of End-to-End Data Flow for a Scientific Satellite Mission

End-to-end testing is an integral part of the verification and validation of the total (mission) system. It is a set of activities that can be employed during selected hardware, software, and system phases throughout the life cycle using developmental forms and external simulators. However, final end-to-end testing should be done on the flight articles in the flight configuration if possible and prior to deployment and launch. In comparison with configuration item testing, end-to-end testing addresses each configuration item (end product) only down to the level designated by the verification plan (generally, the segment r element) and focuses on external interfaces, which can be either hardware, software, or human-based. Internal interfaces (e.g., software subroutine calls, analog-to-digital conversion) of a designated configuration item are not within the scope of end-to-end testing.

5.3.2.5.1 How to Perform End-to-End Testing

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

End-to-end testing should include testing under possible failure scenarios, when possible and practical, particularly when fault detection and correction are required of the system. Specialized test capabilities to introduce faults within nominal system operations are often required.

Approaches for end-to-end testing of fault management applications include simulating fault conditions in the system (either in a prototype or in a simulation) and then running through nominal and failure scenarios to verify that fault management components (such as detection, diagnosis, location, prediction, recovery, and response) function correctly. Careful selection and prioritization of fault scenarios ensures adequate coverage of the fault space (possible set of faults) for end-to-end testing without an overly expansive test space (configurations and activities).

The end-to-end test encompasses the entire chain of flight and operational products and operations that will occur during all mission modes in such a manner as to ensure that the system will fulfill mission requirements. The mission environment should be simulated as realistically as possible, and the instruments should receive stimuli of the kind they will receive during the mission. The Radio Frequency (RF) links, ground station operations, and software functions should be fully exercised. When acceptable simulation facilities are available for portions of the operational systems, they may be used for the test instead of the actual system elements. The specific environments under which the end-to-end test is conducted and the stimuli, payload configuration, RF links, and other system elements to be used should be determined in accordance with the characteristics of the mission and captured in the verification plan.

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

Plans, agreements, and facilities should be put in place well in advance of the test to enable end-to-end testing between all components of the system.

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

> Note: End-to-end testing is particularly important when missions are developed with international or external partners.

As part of end-to-end testing, the following activities are completed for each designated configuration item (end product) within the scope of the test:

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

End-to-end testing activities include the following:

1. Operational scenarios are created that span all of the following items (during nominal, off-nominal, and stressful conditions) that could occur during the mission:

- Mission phase, mode, and state transitions;
- First-time events;
- Operational performance limits;
- Fault protection routines;
- Failure Detection, Isolation, and Recovery (FDIR) logic;
- Safety properties;
- Operational responses to transient or off-nominal sensor signals; and
- Communication uplink and downlink.

2. The operational scenarios are used to test the configuration items, interfaces, and end-to-end performance as early as possible in the configuration items' development life cycle. This typically means simulators or software stubs have to be created to implement a full scenario. It is extremely important to produce a skeleton of the actual system to run full scenarios as soon as possible with both simulated / stubbed-out and actual configuration items.
3. A complete diagram and inventory of all interfaces are documented.
4. Test cases are executed to cover human-human, human-hardware, human-software, hardware-software, software-software, and subsystem-subsystem interfaces and associated inputs, outputs, and modes of operation (including safing modes).
5. It is strongly recommended that during end-to-end testing, an operations staff member who has not previously been involved in the testing activities be designated to exercise the system as it is intended to be used to determine if it will fail to meet its requirements or exhibit off-nominal response.
6. The test environment should approximate/simulate the actual operational conditions when possible. The fidelity of the test environment should be authenticated. Differences between the test and operational environment should be documented in the verification plan or test report.
7. When testing of a requirement is not possible, verification is achieved by other means; i.e., by analysis including modeling and simulation. If true end-to-end testing cannot be achieved, then the testing should be done piecemeal and patched together by analysis. An example of this would be a system that is assembled onorbit where the various elements come together for the first time onorbit.
8. When a defect or nonconformance is identified in the developed system and fixed (i.e., appropriately dispositioned), regression testing of the system or component is performed to ensure that modifications have not caused unintended effects and that the system or component still complies with previously tested specified requirements.
9. When tests are aborted or a test is known to be flawed (e.g., due to configuration or test environment), the test should be rerun after the identified problem is fixed.
10. Prior to system delivery, test cases should be executed to cover all of the operations documented in the operations plan in the order in which they are expected to occur during the mission. The operational scenarios should be used to formulate and validate the final operations plan and procedures.

End-to-end test documentation includes the following:

- End-to-end testing plans as a part of the verification plan.
- A document, matrix, or database under configuration control that traces the end-to-end system test suite to the results. Data that are typically recorded include the test-case identifier, subsystems/hardware/ program sets exercised, list of the requirements being verified, interfaces exercised, date, and outcome of test (i.e., whether the test actual output met the expected output).
- End-to-end test cases and procedures (including inputs and expected outputs).
- A record of end-to-end problems/failures/anomalies (discrepancy reports for qualification and flight products).

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

5.3.2.6 Modeling and Simulation

For the Product Verification Process, a model is a physical, mathematical, or logical representation of an end product to be verified. Modeling and Simulation (M&S) can be used to augment and support the Product Verification Process and is an effective tool for performing the verification whether in early life-cycle phases or later. A physical model may be an early development phase representation or form of an end product (prototype or engineering development unit) used to guide or verify design. It can also be used to simulate the form, fit, and/or behavior (function and performance) of an end item when interacting with another end item or model thereof. A mathematical or logical representation can be used to support verification by analysis, which can be used as an alternative to testing when testing is impractical or impossible. The model/simulation uncertainties will need to be accounted for in order to satisfy a verification requirement when using this method.

Both the facilities associated with the model and the model itself are developed using the system design and product realization processes. The model used, as well as the M&S facility, are enabling products and should use the 17 technical processes (see Figure 2.1-1 or chapter 3 of NPR 7123.1, NASA Systems Engineering Processes and Requirements) for their development and realization (including acceptance by the operational community) to ensure that the model and simulation adequately represent the physical characteristics or function and performance of the modeled end product. When a model interacts with a qualification or flight unit, it must be subjected to configuration control and verified/validated itself. Additionally, in some cases certification is required before models and simulations can be used.

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

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

For additional information refer to *NASA-STD-7009, Standard for Models and Simulations.*

5.3.2.7 Hardware-in-the-Loop

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

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

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

5.4 Product Validation

The Product Validation Process is the second of the verification and validation processes conducted on an implemented or integrated end product. While verification proves whether "the product was done right," validation proves whether "the right product was done." In other words, verification provides objective evidence that every "shall" statement in the requirements document or specification was met, whereas validation is performed for the benefit of the customers and users to ensure that the system functions in the expected manner when placed in the intended environment. This is achieved by examining the products of the system at every level of the product structure and comparing them to the stakeholder expectations for that level. A well-structured validation process can save cost and schedule while meeting the stakeholder expectations.

System validation confirms that the integrated realized end products conform to stakeholder expectations as captured in the MOEs, MOPs, and ConOps. Validation also ensures that any anomalies discovered are appropriately resolved prior to product delivery. This section discusses the process activities, methods of validation, inputs and outputs, and potential deficiencies.

Distinctions between Product Verification and Product Validation. From a process activities perspective, product verification and product validation may appear to be similar in nature, but the objectives are fundamentally different. A customer's interest is in whether the end product provided will do what the customer intends within the environment of use. Examination of this condition is validation. When cost-effective and warranted by analysis, various combined tests are used, and verification and validation can be performed simultaneously. The expense of validation testing alone can be mitigated by ensuring that each end product in the system structure was correctly realized in accordance with its specified requirements (verified) before conducting validation. This subsection discusses the process activities, inputs, outcomes, and potential product deficiencies of validation.

5.4.1 Process Description

Figure 5.4-1, taken from NPR 7123.1, provides a typical flow diagram for the Product Validation Process and identifies typical inputs, outputs, and activities to consider in addressing product validation.

5.4.1.1 Inputs

Key inputs to the process are:

- **End product to be validated**: This is the end product that is to be validated and which has successfully passed through the verification process.
- **Validation plan:** This plan would have been developed under the Technical Planning Process and baselined prior to entering this process. This plan may be a separate document or a section within the Verification and Validation Plan.
- **Baselined stakeholder expectations:** These would have been developed for the product at this level during the Stakeholder Expectations Definition Process. It includes the needs, goals, and objectives as well as the baselined and updated concept of operations and MOEs.

- **Any enabling products:** These are any special equipment, facilities, test fixtures, applications, or other items needed to perform the Product Validation Process.

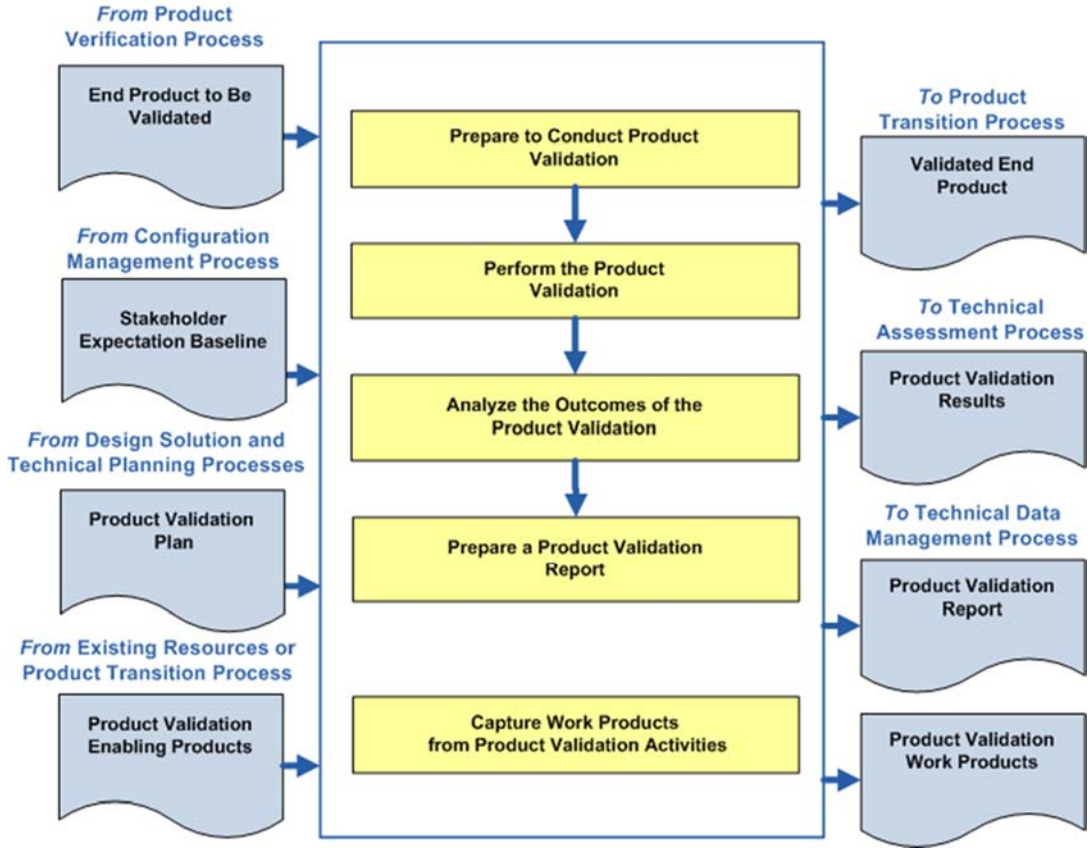


Figure 5.4-1 Product Validation Process

5.4.1.2 Process Activities

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

There are five major steps in the validation process: (1) preparing to conduct validation, (2) conduct planned validation (perform validation), (3) analyze validation results, (4) prepare a validation report, and (5) capture the validation work products.

The objectives of the Product Validation Process are:

- To confirm that the end product fulfills its intended use when operated in its intended environment:
  - Validation is performed for each implemented or integrated and verified end product from the lowest end product in a system structure branch up to the top level end product (the system).

- Evidence is generated as necessary to confirm that products at each layer of the system structure meet the capability and other operational expectations of the customer / user / operator and other interested parties for that product.

- To ensure the human has been properly integrated into the system:
  - The user interface meets human engineering criteria.
  - Operators and maintainers have the required skills and abilities.
  - Instructions are provided and training programs are in place.
  - The working environment supports crew health and safety.
- To ensure that any problems discovered are appropriately resolved prior to delivery of the end product (if validation is done by the supplier of the product) or prior to integration with other products into a higher level assembled product (if validation is done by the receiver of the product).

5.4.1.2.1 Product Validation Preparation

To prepare for performing product validation, the appropriate set of expectations, including MOEs and MOPs, against which the validation is to be made should be obtained. In addition to the V&V Plan, other documentation such as the ConOps and HSI Plan may be useful. The product to be validated (output from implementation, or integration and verification), as well as the appropriate enabling products and support resources (requirements identified and acquisition initiated by design solution activities) with which validation will be conducted should be collected. Enabling products includes those representing external interfacing products and special test equipment. Support resources include personnel necessary to support validation and operators. Procedures, capturing detailed step-by-step activities and based on the validation type and methods are finalized and approved. Development of procedures typically begins during the design phase of the project life cycle and matures as the design is matured. The validation environment is considered as part of procedure development. Operational scenarios are assessed to explore all possible validation activities to be performed. The final element is preparation of the validation environment; e.g., facilities, equipment, software, and climatic conditions.

When operator or other user interaction is involved, it is important to ensure that humans are properly represented in the validation activities. This includes physical size, skills, knowledge, training, clothing, special gear, and tools. When possible, actual end users/operators should be used and other stakeholders should participate or observe activities as appropriate and practical.

Outcomes of validation preparation include the following:

- The validation plan, approved procedures, supporting configuration documentation, and an appropriate baseline set of stakeholder expectations are available and on hand;
- Enabling products are integrated within the validation environment according to plans and schedules;
- Users/operators and other resources are available according to validation plans and schedules; and

- The validation environment is evaluated for adequacy, completeness, readiness, and integration.

5.4.1.2.2 Perform Product Validation

The act of validating the end product is performed as spelled out in the validation plans and procedures, and the conformance established to each specified stakeholder expectation (MOEs and ConOps) shows that the validation objectives were met. Validation differs from qualification testing. Validation testing is focused on the expected environments and operations of the system where as qualification testing includes the worst case loads and environmental requirements within which the system is expected to perform or survive. The verification lead should ensure that the procedures were followed and performed as planned, the validation-enabling products and instrumentation were calibrated correctly, and the data were collected and recorded for required validation measures.

When a discrepancy is observed, the validation should be stopped and a discrepancy report generated. The activities and events leading up to the discrepancy should be analyzed to determine if a nonconforming product exists or there is an issue with the verification procedure, conduct, or conditions. If there are no product issues, the validation is replanned as necessary, the environment preparation anomalies are corrected, and the validation is conducted again with improved or correct procedures and resources. The Decision Analysis Process should be used to make decisions with respect to needed changes to the validation plans, environment, and/or conduct.

Outcomes of performing validation include the following:

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

5.4.1.2.3 Analyze Product Validation Results

Once the validation activities have been completed, the results are collected and the data are analyzed to confirm that the end product provided will supply the customer's needed capabilities within the intended environments of use, validation procedures were followed, and enabling products and supporting resources functioned correctly. The data are also analyzed for quality, integrity, correctness, consistency, and validity, and any unsuitable products or product attributes are identified and reported.

It is important to compare the actual validation results to the expected results. If discrepancies are found, it needs to be determined if they are a result of the test configuration or analysis

assumptions or whether they are a true characteristic or behavior of the end product. If it is found to be a result of the test configuration, the configuration should be corrected and the validation repeated. If it is found to be a result of the end product being validated, discussions with the customer should be held and any required system design and product realization process activities should be conducted to resolve deficiencies. The deficiencies along with recommended corrective actions and resolution results should be recorded, and validation should be repeated, as required.

Outcomes of analyzing validation results include the following:

- Product anomalies, variations, deficiencies, nonconformance and/or issues are identified.
- Assurances that appropriate replanning, redefinition of requirements, design, and revalidation have been accomplished for resolution of anomalies, variations, deficiencies or out-of-compliance conditions (for problems not caused by poor validation conduct).
- Discrepancy and corrective action reports are generated as needed.
- The validation report is completed.

Reengineering

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

Validation Deficiencies

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

- Reengineering end products lower in the system structure that make up the end product that was found to be deficient (i.e., that failed to satisfy validation requirements); and/or
- Re-performing any needed verification and validation processes.

Other reasons for validation deficiencies (particularly when M&S are involved) may be incorrect and/or inappropriate initial or boundary conditions; poor formulation of the modeled equations or behaviors; the impact of approximations within the modeled equations or behaviors; failure to provide the required geometric and physics fidelities needed for credible simulations for the intended purpose; and/or poor spatial, temporal, and perhaps, statistical resolution of physical phenomena used in M&S.

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

Of course, the ultimate reason for performing validation is to determine if the design itself is the right design for meeting stakeholder expectations. After any and all validation test deficiencies are ruled out, the true value of validation is to identify design changes needed to ensure the program / product's mission. Validation should be performed as early and as iteratively as possible in the SE process since the earlier reengineering needs are discovered, the less expensive they are to resolve.

Pass Verification but Fail Validation?

Sometimes systems successfully complete verification but then are unsuccessful in some critical phase of the validation process, delaying development and causing extensive rework and possible compromises with the stakeholder. Developing a solid ConOps in early phases of the project (and refining it through the requirements development and design phases) is critical to preventing unsuccessful validation. Similarly, developing clear expectations for user community involvement in the HSI Plan is critical to successful validation. Frequent and iterative communications with stakeholders helps to identify operational scenarios and key needs that should be understood when designing and implementing the end product. Should the product fail validation, redesign may be a necessary reality. Review of the understood requirements set, the existing design, operational scenarios, user population numbers and skills, training, and support material may be necessary, as well as negotiations and compromises with the customer, other stakeholders, and/or end users to determine what, if anything, can be done to correct or resolve the situation. This can add time and cost to the overall project or, in some cases, cause the project to fail or be cancelled. However, recall from Figure 2.5-3 that the earlier design issues are discovered, the less costly the corrective action.

5.4.1.2.4 Prepare Report and Capture Product Validation Work Products

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

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

Outcomes of capturing validation work products include the following:

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

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

> Note: For systems where only a single deliverable item is developed, the Product Validation Process normally completes acceptance testing of the system. However, for systems with several production units, it is important to understand that continuing verification and validation is not an appropriate approach to use for the items following the first deliverable. Instead, acceptance testing is the preferred means to ensure that subsequent deliverables meet stakeholder expectations.

5.4.1.3 Outputs

Key outputs of validation are:

- **Validated end product:** This is the end product that has successfully passed validation and is ready to be transitioned to the next product layer or to the customer.
- **Product validation results:** These are the raw results of performing the validations.
- **Product validation report:** This report provides the evidence of product conformance with the stakeholder expectations that were identified as being validated for the product at this layer. It includes any nonconformance, anomalies, or other corrective actions that were taken.
- **Work product**s: These include procedures, required personnel training, certifications, configuration drawings, and other records generated during the validation activities.

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

5.4.2 Product Validation Guidance

The following is some generic guidance for the Product Validation Process.

5.4.2.1 Modeling and Simulation

As stressed in the verification process material, M&S is also an important validation tool. M&S usage considerations involve the verification, validation, and certification of the models and simulations.

Model Verification and Validation

- **Model Verification:** Degree to which a model accurately meets its specifications. Answers "Is it what I intended?"
- **Model Validation:** The process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model.
- **Model Certification:** Certification for use for a specific purpose. Answers, "Should I endorse this model?"

5.4.2.2 Software

Software verification is a software engineering activity that demonstrates that the software products meet specified requirements. Methods of software verification include peer reviews/inspections of software engineering products for discovery of defects, software verification of requirements by use of simulations, black box and white box testing techniques, software load testing, software stress testing, software performance testing, decision table-based testing, functional decomposition-based testing, acceptance testing, path coverage testing, analyses of requirement implementation, and software product demonstrations.

Software validation is a software engineering activity that demonstrates the as-built software product or software product component satisfies its intended use in its intended environment. Methods of software validation include formal reviews, prototype demonstrations, functional demonstrations, software testing, software peer reviews/inspections, behavior in a simulated environment, acceptance testing against mathematical models, analyses, and operational environment demonstrations.

The rigor and techniques used to verify and validate software depend upon software classifications (which are different from project and payload classifications). A complex project will typically contain multiple systems and subsystems having different software classifications. It is important for the project to classify its software and plan verification and validation approaches that appropriately address the risks associated with each class. Specific Agency-level requirements for software verification and validation, peer reviews (see appendix N), testing, and reporting are contained in NPR 7150.2, NASA Software Engineering Requirements.

In some instances, a project is required or is selected for additional independent software verification and validation (IV&V) support. In cases where IV&V is required, an IV&V Project Execution Plan (IPEP) is developed. The scope of IV&V services is determined by the project and the IV&V provider and is documented in the IPEP. The IPEP is developed by the IV&V provider and serves as the operational document that is shared with the project receiving IV&V support. In accordance with the responsibilities defined in NPD 7120.4, NASA Engineering and Program/Project Management Policy, projects ensure that software providers allow access to

software and associated artifacts to enable implementation of IV&V. The Agency-level requirement to deter the need for software IV&V is contained in NPR 7150.2, NASA Software Engineering Requirements.

5.4.2.3 Taking Credit for Validation

Validation is one of the under-utilized or unacknowledged techniques in the evaluation of an end product. In many cases, programs/projects have a V&V Plan that contains only verification activities. However, most projects actually perform validation in one form or another, but it is not always preplanned or formally acknowledged. Such validation is often called a "functional test" or an "engineering test" because of the perception that official testing should only be to requirements. However, many of these tests or analyses are true validation tests and credit could be taken for them if they were preplanned (i.e., part of the V&V Plan), conducted in a relevant environment, and results officially recorded and evaluated.

The official acknowledgement and proper use of validation is a good way to reduce costs. For example, say one aspect of the end product is "the chair is soft." In a verification world, that would mean that the term "soft" would have to be translated into several dozen derived discrete "shall" statements (e.g., The padding in the seat shall be 2 inches or greater, the springs in the seat shall have a 1-inch minimum compression, etc.). Each one of these derived requirements then requires official verification testing, evaluation, analysis, documentation, etc. However, it could be more cost-effective if it is planned early that instead of generating many derived requirements trying to quantify "softness," a validation test/demonstration will be performed using a planned number of test subjects to sit in the chair for the expected amount of time and say if they felt it was "soft." This would be done early on prototypes so that the reactions of the test subjects could be incorporated into the final designs. It would likely then be repeated on other versions as necessary as the design matures.

If customers and stakeholders know that at defined places in the design and development cycles they will get an opportunity to evaluate products, they may also be less vehement about trying to get everything defined as a requirement in extreme detail and be more assured that they will be getting a product they can use.

Not only does product validation help ensure that the stakeholders expectations are met, it saves cost and schedule by reducing the number of verifications that have to be done and the time and expense of having to figure out how to translate qualitative expectations such as "softness" or "readability," and it mitigates customer anxiety on the usability of the end product.

5.5 Product Transition

The Product Transition Process is used to transition a verified and validated end product that has been generated by product implementation or product integration to the customer at the next level in the system structure for integration into an end product or, for the top-level end product, transitioned to the intended end user. The form of the product transitioned will be a function of the product life-cycle phase success criteria and the location within the system structure of the WBS model in which the end product exists. The systems engineer involvement in this process includes ensuring the product being transitioned has been properly tested and verified/validated prior to being shipped to the next level stakeholder/customer.

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

The Product Transition Process is the key to bridge from one activity, subsystem, or element to the overall engineered system. As the system development nears completion, the Product Transition Process is again applied for the end product, but with much more rigor since now the transition objective is delivery of the system-level end product to the actual end user. Depending on the kind or category of system developed, this may involve a Center or the Agency and impact thousands of individuals storing, handling, and transporting multiple end products; preparing user sites; training operators and maintenance personnel; and installing and sustaining, as applicable. Examples are transitioning the external tank, solid rocket boosters, and orbiter to Kennedy Space Center (KSC) for integration and flight. Another example is the transition of a software subsystem for integration into a combined hardware/software system.

5.5.1 Process Description

Figure 5.5-1 provides a typical flow diagram for the Product Transition Process and identifies typical inputs, outputs, and activities to consider in addressing product transition.

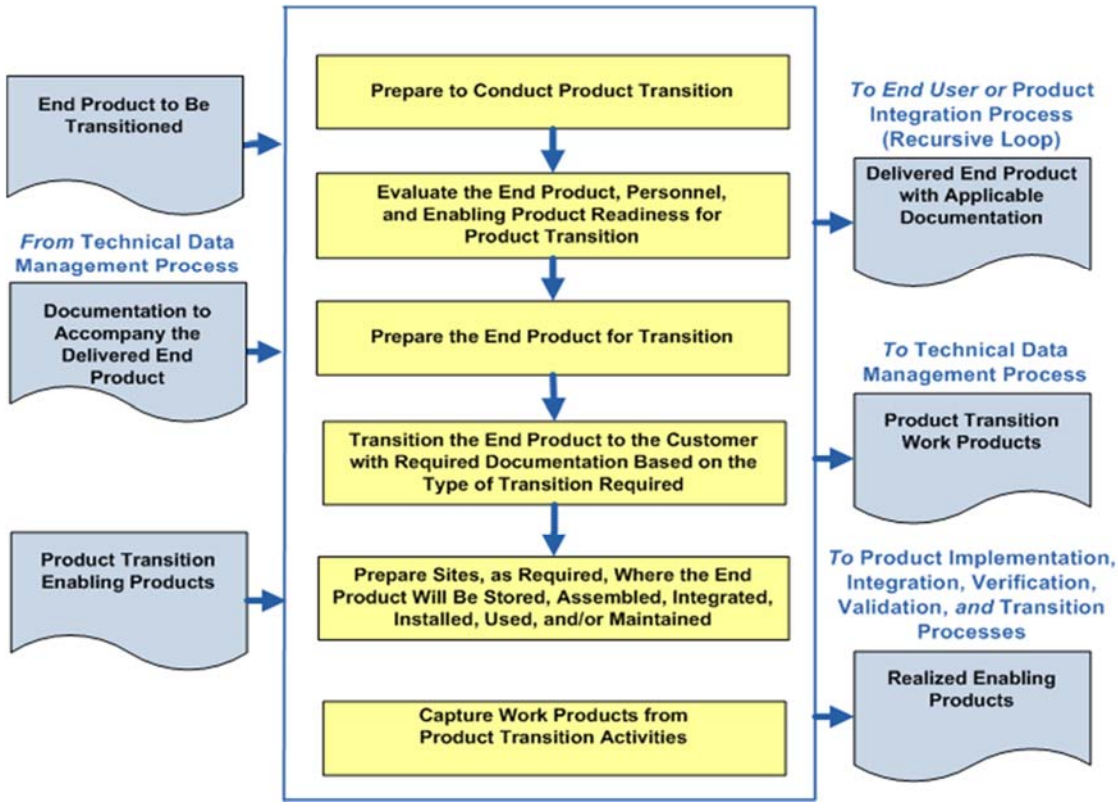


Figure 5.5-1 Product Transition Process

5.5.1.1 Inputs

Inputs to the Product Transition Process depend primarily on the transition requirements, the product that is being transitioned, the form of the product transition that is taking place, and the location to which the product is transitioning. Typical inputs are shown in Figure 5.5-1 and described below.

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

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

- **Documentation including manuals, procedures, and processes that are to accompany the end product (from the Technical Data Management Process)**: The documentation required for the Product Transition Process depends on the specific end product; its current location within the system structure; and the requirements identified in various agreements, plans, or requirements documents. Typically, a product has a unique identification (i.e., serial or version number) and may have a pedigree (documentation) that specifies its heritage and current state. Pertinent information may be controlled using a configuration control process or work order system as well as design drawings and test reports. Documentation often includes proof of verification and validation conformance. A COTS product would typically contain a manufacturer's specification or fact sheet. Documentation may include operations manuals, installation instructions, and other information.

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

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

- **Product transition-enabling products, including packaging materials; containers; handling equipment; and storage, receiving, and shipping facilities (from existing resources or the Product Transition Process for enabling product realization)**: Product transition-enabling products may be required to facilitate the implementation, integration, evaluation, transition, training, operations, support, and/or retirement of the transition product at its next higher level or for the transition of the final end product. Some or all of the enabling products may be defined in transition-related agreements, system requirements documents, or project plans. In some cases, product transition-enabling products are developed during the realization of the product itself or may be required to be developed during the transition stage.

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

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

5.5.1.2 Process Activities

Transitioning the product can take one of two forms:

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

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

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

Figure 5.5-1 shows what kind of inputs, outputs, and activities are performed during product transition regardless of where in the product hierarchy or life cycle the product is. These activities include preparing to conduct the transition; making sure the end product, all personnel, and any enabling products are ready for transitioning; preparing the site; and performing the transition including capturing and documenting all work products.

How these activities are performed and what form the documentation takes depends on where the end items are in the product hierarchy and the life-cycle phase.

Refer to Section 7.1 for special considerations when end items to be integrated into a larger program or system are obtained through a diversity of acquisition contract mechanisms.

5.5.1.2.1 Prepare to Conduct Transition

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

Other tasks in preparing to transition a product involve making sure the end product, personnel, and any enabling products are ready for that transition. This includes the availability of the documentation or models that will be sent with the end product, including proof of verification and validation conformance. The appropriateness of detail for that documentation depends upon

where the product is within the product hierarchy and the life cycle. Early in the life cycle, this documentation may be preliminary in nature. Later in the life cycle, the documentation may be detailed design documents, user manuals, drawings, or other work products. Procedures necessary for conducting the transition should be reviewed and approved by this time.

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

5.5.1.2.2 Prepare the Site to Receive the Product

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

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

A site survey may need to be performed to determine the issues and needs. This should address the adequacy of existing facilities to accept, store, and operate the new end product and identify any logistical-support-enabling products and services required but not planned for. Additionally, any modifications to existing facilities should be planned well in advance of fielding; therefore, the site survey should be made during an early phase in the product life cycle. These may include logistical enabling products and services to provide support for end-product use, operations, maintenance, and disposal. Training for users, operators, maintainers, and other support personnel may need to be conducted. National Environmental Policy Act documentation or approvals may need to be obtained prior to the receipt of the end product.

Prior to shipment or after receipt, the end product may need to be stored in suitable storage conditions to protect and secure the product and prevent damage or the deterioration of it. These conditions should have been identified early in the design life cycle.

5.5.1.2.3 Prepare the Product for Transition

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

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

5.5.1.2.4 Transition the Product

The end product is then transitioned (i.e., moved, transported, or shipped) with required documentation to the customer based on the type of transition required, e.g., to the next higher level item in the product hierarchy (often called the Product Breakdown Structure (PBS)) for product integration or to the end user. Documentation may include operations manuals, installation instructions, and other information.

The end product is finally installed into the next higher assembly or into the customer/user site using the preapproved installation procedures.

Confirm Ready to Support

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

5.5.1.2.5 Capture Product Transition Work Products

Other work products generated during the transition process are captured and archived as appropriate. These may include site plans, special handling procedures, training, certifications, videos, inspections, or other products from these activities

5.5.1.3 Outputs

- **Delivered end product with applicable documentation:** This may take one of two forms:
    1. **Delivered end product for integration to next level up in system structure:** This includes the appropriate documentation. The form of the end product and applicable documentation are a function of the life-cycle phase and the placement within the system structure. (The form of the end product could be hardware, software, model, prototype, first article for test, or single operational article or multiple production articles.) Documentation includes applicable draft installation, operation, user, maintenance, or training manuals; applicable baseline documents (configuration baseline, specifications, and stakeholder expectations); and test results that reflect completion of verification and validation of the end product.
    2. **Delivered operational end product for end users:** The appropriate documentation is to accompany the delivered end product as well as the operational end product appropriately packaged. Documentation includes applicable final installation, operation, user, maintenance, or training manuals; applicable baseline documents (configuration baseline,

specifications, stakeholder expectations); and test results that reflect completion of verification and validation of the end product. If the end user will perform end product validation, sufficient documentation to support end user validation activities is delivered with the end product.

- **Work products from transition activities to technical data management:** Work products could include the transition plan, site surveys, measures, training modules, procedures, decisions, lessons learned, corrective actions, etc.
- **Realized enabling end products to appropriate life-cycle support organization:** Some of the enabling products that were developed during the various phases could include fabrication or integration specialized machines; tools; jigs; fabrication processes and manuals; integration processes and manuals; specialized inspection, analysis, demonstration, or test equipment; tools; test stands; specialized packaging materials and containers; handling equipment; storage-site environments; shipping or transportation vehicles or equipment; specialized courseware; instructional site environments; and delivery of the training instruction. For the later life-cycle phases, enabling products that are to be delivered may include specialized mission control equipment; data collection equipment; data analysis equipment; operations manuals; specialized maintenance equipment, tools, manuals, and spare parts; specialized recovery equipment; disposal equipment; and readying recovery or disposal site environments.

The process is complete when the following activities have been accomplished:

- For deliveries to the integration path, the end product is delivered to intended usage sites in a condition suitable for integration with other end products or composites of end products. Procedures, decisions, assumptions, anomalies, corrective actions, lessons learned, etc., resulting from transition for integration are recorded.
- For delivery to the end user path, the end products are installed at the appropriate sites; appropriate acceptance and certification activities are completed; training of users, operators, maintainers, and other necessary personnel is completed; and delivery is closed out with appropriate acceptance documentation.
- Any realized enabling end products are also delivered as appropriate including procedures, decisions, assumptions, anomalies, corrective actions, lessons learned, etc., resulting from transition-enabling products.

5.5.2 Product Transition Guidance

5.5.2.1 Additional Product Transition Considerations

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

- **Transportability Requirements:** If applicable, requirements in this section define the required configuration of the system of interest for transport. Further, this section details the external systems (and the interfaces to those systems) required for transport of the system of interest.

- **Environmental Requirements:** Requirements in this section define the environmental conditions required for the system of interest during transition (including storage and transportation).
- **Maintainability Requirements:** Requirements in this section detail how frequently, by whom, and by what means the system of interest requires maintenance (also any "care and feeding" if required).
- **Safety Requirements:** Requirements in this section define the life-cycle safety requirements for the system of interest and associated equipment, facilities, and personnel.
- **Security Requirements:** This section defines the Information Technology (IT) requirements, Federal and international export and security requirements, and physical security requirements for the system of interest.
- **Programmatic Requirements:** Requirements in this section define cost and schedule requirements.

5.5.2.2 After Product Transition to the End User—What Next?

As mentioned in chapter 2.0, there is a relationship between the SE engine and the activities performed after the product is transitioned to the end user. As shown in Figure 2.3-8, after the final deployment to the end user, the end product is operated, managed, and maintained through sustaining engineering functions. The technical management processes described in Section 6.0 are used during these activities. If at any time a new capability, upgrade, or enabling product is needed, the developmental processes of the engine are reengaged. When the end product's use is completed, the plans developed early in the life cycle to decommission, dispose, retire, or phase out the product are enacted. Also refer to Section 7.1 for special integration considerations when components of a larger program or system are obtained through a diversity of acquisition contract mechanisms.

6.0 Crosscutting Technical Management

This chapter describes the activities in the technical management processes listed in the systems engineering engine (Figure 2.1-1). The processes described in Chapters 4 and 5 are performed through the design and realization phases of a product. These processes can occur throughout the product lifecycle, from concept through disposal. They may occur simultaneously with any of the other processes. The chapter is separated into sections corresponding to the technical management processes 10 through 17 listed in Figure 2.1-1. Each technical management process is discussed in terms of the inputs, the activities, and the outputs. Additional guidance is provided using examples that are relevant to NASA projects.

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

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

Crosscutting Technical Management Keys

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

6.1 Technical Planning

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

This effort starts with the technical team conducting extensive planning early in Pre-Phase A. With this early planning, technical team members will understand the roles and responsibilities of each team member, and can establish cost and schedule goals and objectives. From this effort, the Systems Engineering Management Plan (SEMP) and other technical plans are developed and baselined. Once the SEMP and technical plans have been established, they should be synchronized with the project master plans and schedule. In addition, the plans for establishing and executing all technical contracting efforts are identified.

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

6.1.1 Process Description

Figure 6.1-1 provides a typical flow diagram for the Technical Planning Process and identifies typical inputs, outputs, and activities to consider in addressing technical planning.

6.1.1.1 Inputs

Input to the Technical Planning Process comes from both the project management and technical teams as outputs from the other common technical processes. Initial planning utilizing external inputs from the project to determine the general scope and framework of the technical effort will be based on known technical and programmatic requirements, constraints, policies, and processes. Throughout the project's life cycle, the technical team continually incorporates results into the technical planning strategy and documentation and any internal changes based on decisions and assessments generated by the other processes of the SE engine or from requirements and constraints mandated by the project.

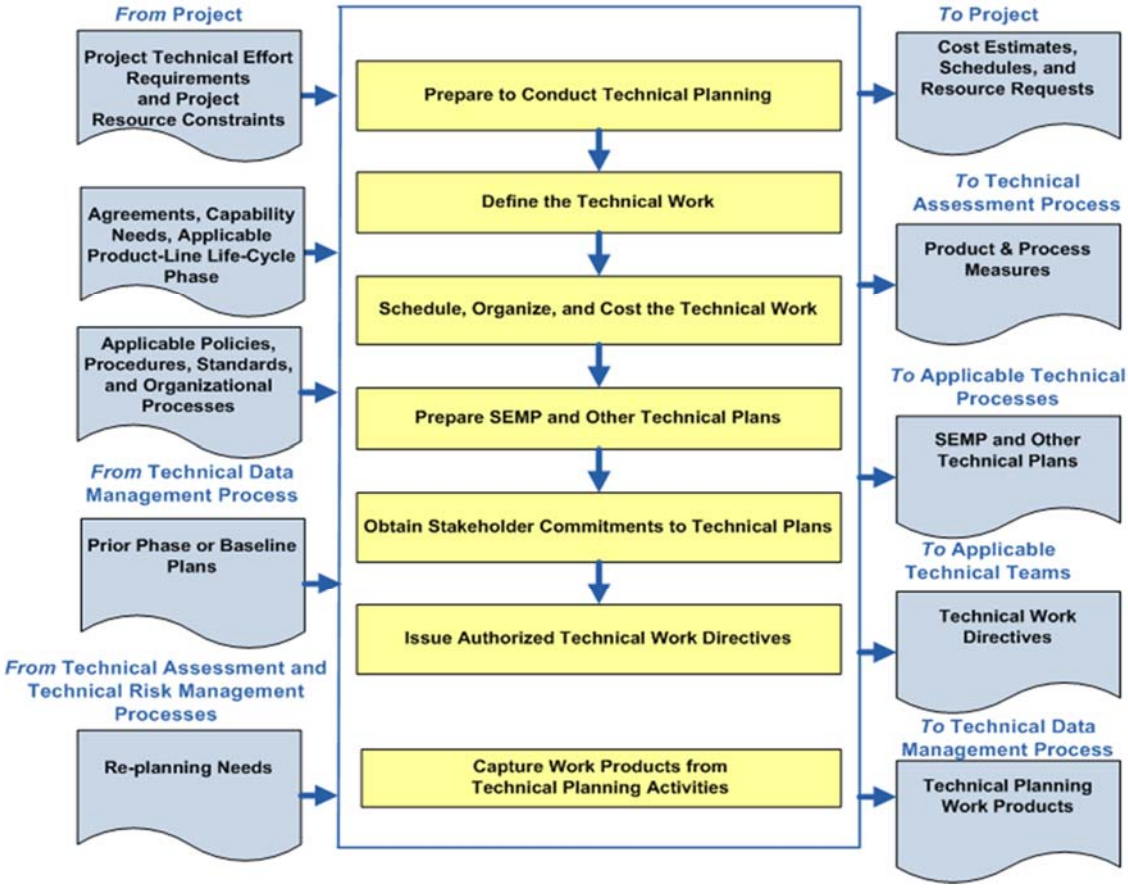


Figure 6.1-1 Technical Planning Process

- **Project Technical Effort Requirements and Project Resource Constraints:** The program/project plan provides the project's top-level technical requirements, the available budget allocated to the program/project from the program, and the desired schedule to support overall program needs. Although the budget and schedule allocated to the program/project serve as constraints, the technical team generates a technical cost estimate and schedule based on the actual work required to satisfy the technical requirements. Discrepancies between the allocated budget and schedule and the technical team's actual cost estimate and schedule should be reconciled continuously throughout the life cycle.
- **Agreements, Capability Needs, Applicable Product Life-Cycle Phase:** The program/project plan also defines the applicable life-cycle phases and milestones, as well as any internal and external agreements or capability needs required for successful execution. The life-cycle phases and programmatic milestones provide the general framework for establishing the technical planning effort and for generating the detailed technical activities and products required to meet the overall milestones in each of the life-cycle phases.
- **Applicable Policies, Procedures, Standards, and Organizational Processes:** The program/project plan includes all programmatic policies, procedures, standards, and organizational processes that should be adhered to during execution of the technical effort. The technical team should develop a technical approach that ensures the program/project requirements are satisfied and that any technical procedures, processes, and standards to be

used in developing the intermediate and final products comply with the policies and processes mandated in the program/project plan.

- **Prior Phase or Baseline Plans:** The latest technical plans (either baselined or from the previous life-cycle phase) from the Data Management or Configuration Management Processes should be used in updating the technical planning for the upcoming life-cycle phase.
- **Replanning Needs:** Technical planning updates may be required based on results from technical reviews conducted in the Technical Assessment Process, issues identified during the Technical Risk Management Process, or from decisions made during the Decision Analysis Process.

6.1.1.2 Process Activities

Technical planning as it relates to systems engineering at NASA is intended to define how the project will be organized, structured, and conducted and to identify, define, and plan how the 17 common technical processes in NPR 7123.1, NASA Systems Engineering Processes and Requirements will be applied in each life-cycle phase for all levels of the product hierarchy (see Section 6.1.2.1.) within the system structure to meet product life-cycle phase success criteria. A key document capturing and updating the details from the technical planning process is the SEMP.

The SEMP is a subordinate document to the project plan. The project plan defines how the project will be managed to achieve its goals and objectives within defined programmatic constraints. The SEMP defines for all project participants how the project will be technically managed within the constraints established by the project. The SEMP also communicates how the systems engineering management techniques will be applied throughout all phases of the project life cycle.

Technical planning should be tightly integrated with the Technical Risk Management Process (see Section 6.4) and the Technical Assessment Process (see Section 6.7) to ensure corrective action for future activities will be incorporated based on current issues identified within the project.

Technical planning, as opposed to program or project planning, addresses the scope of the technical effort required to develop the system products. While the project manager concentrates on managing the overall project life cycle, the technical team, led by the systems engineer, concentrates on managing the technical aspects of the project. The technical team identifies, defines, and develops plans for performing decomposition, definition, integration, verification, and validation of the system while orchestrating and incorporating the appropriate concurrent and crosscutting engineering. Additional planning includes defining and planning for the appropriate technical reviews, audits, assessments, and status reports and determining crosscutting engineering discipline and/or design verification requirements.

This section describes how to perform the activities contained in the Technical Planning Process shown in Figure 6.1-1. The initial technical planning at the beginning of the project establishes the technical team members; their roles and responsibilities; and the tools, processes, and

resources that will be utilized in executing the technical effort. In addition, the expected activities that the technical team will perform and the products it will produce are identified, defined, and scheduled. Technical planning continues to evolve as actual data from completed tasks are received and details of near-term and future activities are known.

6.1.1.2.1 Technical Planning Preparation

For technical planning to be conducted properly, the processes and procedures that are needed to conduct technical planning should be identified, defined, and communicated. As participants are identified, their roles and responsibilities and any training and/or certification activities should be clearly defined and communicated.

Team Selection

Teams engaged in the early part of the technical planning process need to identify the required skill mix for technical teams that will develop and produce a product. Typically, a technical team consists of a mix of both subsystem and discipline engineers. Considering a spacecraft example, subsystem engineers normally have cognizance over development of a particular subsystem (e.g., mechanical, power, etc.), whereas discipline engineers normally provide specific analyses (e.g., flight dynamics, radiation, etc.). The availability of appropriately skilled personnel also needs to be considered.

To an extent, determining the skill mix required for developing any particular product is a subjective process. Due to this, the skill mix is normally determined in consultation with people experienced in leading design teams for a particular mission or technical application. Some of the subjective considerations involved include the product and its requirements, the mission class, and the project phase.

Continuing with a spacecraft example, most teams typically share a common core of required skills, such as subsystem engineering for mechanical, thermal, power, etc. However, the particular requirements of a spacecraft and mission can cause the skill mix to vary. For example, as opposed to robotic space missions, human-rated systems typically add the need for human systems discipline engineering and environmental control and life support subsystem engineering. As opposed to near Earth space missions, deep space missions may add the need for safety and planetary protection discipline engineering specific to contamination of the Earth or remote solar system bodies. And, as opposed to teams designing spacecraft instruments that operate at moderate temperatures, teams designing spacecraft instruments that operate at cryogenic temperatures will need cryogenics subsystem support.

Mission class and project phase may also influence the required team skill mix. For example, with respect to mission class, certain discipline analyses needed for Class A and B missions may not be required for Class D (or lower) missions. And with respect to project phase, some design and analyses may be performed by a single general discipline in Pre-Phase A and Phase A, whereas the need to conduct design and analyses in more detail in Phases B and C may indicate the need for multiple specialized subsystem design and discipline engineering skills.

An example skill mix for a Pre-Phase A technical team tasked to design a cryogenic interferometer space observatory is shown in Table 6.1-1 for purposes of illustration. For

simplicity, analysis and technology development is assumed to be included in the subsystem or discipline shown. For example, this means "mechanical subsystem" includes both loads and dynamics analysis and mechanical technology development.

Table 6.1-1 Example Engineering Team Disciplines in Pre-Phase A for Robotic Infrared Observatory

Systems Engineering
-- Mission Systems Engineer
-- Instrument Systems Engineer
Spacecraft Bus, Flight Dynamics, Launch Vehicle Interface, Ground System Interface Subteam
-- Flight Dynamics Analysis
-- Mission Operations (includes ConOps, & interfaces with ground station, mission ops center, science ops center)
-- Bus Mechanical Subsystem (includes mechanisms)
-- Bus Power Subsystem (includes electrical harness)
-- Bus Thermal Subsystem
-- Bus Propulsion Subsystem
-- Bus Attitude Control and Determination Subsystem
-- Bus Avionics Subsystem
-- Bus Communications Subsystem
-- Bus Flight Software Subsystem
-- Integration & Test (bus, observatory)
-- Launch Vehicle Integration
-- Radiation Analysis
-- Orbital Debris/End of Mission Planning Analysis
-- System Reliability/Fault Tolerance Analysis (includes analysis of instrument)
Instrument Subteam
-- Mechanical Subsystem
-- Mechanisms Subsystem
-- Thermal Subsystem
-- Cryogenics Subsystem
-- Avionics Subsystem (incl. Electrical Harness)
-- Mechanism Drive Electronics Subsystem
-- Detector Subsystem
-- Optics Subsystem
-- Control Subsystem
-- Metrology Subsystem
-- Flight Software Subsystem
-- Integration & Test
-- Stray Light/Radiometry Analysis

-- Other Specialty Disciplines (e.g., Contamination Analysis) as needed

Once the processes, people, and roles and responsibilities are in place, a planning strategy may be formulated for the technical effort. A basic technical planning strategy should address the following:

- The communication strategy within the technical team and for up and out communications;
- Identification and tailoring of NASA procedural requirements that apply to each level of the PBS structure;
- The level of planning documentation required for the SEMP and all other technical planning documents;
- Identifying and collecting input documentation;
- The sequence of technical work to be conducted, including inputs and outputs;
- The deliverable products from the technical work;
- How to capture the work products of technical activities;
- How technical risks will be identified and managed;
- The tools, methods, and training needed to conduct the technical effort;
- The involvement of stakeholders in each facet of the technical effort;
- How the NASA technical team will be involved with the technical efforts of external contractors;
- The entry and success criteria for milestones, such as technical reviews and life-cycle phases;
- The identification, definition, and control of internal and external interfaces;
- The identification and incorporation of relevant lessons learned into the technical planning;
- The team's approach to capturing lessons learned during the project and how those lessons will be recorded;
- The approach for technology development and how the resulting technology will be incorporated into the project;
- The identification and definition of the technical metrics for measuring and tracking progress to the realized product;
- The criteria for make, buy, or reuse decisions and incorporation criteria for Commercial Off-the-Shelf (COTS) software and hardware;
- The plan to identify and mitigate off-nominal performance;
- The "how-tos" for contingency planning and replanning;
- The plan for status assessment and reporting;
- The approach to decision analysis, including materials needed, skills required, and expectations in terms of accuracy; and

- The plan for managing the human element in the technical activities and product.

By addressing these items and others unique to the project, the technical team will have a basis for understanding and defining the scope of the technical effort, including the deliverable products that the overall technical effort will produce, the schedule and key milestones for the project that the technical team should support, and the resources required by the technical team to perform the work.

A key element in defining the technical planning effort is understanding the amount of work associated with performing the identified activities. Once the scope of the technical effort begins to coalesce, the technical team may begin to define specific planning activities and to estimate the amount of effort and resources required to perform each task. Historically, many projects have underestimated the resources required to perform proper planning activities and have been forced into a position of continuous crisis management in order to keep up with changes in the project.

Identifying Facilities

The planning process also includes identifying the required facilities, laboratories, test beds, and instrumentation needed to build, test, launch, and operate a variety of commercial and Government products. A sample list of the kinds of facilities that might be considered when planning is illustrated in Table 6.1-2.

Table 6.1-2 Examples of Types of Facilities to Consider during Planning

Communications & Tracking Labs	Models & Simulation Labs	Thermal Chambers
Power Systems Labs	Prototype Development Shops	Vibration Labs
Propulsion Test Stands	Calibration Labs	Radiation Labs
Mechanical/Structures Labs	Biological Labs	Animal Care Labs
Instrumentation Labs	Space Materials Curation Labs	Flight Hardware Storage Areas
Human Systems Labs	Electromagnetic Effects Labs	Design Visualization
Guidance and Navigation Labs	Materials Labs	Wiring Shops
Robotics Labs	Vacuum Chambers	NDE Labs
Software Development Environment	Mission Control Center	Logistics Warehouse
Meeting rooms	Training Facilities	Conference facilities
Education/Outreach centers	Server farms	Project documentation centers

6.1.1.2.2 Define the Technical Work

The technical effort should be defined commensurate with the level of detail needed for the life cycle phase. When performing the technical planning, realistic values for cost, schedule, and labor resources should be used. Whether extrapolated from historical databases or from interactive planning sessions with the project and stakeholders, realistic values should be calculated and provided to the project team. Contingency should be included in any estimate and

should be based on the complexity and criticality of the effort. Contingency planning should be conducted. The following are examples of contingency planning:

- Additional, unplanned-for software engineering resources are typically needed during hardware and systems development and testing to aid in troubleshooting errors/anomalies. Frequently, software engineers are called upon to help troubleshoot problems and pinpoint the source of errors in hardware and systems development and testing (e.g., for writing additional test drivers to debug hardware problems). Additional software staff should be planned into the project contingencies to accommodate inevitable component and system debugging and avoid cost and schedule overruns.
- Hardware-In-the-Loop (HWIL) should be accounted for in the technical planning contingencies. HWIL testing is typically accomplished as a debugging exercise where the hardware and software are brought together for the first time in the costly environment of HWIL. If upfront work is not done to understand the messages and errors arising during this test, additional time in the HWIL facility may result in significant cost and schedule impacts. Impacts may be mitigated through upfront planning, such as making appropriate debugging software available to the technical team prior to the test, etc.
- Similarly, Human-In-The-Loop (HITL) evaluations identify contingency operational issues. HITL investigations are particularly critical early in the design process to expose, identify, and cost-effectively correct operational issues—nominal, maintenance, repair, off-nominal, training, etc.—in the required human interactions with the planned design. HITL testing should also be approached as a debugging exercise where hardware, software, and human elements interact and their performance is evaluated. If operational design and/or performance issues are not identified early, the cost of late design changes will be significant.

6.1.1.2.3 Schedule, Organize, and Budget the Technical Effort

Once the technical team has defined the technical work to be done, efforts can focus on producing a schedule and cost estimate for the technical portion of the project. The technical team should organize the technical tasks according to the project WBS in a logical sequence of events, taking into consideration the major project milestones, phasing of available funding, and timing of the availability of supporting resources.

Scheduling

Products described in the WBS are the result of activities that take time to complete. These activities have time precedence relationships among them that may be used to create a network schedule explicitly defining the dependencies of each activity on other activities, the availability of resources, and the receipt of receivables from outside sources. Use of a scheduling tool may facilitate the development and maintenance of the schedule.

Scheduling is an essential component of planning and managing the activities of a project. The process of creating a network schedule provides a standard method for defining and communicating what needs to be done, how long it will take, and how each element of the project WBS might affect other elements. A complete network schedule may be used to calculate how long it will take to complete a project; which activities determine that duration (i.e., critical

path activities); and how much spare time (i.e., float) exists for all the other activities of the project.

"Critical path" is the sequence of dependent tasks that determines the longest duration of time needed to complete the project. These tasks drive the schedule and continually change, so they should be updated. The critical path may encompass only one task or a series of interrelated tasks. It is important to identify the critical path and the resources needed to complete the critical tasks along the path if the project is to be completed on time and within its resources. As the project progresses, the critical path will change as the critical tasks are completed or as other tasks are delayed. This evolving critical path with its identified tasks needs to be carefully monitored during the progression of the project.

Network scheduling systems help managers accurately assess the impact of both technical and resource changes on the cost and schedule of a project. Cost and technical problems often show up first as schedule problems. Understanding the project's schedule is a prerequisite for determining an accurate project budget and for tracking performance and progress. Because network schedules show how each activity affects other activities, they assist in assessing and predicting the consequences of schedule slips or accelerations of an activity on the entire project.

For additional information on scheduling, refer to *NASA/SP-2010-3403, NASA Schedule Management Handbook*

Network Schedule Data and Graphical Formats

Network schedule data consists of:

- Activities and associated tasks;
- Dependencies among activities (e.g., where an activity depends upon another activity for a receivable);
- Products or milestones that occur as a result of one or more activities; and
- Duration of each activity.

A network schedule contains all four of the above data items. When creating a network schedule, creating graphical formats of these data elements may be a useful first step in planning and organizing schedule data.

Workflow Diagrams

A workflow diagram is a graphical display of the first three data items. Two general types of graphical formats are used as shown in Figure 6.1-2. One places activities on arrows with products and dependencies at the beginning and end of the arrow. This is the typical format of the Program Evaluation and Review Technique (PERT) chart.

The second format, called precedence diagrams, uses boxes to represent activities; dependencies are then shown by arrows. The precedence diagram format allows for simple depiction of the following logical relationships:

- Activity B begins when Activity A begins (start-start).
- Activity B begins only after Activity A ends (finish-start).

- Activity B ends when Activity A ends (finish-finish).

Each of these three activity relationships may be modified by attaching a lag (+ or –) to the relationship, as shown in figure 6.1-2. It is possible to summarize a number of low-level activities in a precedence diagram with a single activity by taking the initial low-level activity and attaching a summary activity to it using the start-start relationship described above. The summary activity is then attached to the final low-level activity using the finish-start relationship. The most common relationship used in precedence diagrams is the finish-start one. The activity-on-arrow format can represent the identical time-precedence logic as a precedence diagram by creating artificial events and activities as needed.

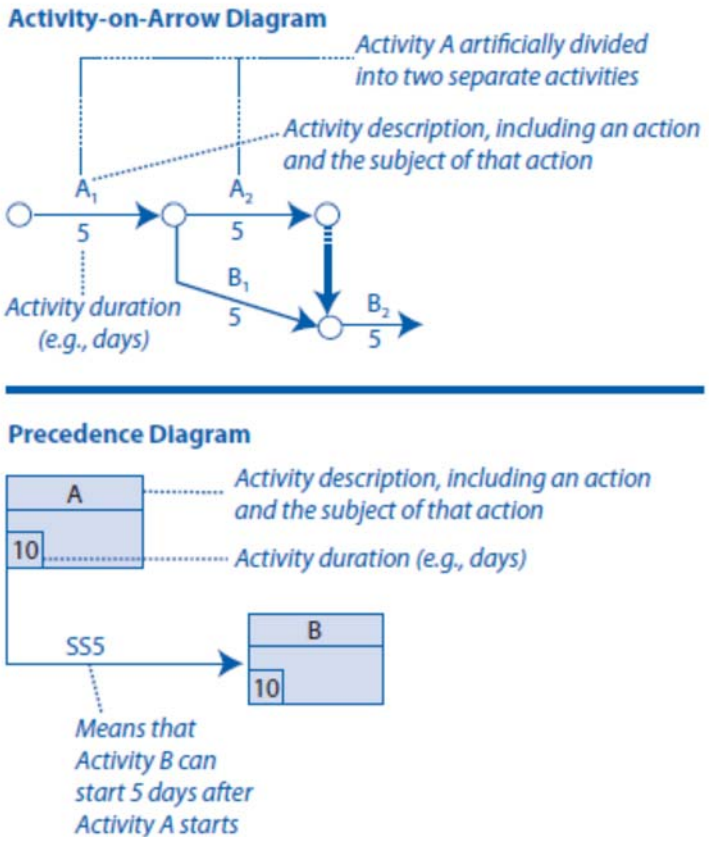


Figure 6.1-2 Activity-on-Arrow and Precedence Diagrams for Network Schedules

Establishing a Network Schedule

Scheduling begins with project-level schedule objectives for delivering the products described in the upper levels of the WBS. To develop network schedules that are consistent with the project's objectives, the following six steps are applied to each element at the lowest available level of the WBS.

Step 1: Identify activities and dependencies needed to complete each WBS element. Enough activities should be identified to show exact schedule dependencies between activities and other WBS elements. This first step is most easily accomplished by:

a. Ensuring that the WBS elements are extended downward to describe all significant products including documents, reports, and hardware and software items.
b. For each product, listing the steps required for its generation and drawing the process as a workflow diagram.
c. Indicating the dependencies among the products, and any integration and verification steps within the work package.

Step 2: Identify and negotiate external dependencies. External dependencies are any receivables from outside of, and any deliverables that go outside of the WBS element. Negotiations should occur to ensure that there is agreement with respect to the content, format, and labeling of products that move across WBS elements so that lower level schedules can be integrated.

Step 3: Estimate durations of all activities. Assumptions behind these estimates (hours required to complete the work, available workforce, availability of facilities, etc.) form a basis of estimate and should be written down for future reference.

Step 4: Enter the data for each WBS element into a scheduling program to obtain a network schedule and an estimate of the critical path for that element. It is not unusual at this point for some iteration of steps 1 to 4 to obtain a satisfactory schedule. Reserve is often added to critical-path activities to ensure that schedule commitments can be met within targeted risk levels.

Step 5: Integrate schedules of lower-level WBS elements so that all dependencies among elements are correctly included in a project network. It is important to include the impact of holidays, weekends, etc., at this point. The critical path for the project is discovered at this step in the process.

Step 6: Review the workforce level and funding profile over time and make a final set of adjustments to logic and durations so that workforce levels and funding levels are within project constraints. Adjustments to the logic and the durations of activities may be needed to converge to the schedule targets established at the project level. Adjustments may include adding more activities to some WBS elements, deleting redundant activities, increasing the workforce for some activities that are on the critical path, or finding ways to do more activities in parallel, rather than in series.

Again, it is good practice to have some schedule reserve, or float, as part of a risk mitigation strategy. The product of these last steps is a feasible baseline schedule for each WBS element that is consistent with the activities of all other WBS elements. The sum of all of these schedules should be consistent with both the technical scope and the schedule goals of the project. There should be enough float in this integrated master schedule so that schedule and associated cost risk are acceptable to the project and to the project's customer. Even when this is done, time estimates for many WBS elements will have been underestimated or work on some WBS elements will not start as early as had been originally assumed due to late arrival of receivables. Consequently, replanning is almost always needed to meet the project's goals.

Reporting Techniques

Summary data about a schedule is usually described in charts. A Gantt chart is a bar chart that depicts a project schedule using start and finish dates of the appropriate product elements tied to the project WBS of a project. Some Gantt charts also show the dependency (i.e., precedence and critical path) relationships among activities and also current status. A good example of a Gantt chart is shown in Figure 6.1-3. (See box on Gantt chart features.)

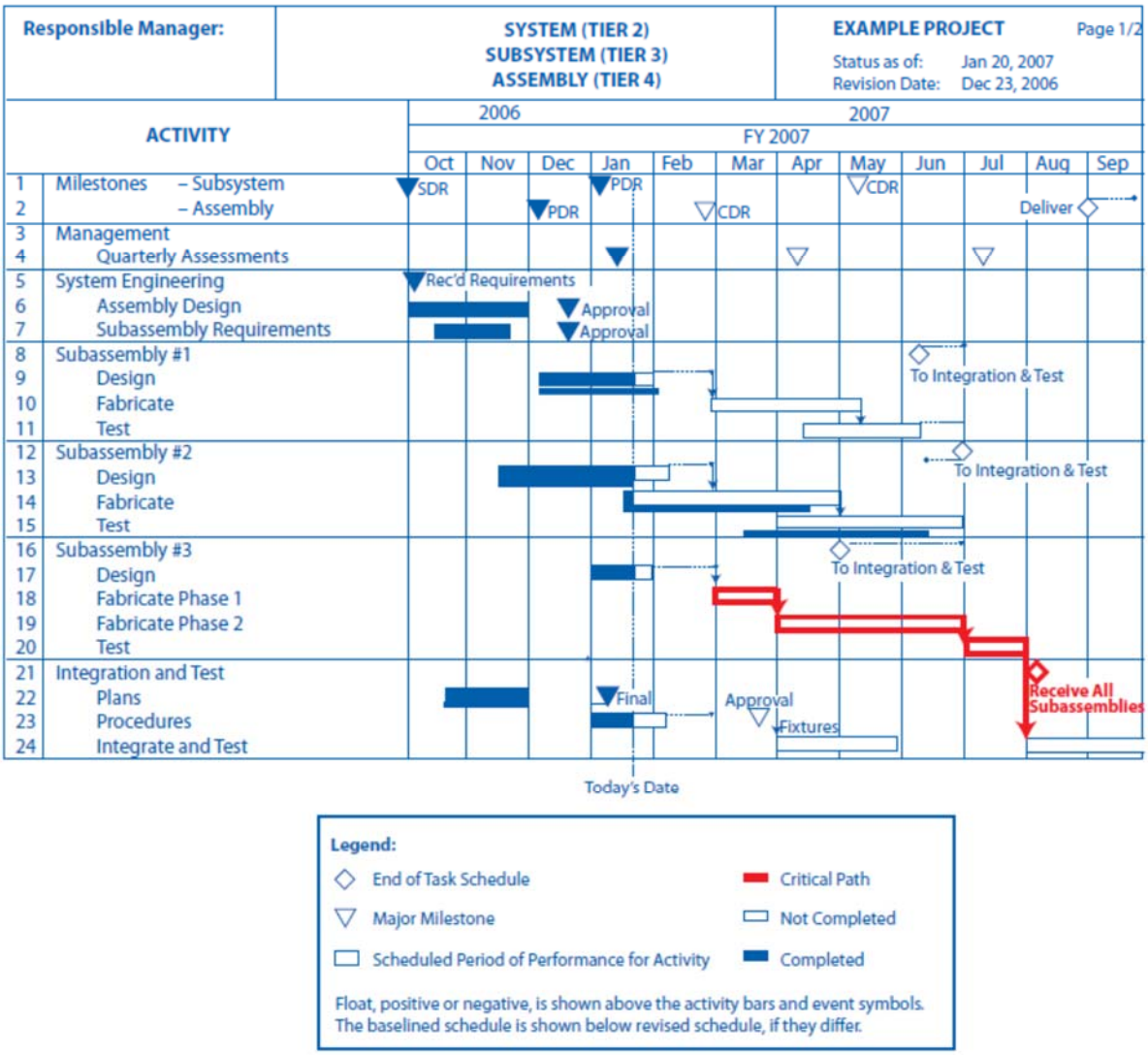


Figure 6.1-3 Gantt Chart

Another type of output format is a table that shows the float and recent changes in float of key activities. For example, a project manager may wish to know precisely how much schedule reserve has been consumed by critical path activities and whether reserves are being consumed or are being preserved in the latest reporting period. Such a table provides information on the rate of change of schedule reserve.

Resource Leveling

Good scheduling systems provide capabilities to show resource requirements over time and to make adjustments so that the schedule is feasible with respect to resource constraints over time. Resources may include workforce level, funding profiles, important facilities, etc. The objective is to move the start dates of tasks that have float to points where the resource profile is feasible. If that is not sufficient, then the assumed task durations for resource-intensive activities should be reexamined and, accordingly, the resource levels changed.

Gantt Chart Features

The Gantt chart shown in figure 6.1-3 illustrates the following desirable features:

- A heading that describes the WBS element, identifies the responsible manager, and provides the date of the baseline used and the date that status was reported.
- A milestone section in the main body (lines 1 and 2).
- An activity section in the main body. Activity data shown includes:
  - WBS elements (lines 3, 5, 8, 12, 16, and 21);
  - Activities (indented from WBS elements);
  - Current plan (shown as thick bars);
  - Baseline plan (same as current plan, or if different, represented by thin bars under the thick bars);
  - Slack for each activity (dotted horizontal line before the milestone on line 12);
  - Schedule slips from the baseline (dotted horizontal lines after the current plan bars);
  - The critical path is shown encompassing lines 18 through 21 and impacting line 24; and
  - Status line (dotted vertical line from top to bottom of the main body of the chart) at the date the status was reported.
- A legend explaining the symbols in the chart.

This Gantt chart shows only 24 lines, which is a summary of the activities currently being worked for this WBS element. It is appropriate to tailor the amount of detail reported to those items most pertinent at the time of status reporting.

Budgeting

Budgeting and resource planning involve establishing a reasonable project baseline budget and the capability to analyze changes to that baseline resulting from technical and/or schedule changes. The project's WBS, baseline schedule, and budget should be viewed as mutually dependent, reflecting the technical content, time, and cost of meeting the project's goals and objectives. The budgeting process needs to take into account whether a fixed cost cap or fixed cost profile exists. When no such cap or profile exists, a baseline budget is developed from the WBS and network schedule. This specifically involves combining the project's workforce and other resource needs with the appropriate workforce rates and other financial and programmatic factors to obtain cost element estimates. These elements of cost include

- Direct labor costs,
- Overhead costs,
- Other direct costs (travel, data processing, etc.),
- Subcontract costs,

- Material costs,
- Equipment costs,
- General and administrative costs,
- Cost of money (i.e., interest payments, if applicable),
- Fee (if applicable), and
- Contingency (Unallocated Future Expenses (UFE)).

When there is a fixed cost cap or a fixed cost profile, there are additional logic gates that should be satisfied before completing the budgeting and planning process. A determination needs to be made whether the WBS and network schedule are feasible with respect to mandated cost caps and/or cost profiles. If not, it will be necessary to consider stretching out a project (usually at an increase in the total cost) or descoping the project's goals and objectives, requirements, design, and/or implementation approach. Cost-reduction activities and affordability events supported by subject matter experts should be conducted with the goal of reducing total life cycle cost.

If a fixed cost cap or fixed cost profile exists, it is especially important to control costs after they have been baselined. An important aspect of cost control is project cost and schedule status reporting and assessment, methods for which are discussed in Section 6.7. Another is cost and schedule risk planning, such as developing risk avoidance and workaround strategies. At the project level, budgeting and resource planning should ensure that an adequate level of contingency funds is included to deal with unforeseen events.

The maturity of the Life-Cycle Cost Estimate (LCCE) should progress as follows:

- Pre-Phase A: Initial LCCE (70 percent confidence level; however, much uncertainty is expected)
- Phase A: LCCE cost and schedule range at KDP 0/KDP B
- Phase B: Approve LCCE (70 percent joint cost and schedule confidence level at KDP I/KDP C)
- Phase C, D, and E: projects with LCCE > $20M report variances to LCCE baseline using Earned Value Management (EVM) and LCCE updates

Credibility of the cost estimate is suspect if:

- WBS cost estimates are expressed only in dollars with no other identifiable units, indicating that requirements are not sufficiently defined for processes and resources to be identified.
- The basis of estimates does not contain sufficient detail for independent verification that work scope and estimated cost (and schedule) are reasonable.
- Actual costs vary significantly from the LCCE.
- Work is performed that was not originally planned, causing cost or schedule variance.
- Schedule and cost earned value performance trends readily indicate unfavorable performance.

- Operations costs of the human element(s) required to operate and maintain the system are not included.

For additional information on cost estimating, refer to the *NASA Cost Estimating Handbook* and NPR 7120.5, NASA Space Flight Program and Project Management Requirements.

6.1.1.2.4 Prepare the SEMP and Other Technical Plans

Systems Engineering Management Plan

The SEMP is the primary, top-level technical management document for the project and is developed early in the Formulation Phase and updated throughout the project life cycle. The SEMP is driven by the type of project, the phase in the project life cycle, and the technical development risks and is written specifically for each project or project element. While the specific content of the SEMP is tailored to the project, the recommended content is discussed in appendix J. It is important to remember that the main value of the SEMP is in the work that goes into the planning.

The technical team, working under the overall project plan, develops and updates the SEMP as necessary. The technical team works with the project manager to review the content and obtain concurrence. This allows for thorough discussion and coordination of how the proposed technical activities would impact the programmatic, cost, and schedule aspects of the project. The SEMP provides the specifics of the technical effort and describes the technical processes that will be used, how the processes will be applied using appropriate activities, how the project will be organized to accomplish the activities, and the cost and schedule associated with accomplishing the activities.

The physical length of a SEMP is not what is important. This will vary from project to project. The plan needs to be adequate to address the specific technical needs of the project. It is a *living* document that is updated as often as necessary to incorporate new information as it becomes available and as the project develops through the Implementation Phase. The SEMP should not duplicate other project documents; however, the SEMP should reference and summarize the content of other technical plans.

The systems engineer and project manager should identify additional required technical plans based on the project scope and type. If plans are not included in the SEMP, they should be referenced and coordinated in the development of the SEMP. Other plans, such as system safety, probabilistic risk assessment, and an HSI Plan also need to be planned for and coordinated with the SEMP. If a technical plan is a stand-alone, it should be referenced in the SEMP. Depending on the size and complexity of the project, these may be separate plans or they may be included within the SEMP. Once identified, the plans can be developed, training on these plans established, and the plans implemented. Examples of technical plans in addition to the SEMP are listed in appendix K.

The SEMP should be developed during pre-formulation. In developing the SEMP, the technical approach to the project's life cycle is developed. This determines the project's length and cost. The development of the programmatic and technical management approaches requires that the key project personnel develop an understanding of the work to be performed and the

relationships among the various parts of that work. Refer to Sections 6.1.2.1 and 6.1.1.2 on WBSs and network scheduling, respectively. The SEMP then flows into the project plan to ensure the proper allocation of resources including cost, schedule, and personnel.

The SEMP's development requires contributions from knowledgeable programmatic and technical experts from all areas of the project that can significantly influence the project's outcome. The involvement of recognized experts is needed to establish a SEMP that is credible to the project manager and to secure the full commitment of the project team.

Role of the SEMP

The SEMP is the rule book that describes to all participants how the project will be technically managed. The NASA technical team on the project should have a SEMP to describe how it will conduct its technical management, and each contractor should have a SEMP to describe how it will manage in accordance with both its contract and NASA's technical management practices. Since the SEMP is unique to a project and contract, it should be updated for each significant programmatic change or it will become outmoded and unused and the project could slide into an uncontrolled state. The lead NASA field Center should have its SEMP developed before attempting to prepare an initial cost estimate since activities that incur cost, such as technical risk reduction and human element accounting, need to be identified and described beforehand. The contractor should have its SEMP developed during the proposal process (prior to costing and pricing) because the SEMP describes the technical content of the project, the potentially costly risk management activities, and the verification and validation techniques to be used, all of which should be included in the preparation of project cost estimates. The SEMPs from the supporting Centers should be developed along with the primary project SEMP. The project SEMP is the senior technical management document for the project; all other technical plans should comply with it. The SEMP should be comprehensive and describe how a fully integrated engineering effort will be managed and conducted.

Verification Plan

The verification plan is developed as part of the Technical Planning Process and is baselined at PDR. As the design matures throughout the life cycle, the plan is updated and refined as needed. The task of preparing the verification plan includes establishing the method of verification to be performed, dependent on the life-cycle phase; the position of the product in the system structure; the form of the product used; and the related costs of verification of individual specified requirements. The verification methods include analyses, inspection, demonstration, and test. In some cases, the complete verification of a given requirement might require more than one method. For example, to verify the performance of a product may require looking at many use cases. This might be accomplished by running a Monte Carlo simulation (analysis) and also running actual tests on a few of the key cases. The verification plan, typically written at a detailed technical level, plays a pivotal role in bottom-up product realization.

Types of Testing

There are many different types of testing that can be used to verify an end product. The following examples are provided for consideration.

- Aerodynamic
- Acceptance
- Acoustic
- Burn-in
- Characterization
- Component
- Drop
- Electromagnetic Compatibility
- Electromagnetic Interference
- Environmental
- G-loading
- Go or No-Go
- High-/Low-Voltage Limits
- Human Factors Engineering/ Human-In-The-Loop Testing
- Integration
- Leak Rates
- Lifetime / Cycling
- Manufacturing/Random Defects
- Nominal
- Off-Nominal
- Operational
- Parametric
- Performance
- Pressure Cycling
- Pressure Limits
- Qualification Flow
- Structural Functional
- Security Checks
- System
- Thermal Cycling
- Thermal Limits
- Thermal Vacuum
- Vibration

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

- Simulated (algorithmic models, virtual reality simulator);
- Mockup (plywood, brassboard, breadboard);
- Concept description (paper report);
- Engineering unit (fully functional but may not be same form/fit);
- Prototype (form, fit, and function);
- Design verification test units (form, fit, and function is the same, but they may not have flight parts);
- Qualification units (identical to flight units but may be subjected to extreme environments); and
- Flight units (end product that is flown, including protoflight units).

Types of Hardware

- **Breadboard:** A low fidelity unit that demonstrates function only without considering form or fit in the case of hardware or platform in the case of software. It often uses commercial and/or ad hoc components and is not intended to provide definitive information regarding operational performance.
- **Brassboard:** A medium fidelity functional unit that typically tries to make use of as much operational hardware/software as possible and begins to address scaling issues associated with the operational system. It does not have the engineering pedigree in all aspects, but is structured to be able to operate in simulated operational environments in order to assess performance of critical functions.
- **Engineering Unit:** A high fidelity unit that demonstrates critical aspects of the engineering processes involved in the development of the operational unit. Engineering test units are intended to closely resemble the final product (hardware/software) to the maximum extent possible and are built and tested so as to establish confidence that the design will function in the expected environments. In some cases, the engineering unit will become the final product, assuming proper traceability has been exercised over the components and hardware handling.
- **Prototype Unit:** The prototype unit demonstrates form, fit, and function at a scale deemed to be representative of the final product operating in its operational environment. A subscale test article provides fidelity sufficient to permit validation of analytical models capable of predicting the behavior of full-scale systems in an operational environment.
- **Qualification Unit:** A unit that is the same as the flight unit (form, fit, function, components, etc.) that will be exposed to the extremes of the environmental criteria (thermal, vibration, etc.). The unit will typically not be flown due to these off-nominal stresses.
- **Protoflight Unit**: In projects that will not develop a qualification unit, the flight unit may be designated as a protoflight unit and a limited version of qualification test ranges will be applied. This unit will be flown.

Verification of the end product—that is, the official "run for the record" verification where the program/project takes credit for meeting a requirement—is usually performed on a qualification, protoflight, or flight unit to ensure its applicability to the flight system. However, with discussion and approval from the program/project and systems engineering teams, verification credit may be taken on lower fidelity units if they can be shown to be sufficiently like the flight units in the areas to be verified.

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

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

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

Procedures should be prepared to conduct verification based on the method (e.g., analysis, inspection, demonstration, or test) planned. These procedures are typically developed during the design phase of the project life cycle and matured as the design is matured. Operational use scenarios are thought through in order to explore all possible verification activities to be performed.

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

Methods of Verification

- **Analysis**: The use of mathematical modeling and analytical techniques to predict the suitability of a design to stakeholder expectations based on calculated data or data derived from lower system structure end product verifications. Analysis is generally used when a prototype; engineering model; or fabricated, assembled, and integrated product is not available. Analysis includes the use of modeling and simulation as analytical tools. A model is a mathematical representation of reality. A simulation is the manipulation of a model. Analysis can include verification by similarity of a heritage product.
- **Demonstration**: Showing that the use of an end product achieves the individual specified requirement. It is generally a basic confirmation of performance capability, differentiated from testing by the lack of detailed data gathering. Demonstrations can involve the use of physical models or mockups; for example, a requirement that all controls shall be reachable by the pilot could be verified by having a pilot perform flight-related tasks in a cockpit mockup or simulator. A demonstration could also be the actual operation of the end product by highly qualified personnel, such as test pilots, who perform a one-time event that demonstrates a capability to operate at extreme limits of system performance, an operation not normally expected from a representative operational pilot.
- **Inspection**: The visual examination of a realized end product. Inspection is generally used to verify physical design features or specific manufacturer identification. For example, if there is a requirement that the safety arming pin has a red flag with the words "Remove Before Flight" stenciled on the flag in black letters, a visual inspection of the arming pin flag can be used to determine if this requirement was met. Inspection can include inspection of drawings, documents, or other records.
- **Test**: The use of an end product to obtain detailed data needed to verify performance or provide sufficient information to verify performance through further analysis. Testing can be conducted on final end products, breadboards, brassboards, or prototypes. Testing produces data at discrete points for each specified requirement under controlled conditions and is the most resource-intensive verification technique. As the saying goes, "Test as you fly, and fly as you test." (See section 5.3.2.5.)

Outcomes of verification planning include the following:

- The verification method that is appropriate for showing or proving that the end product conforms to its specified requirements is selected.

- The product verification procedures are clearly defined based on: (1) the procedures for each method of verification selected, (2) the purpose and objective of each procedure, (3) any pre-verification and post-verification actions, and (4) the criteria for determining the success or failure of the procedure.
- The verification environment (e.g., facilities, equipment, tools, simulations, measuring devices, personnel, and climatic conditions) in which the verification procedures will be implemented is defined.

Note: Verification planning begins early in the project life cycle during the requirements development phase. (See section 4.2.) The verification approach to use should be included as part of requirements development to plan for future activities, to establish special requirements derived from identified verification-enabling products, and to ensure that the requirements are verifiable. Updates to verification planning continue throughout logical decomposition and design development, especially as design reviews and simulations shed light on items under consideration. (See section 6.1.)

- As appropriate, project risk items are updated based on approved verification strategies that cannot duplicate fully integrated test systems, configurations, and/or target operating environments. Rationales, trade space, optimization results, and implications of the approaches are documented in the new or revised risk statements as well as references to accommodate future design, test, and operational changes to the project baseline.

Validation Plan

The validation plan is one of the work products of the Technical Planning Process and is generated during the Design Solution Process to validate the end product against the baselined stakeholder expectations. This plan can take many forms. The plan describes the total Test and Evaluation (T&E) planning from development of lower-end through higher-end products in the system structure and through operational T&E into production and acceptance. It may combine the verification and validation plans into a single document. (See appendix I for a sample Verification and Validation Plan outline.)

The methods of validation include test, demonstration, inspection, and analysis. While the name of each method is the same as the name of the methods for verification, the purpose and intent as described above are quite different.

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

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

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

Procedures should be prepared to conduct validation based on the method planned; e.g., analysis, inspection, demonstration, or test). These procedures are typically developed during the design phase of the project life cycle and matured as the design is matured. Operational and use-case scenarios are thought through in order to explore all possible validation activities to be performed.

Methods of Validation

- **Analysis**: The use of mathematical modeling and analytical techniques to predict the suitability of a design to stakeholder expectations based on calculated data or data derived from lower system structure end product verifications. Analysis is generally used when a prototype; engineering model; or fabricated, assembled, and integrated product is not available. Analysis includes the use of modeling and simulation as analytical tools.
  A model is a mathematical representation of reality. A simulation is the manipulation of a model.
- **Demonstration**: Showing that the use of an end product achieves the stakeholder expectations as defined in the NGOs and the ConOps. It is generally a basic confirmation of behavioral capability, differentiated from testing by the lack of detailed data gathering. Demonstrations can involve the use of physical models or mockups; for example, an expectation that controls are readable by the pilot in low light conditions could be validated by having a pilot perform flight-related tasks in a cockpit mockup or simulator under those conditions.
- **Inspection**: The visual examination of a realized end product. Inspection is generally used to validate the presence of a physical design features or specific manufacturer identification. For example, if there is an expectation that the safety arming pin has a red flag with the words "Remove Before Flight" stenciled on the flag in black letters, a visual inspection of the arming pin flag can be used to determine if this expectation has been met.
- **Test**: The use of an end product to obtain detailed data needed to determine a behavior, or provide sufficient information to determine a behavior through further analysis. Testing can be conducted on final end products, breadboards, brassboards, or prototypes. Testing produces information at discrete points for each specified expectation under controlled conditions and is

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

Regardless of the source (buy, make, reuse, assemble and integrate) and the position in the system structure, all realized end products should be validated to demonstrate/confirm satisfaction of stakeholder expectations. Variations, anomalies, and out-of-compliance

conditions, where such have been detected, are documented along with the actions taken to resolve the discrepancies. Validation is typically carried out in the intended operational environment or a relevant environment under simulated or actual operational conditions, not necessarily under the tightly controlled conditions usually employed for the Product Verification Process.

Environments

- **Relevant Environment:** Not all systems, subsystems, and/or components need to be operated in the operational environment in order to satisfactorily address performance margin requirements or stakeholder expectations. Consequently, the relevant environment is the specific subset of the operational environment that is required to demonstrate critical "at risk" aspects of the final product performance in an operational environment.
- **Operational Environment:** The environment in which the final product will be operated. In the case of space flight hardware/software, it is space. In the case of ground-based or airborne systems that are not directed toward space flight, it is the environments defined by the scope of operations. For software, the environment is defined by the operational platform.

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

- Simulated (algorithmic models, virtual reality simulator);
- Mockup (plywood, brassboard, breadboard);
- Concept description (paper report);
- Engineering unit (functional but may not be same form/fit);
- Prototype (product with form, fit, and function);
- Design validation test units (form, fit, and function may be the same, but they may not have flight parts);
- Qualification unit (identical to flight unit but may be subjected to extreme environments); and
- Flight unit (end product that is flown).

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

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

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

Outcomes of validation planning include the following:

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

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

Integration Plans

The integration plan captures the system interactions within itself and with the environment by

- Identifying system interfaces,
- Establishing the system environments (In some programs, this may be defined in separate natural environment and induced environment documents, which should be referenced by the integration plan.),
- Identifying organizational relationship interactions,
- Defining the key system analysis to be conducted,
- Defining the test strategy (In some programs, a separate test plan may contain this and should be referenced by the integration plan.), and
- Defining the assembly and integration plans. (In some programs, this may be captured in a separate assembly and integration plan, which should be referenced by the integration plan.)

The closeout plan identifies key activities and system features (derived requirements) necessary to enable decommissioning and/or disposal of the system. The major steps in preparing closeout plans include the following:

- Identifying external dependencies and relationships;
- Identifying system interactions: subsystem and environmental;
- Defining organizational integration and information flows;
- Defining interfaces (thermal, fluids, electrical, mechanical, data, logical, human, etc.);
- Defining assembly, test, and maintenance functions and interfaces;
- Establishing the integration plan;
- Defining and documenting system closeout approaches, methods, and processes.

Conduct System Analysis

Analyzing the system design and system operation will establish the integrated system functionality and performance. The system interactions need to be accounted for in this analysis including subsystem interactions and environment interactions. These interactions can include external systems (e.g., Shuttle dependency on the International Space Station while docked) and incorporates the system internal and external interfaces. Derived requirements are generated as configuration options are down-selected and should be captured as part of the technical requirements discussed in Section 4.2.

- Define and execute system analysis (e.g., controllability, loads, mass margin, power, data bandwidth, flight measurements, developmental measurements, system exergy);
- Ensure interface compatibility at all defined interfaces;
- Identify fault management responses for unintended consequences of system interactions; and
- Ensure that operational system upgrades do not induce unwanted interactions.

Establish and Manage Organizational Interactions

The organizational structure affects how well integration activities are coordinated and how information is shared between design and/or operational disciplines. This structure should be tuned for the efficient system development and/or operation and information flow, which is managed to deal with potential organizational blind spots and information bottle necks.

Establish Test, Assembly, and Maintenance Functions

The derived requirements necessary to support test, assembly, and maintenance of the system need to be determined. These include test fixtures, ground support equipment, manufacturing tooling and fixtures, maintenance functions, and access. Plans for these needed functions should be documented in the integration plan or separate test plan, assembly and integration plan, and maintenance plan.

- Define system tests and test objectives for system and environmental interactions;
- Determine system tests necessary to anchor system models;
- Define system assembly sequences and accessibility; and
- Define system maintenance functions and accessibility.

6.1.1.2.5 Obtain Stakeholder Commitments to Technical Plans

Stakeholder Roles in Project Planning

To obtain commitments to the technical plans from the stakeholders, the technical team should ensure that the appropriate stakeholders, including subject domain experts, have a method to provide inputs and to review the project planning for implementation of stakeholder interests.

During the Formulation Phase, the roles of the stakeholders should be defined in the project plan and the SEMP. Review of these plans and the agreements from the stakeholders to the content of these plans constitutes buy-in from the stakeholders to the technical approach. It is essential to

identify the stakeholders and get their concurrence on the technical approach.

Later in the project life cycle, stakeholders may be responsible for delivering products to the project. Initial agreements regarding the responsibilities of the stakeholders are key to ensuring that the project technical team obtains the appropriate deliveries from stakeholders.

Stakeholder Involvement in Defining Requirements

The identification of stakeholders is one of the early steps in the systems engineering process. As the project progresses, stakeholder expectations are flowed down through the Logical Decomposition Process, and specific stakeholders are identified for all of the primary and derived requirements. A critical part of the stakeholders' involvement is in the definition of the technical requirements. As requirements and the ConOps are developed, the stakeholders will be required to agree to these products. Inadequate stakeholder involvement leads to inadequate requirements and a resultant product that does not meet the stakeholder expectations. Status on relevant stakeholder involvement should be tracked and corrective action taken if stakeholders are not participating as planned.

Stakeholder Agreements

Throughout the project life cycle, communication with the stakeholders and commitments from the stakeholders may be accomplished through the use of agreements. Organizations may use an Internal Task Agreement (ITA), a Memorandum Of Understanding (MOU), or other similar documentation to establish the relationship between the project and the stakeholder. These agreements are also used to document the customer and provider responsibilities for defining products to be delivered. These agreements should establish the Measures of Effectiveness (MOEs) or Measures of Performance (MOPs) that will be used to monitor the progress of activities. Reporting requirements and schedule requirements should be established in these agreements. Preparation of these agreements will ensure that the stakeholders' roles and responsibilities support the project goals and that the project has a method to address risks and issues as they are identified.

Stakeholder Support for Forums

During development of the project plan and the SEMP, forums are established to facilitate communication and document decisions during the life cycle of the project. These forums include meetings, working groups, decision panels, and control boards. Each of these forums should establish a charter to define the scope and authority of the forum and identify necessary voting or nonvoting participants. Ad hoc members may be identified when the expertise or input of specific stakeholders is needed when specific topics are addressed. It is important to ensure that stakeholders have been identified to support the forum.

6.1.1.2.6 Issue Technical Work Directives

The technical team provides technical work directives to Cost Account Managers (CAMs). This enables the CAMs to prepare detailed plans that are mutually consistent and collectively address all of the work to be performed. These plans include the detailed schedules and budgets for cost accounts that are needed for cost management and EVM.

Issuing technical work directives is an essential activity during Phase B of a project when a detailed planning baseline is required. If this activity is not implemented, then the CAMs are often left with insufficient guidance for detailed planning. The schedules and budgets that are needed for EVM will then be based on assumptions and local interpretations of project-level information. If this is the case, it is highly likely that substantial variances will occur between the baseline plan and the work performed. Providing technical work directives to CAMs produces a more organized technical team. This activity may be repeated when replanning occurs.

This "technical work directives" step produces: (1) planning directives to CAMs that result in (2) a consistent set of cost account plans. Where EVM is called for, it produces (3) an EVM planning baseline, including a Budgeted Cost of Work Scheduled (BCWS).

This activity is not limited to systems engineering. This is a normal part of project planning wherever there is a need for an accurate planning baseline.

Content of Technical Work Directives

The technical team provides technical directives to CAMs for every cost account within the SE element of the WBS. These directives may be in any format but should clearly communicate the following information for each account:

- Technical products expected;
- Documents and technical reporting requirements for each cost account;
- Critical events and specific products expected from a particular CAM in support of such an event (e.g., a particular CAM is expected to deliver a presentation on specific topics at the PDR);
- References to applicable requirements, policies, and standards;
- Identification of particular tools that should be used;
- Instructions on how the technical team wants to coordinate and review cost account plans before they go to project management; and
- Decisions that have been made on how work needs to be performed and who is to perform it.

Cost Account Plans

CAMs receive these technical directives along with the project planning guidelines and prepare cost account plans. These plans may be in any format and may have various names at different Centers, but minimally they will include the following:

- Scope of the cost account, which includes:
  - Technical products delivered;
  - Cost of the human element capital required to operate and maintain the system;
  - Other products developed that will be needed to complete deliverables (e.g., a

Configuration Management (CM) system may need to be developed in order to deliver the product of a "managed configuration");

- A brief description of the procedures that need to be followed to complete work on these products, such as:
  - Product X will be prepared in-house using the local procedure A, which is commonly used in organization ABC,
  - Product X will be verified/validated in the following manner…,
  - Product X will be delivered to the project in the following manner…,
  - Product X delivery will include the following reports (e.g., delivery of a CM system to the project would include regular reports on the status of the configuration, etc.),
  - Product Y will be procured in accordance with procurement procedure B.

- A schedule attached to this plan in a format compatible with project guidelines for schedules. This schedule would contain each of the procedures and deliverables mentioned above and provide additional information on the activity steps of each procedure.
- A budget attached to this plan in a system compatible with project guidelines for budgets. This budget would be consistent with the resources needed to accomplish the scheduled activities.
- Any necessary agreements and approvals.

Work Packages

If the project is going to use EVM, then the scope of a cost account needs to further identify a number of "work packages," which are units of work that can be scheduled and given cost estimates. Work packages should be based on completed products to the greatest extent possible but may also be based on completed procedures (e.g., completion of validation). Each work package will have its own schedule and a budget. The budget for this work package becomes part of the Budgeted Cost of Work Scheduled (BCWS) in the EVM system. When this unit of work is completed, the project's earned value will increase by this amount. There may be future work in this cost account that is not well enough defined to be described as a set of work packages. For example, launch operations will be supported by the technical team, but the details of what will be done often have not been worked out during Phase B. In this case, this future work is called a "planning package," which has a high-level schedule and an overall budget. When this work is understood better, the planning package is broken up into work packages so that the EVM system can continue to operate during launch operations.

Review and Approval of Cost Account Plans

Cost account plans should be reviewed and approved by the technical team and by the line manager of the cost account manager's home organization. Planning guidelines may identify additional review and approval requirements.

The planning process described above is not limited to systems engineering. This is the expected process for all elements of a flight project. One role that the systems engineer may have in planning is to verify that the scope of work described in cost account plans across the project is consistent with the project WBS dictionary, and that the WBS dictionary is consistent with the architecture of the project.

6.1.1.2.7 Capture Technical Planning Work Products

The work products from the Technical Planning Process should be managed using either the Technical Data Management Process or the Configuration Management Process as required. Some of the more important products of technical planning (i.e., the WBS, the SEMP, and the schedule, etc.) are kept under configuration control and captured using the CM process. The Technical Data Management Process is used to capture trade studies, cost estimates, technical analyses, reports, and other important documents not under formal configuration control. Work products, such as meeting minutes and correspondence (including e-mail) containing decisions or agreements with stakeholders also should be retained and stored in project files for later reference.

6.1.1.3 Outputs

Typical outputs from technical planning activities are:

- **Technical work cost estimates, schedules, and resource needs**: e.g., funds, workforce, facilities, and equipment (to the project) within the project resources;
- **Product and process measures:** Those needed to assess progress of the technical effort and the effectiveness of processes (to the Technical Assessment Process);
- **SEMP and other technical plans:** Technical planning strategy, WBS, SEMP, HSI Plan, V&V Plan, and other technical plans that support implementation of the technical effort (to all processes; applicable plans to technical processes);
- **Technical work directives**: e.g., work packages or task orders with work authorization (to applicable technical teams); and
- **Technical Planning Process work products:** Includes products needed to provide reports, records, and nondeliverable outcomes of process activities (to the Technical Data Management Process).

The resulting technical planning strategy constitutes an outline, or rough draft, of the SEMP. This serves as a starting point for the overall Technical Planning Process after initial preparation is complete. When preparations for technical planning are complete, the technical team should have a cost estimate and schedule for the technical planning effort. The budget and schedule to support the defined technical planning effort can then be negotiated with the project manager to resolve any discrepancies between what is needed and what is available. The SEMP baseline needs to be completed. Planning for the update of the SEMP based on programmatic changes needs to be developed and implemented. The SEMP needs to be approved by the appropriate level of authority.

6.1.2 Technical Planning Guidance

6.1.2.1 Work Breakdown Structure

Common to both the project management and systems engineering disciplines is the requirement for organizing and managing a system throughout its life cycle within a systematic and structured framework that is reflective of the work to be performed and the associated cost, schedule, technical, and risk data to be accumulated, summarized, and reported. (See NPR 7120.5.)

A key element of this framework is a hierarchical, product-oriented Work Breakdown Structure (WBS). Derived from both the physical and system architectures, the WBS provides a systematic, logical approach for defining and translating initial mission goals and technical concepts into tangible project goals, system products, and life-cycle support (or enabling) functions.

When appropriately structured and used in conjunction with sound engineering principles, the WBS supplies a common framework for subdividing the total project into clearly defined, product-oriented work components, logically related and sequenced according to hierarchy, schedule, and responsibility assignment.

A product-based WBS is the organizing structure for:

- Project and technical planning and scheduling.
- Cost estimation and budget formulation. (In particular, costs collected in a product-based WBS can be compared to historical data. This is identified as a primary objective by DOD standards for WBSs.)
- Defining the scope of Statements Of Work (SOWs) and specifications for contract efforts.
- Project status reporting, including schedule, cost, workforce, technical performance, and integrated cost/schedule data (such as Earned Value (EV) and Estimated cost At Completion (EAC)).
- Plans, such as the SEMP, and other documentation products, such as specifications and drawings.

The WBS provides a logical outline and vocabulary that describes the entire project and integrates information in a consistent way. If there is a schedule slip in one element of a WBS, an observer can determine which other WBS elements are most likely to be affected. Cost impacts are more accurately estimated. If there is a design change in one element of the WBS, an observer can determine which other WBS elements will most likely be affected, and these elements can be consulted for potential adverse impacts.

This subsection provides some techniques for developing a WBS and points out some mistakes to avoid.

6.1.2.1.1 Techniques for Developing the WBS

The composition and level of detail required in the WBS hierarchy is determined by the project management and technical teams based on careful consideration of the project's size and the

complexity, constraints, and risk associated with the technical effort. The initial WBS provides a structured framework for conceptualizing and defining the program/project objectives and for translating the initial concepts into the major systems, component products, and services to be developed, produced, and/or obtained. As successive levels of detail are defined, the WBS hierarchy evolves to reflect a comprehensive, complete view of both the total project effort and each system or end product to be realized throughout the project's life cycle.

Developing a successful product-based project WBS that exhibits a hierarchical division of deliverable items and associated services is likely to require several iterations through the project life cycle since it is not always obvious at the outset what the full extent of the work may be.

Product Breakdown Structure

Prior to developing a preliminary WBS, there should be some development of the system architecture to the point where a preliminary Product Breakdown Structure (PBS) can be created

The PBS and associated WBS can then be developed level by level from the top down with the specified prime product(s) at the top and the systems, segments, subsystems, etc., at successive lower levels. At the lowest level are products such as hardware items, software items, and information items (documents, databases, etc.) for which there is a cognizant engineer or manager.

In this approach, a project-level systems engineer finalizes the PBS at the project level and provides a draft PBS for the next lower level. The WBS is then derived by adding appropriate services such as management and systems engineering to that lower level. This process is repeated recursively until a WBS exists down to the desired cost account level.

An alternative approach is to define all levels of a complete PBS in one design activity and then develop the complete WBS. When this approach is chosen, it is necessary to take great care to develop the PBS so that all products are included and all assembly/Integration and Verification (I&V) branches are correct. The involvement of people who will be responsible for the lower-level WBS elements is recommended.

Branch points in the hierarchy should show how the PBS elements are to be integrated. The WBS is built, in part, from the PBS by adding, at each branch point of the PBS, any necessary service elements, such as management, systems engineering, Integration and Verification (I&V), and integrated logistics support. If several WBS elements require similar equipment or software, then a higher-level WBS element might be defined from the system level to perform a block buy or a development activity; e.g., system support equipment. Figure 6.1-4 shows the relationship between a system, a PBS, and a WBS.

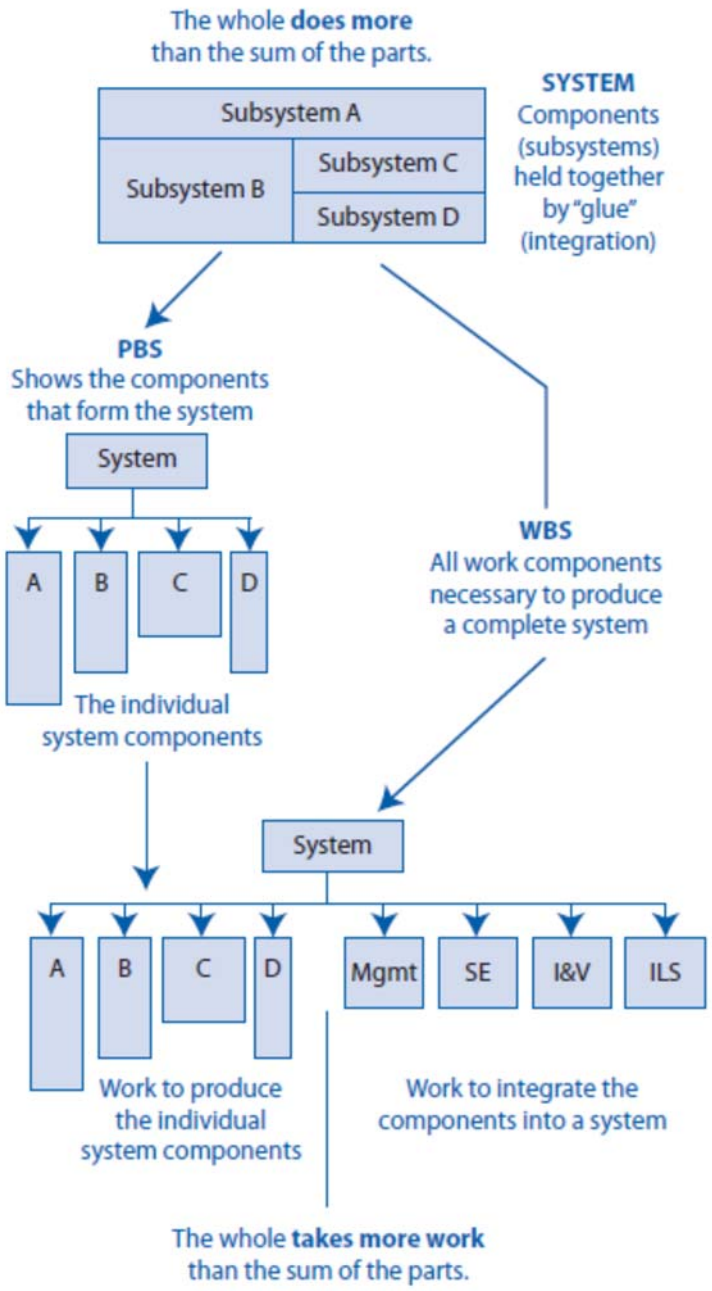


Figure 6.1-4 Relationship between a System, a PBS, and a WBS

A project WBS should be carried down to the cost account level appropriate to the risks to be managed. The appropriate level of detail for a cost account is determined by management's desire to have visibility into costs, balanced against the cost of planning and reporting.

Contract WBS

Contractors may have a Contract WBS (CWBS) that is appropriate to their need to control costs. A summary CWBS, consisting of the upper levels of the full CWBS, is usually included in the

project WBS to report costs to the contracting organization. WBS elements should be identified by title and by a numbering system that performs the following functions:

- Identifies the level of the WBS element;
- Identifies the higher-level element into which the WBS element will be integrated; and
- Shows the cost account number of the element.

WBS Dictionary

A WBS should also have a companion WBS dictionary that contains each element's title, identification number, objective, description, and any dependencies (e.g., receivables) on other WBS elements. This dictionary provides a structured project description that is valuable for orienting project members and other interested parties. It fully describes the products and/or services expected from each WBS element.

6.1.2.1.2 Common Errors in Developing a WBS

There are three common errors found in WBSs:

- **Error 1:** The WBS describes functions, not products. This makes the project manager the only one formally responsible for products.
- **Error 2:** The WBS has branch points that are not consistent with how the WBS elements will be integrated. For instance, in a flight operations system with a distributed architecture, there is typically software associated with hardware items that will be integrated and verified at lower levels of a WBS. It would then be inappropriate to separate hardware and software as if they were separate systems to be integrated at the system level. This would make it difficult to assign accountability for integration and to identify the costs of integrating and testing components of a system.
- **Error 3:** The WBS is inconsistent with the PBS. This makes it possible that the PBS will not be fully implemented and generally complicates the management process.

Some examples of these errors are shown in figure 6.1-5. Each one prevents the WBS from successfully performing its roles in project planning and organizing. These errors are avoided by using the WBS development techniques described above.

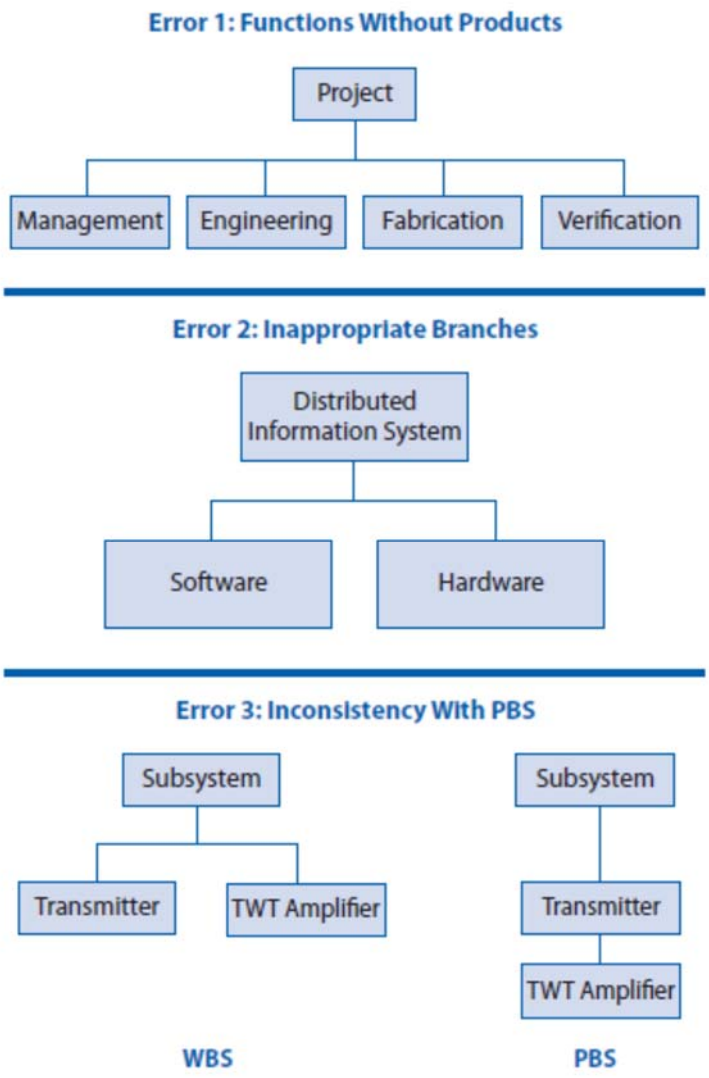


Figure 6.1-5 Examples of WBS Development Errors

6.1.2.1.3 WBS Evolution

Decomposition of the major deliverables into unique, tangible product or service elements should continue to a level representative of how each WBS element will be planned and managed. Whether assigned to in-house or contractor organizations, these lower WBS elements are subdivided into subordinate tasks and activities and aggregated into the work packages and control accounts used to populate the project's cost plans, schedules, and performance metrics.

At a minimum, the WBS should reflect the major system products and services to be developed and/or procured, the enabling (support) products and services, and any high-cost and/or high-risk product elements residing at lower levels in the hierarchy.[3] The baseline WBS is documented as

[3] IEEE Standard 1220, section C.3, "The system products and life cycle enabling products should be jointly engineered and once the enabling products and services are identified, should be treated as systems in the overall

part of the program plan and used to structure the SEMP. The cost estimates and the WBS dictionary are maintained throughout the project's life cycle to reflect the project's current scope.

The preparation and approval of three key program/project documents, the Formulation Authorization Document (FAD), the program commitment agreement, and the program/project plans are significant contributors to early WBS development. The initial contents of these documents establishes the purpose, scope, objectives, and applicable agreements for the program of interest and includes a list of approved projects, control plans, management approaches, and any commitments and constraints identified. As the design of the product matures, the WBS is expanded and updated to cover the additional work content.

The technical team selects the appropriate system design processes to be employed in the top-down definition of each product in the system structure. Subdivision of the project and system architecture into smaller, more manageable components provides logical summary points for assessing the overall project's accomplishments and for measuring cost and schedule performance.

Once the initial mission goals and objectives have evolved into the build-to or final design, the WBS is refined and updated to reflect the evolving scope and architecture of the project and the bottom-up realization of each product in the system structure.

Throughout the applicable life-cycle phases, the WBS and WBS dictionary are updated to reflect the project's current scope and to ensure control of high-risk and cost/schedule performance issues.

The technical team should receive planning guidelines from the project office. The technical team should provide the project office with any appropriate tailoring or expansion of the systems engineering WBS element, and have project-level concurrence on the WBS and WBS dictionary before issuing technical work directives.

For additional information on the WBS, refer to *NASA/SP-2010-3404, NASA Work Breakdown Structure Handbook.*

system hierarchy."

WBS Hierarchies for Systems

While all types of programs/projects are expected to form a WBS, a standard Level 2 WBS structure is mandated for NASA space flight projects in NPR 7120.5. It is expected that the programs/projects will take this standard and mature it out into the lower levels with the help of the technical team. The WBS mandated levels reflect the scope of a major Agency project and, therefore, the WBS is structured to include the development, operation, and disposal of more than one major system of interest during the project's normal life cycle.

WBS hierarchies for NASA's space flight projects include high-level system products, such as payload, spacecraft, and ground systems, and enabling products and services, such as project management, systems engineering, and education. These standard product elements have been established to facilitate alignment with the Agency's accounting, acquisition, and reporting systems.

Creation of a technical WBS focuses on the development and realization of both the overall end product and each subproduct included as a lower level element in the overall system structure.

NPR 7123.1, NASA Systems Engineering Processes and Requirements, mandates a standard, systematic technical approach to system or end-product development and realization. Utilizing a building-block or product-hierarchy approach, the system architecture is successively defined and decomposed into subsystems (elements performing the operational functions of the system) and associated and interrelated subelements (assemblies, components, parts, and enabling life-cycle products).

The resulting hierarchy or family-product tree depicts the entire system architecture in a PBS. Recognized by Government and industry as a "best practice," utilization of the PBS and its building-block configuration facilitates both the application of NPR 7123.1's 17 common technical processes at all levels of the PBS structure and the definition and realization of successively lower level elements of the system's hierarchy.

Definition and application of the work effort to the PBS structure yields a series of functional subproducts or "children" WBS models. The overall parent or system WBS model is realized through the rollup of successive levels of these product-based, subelement WBS models.

Each WBS model represents one unique unit or functional end product in the overall system configuration and, when related by the PBS into a hierarchy of individual models, represents one functional system end product or "parent" WBS model.

(See NPR 7120.5, NASA Space Flight Program and Project Management Requirements.)

6.1.2.2 Cost Definition and Modeling

This subsection deals with the role of cost in the systems engineering process, how to measure it, how to control it, and how to obtain estimates of it. The reason costs and their estimates are of great importance in systems engineering goes back to a principal objective of systems engineering: fulfilling the system's goals in the most cost-effective manner. The cost of each alternative should be one of the outcome variables in trade studies performed during the systems engineering process.

One role, then, for cost estimates is in helping to choose rationally among alternatives. Another is as a control mechanism during the project life cycle. Cost measures produced for project life-cycle reviews are important in determining whether the system goals and objectives are still

deemed valid and achievable, and whether constraints and boundaries are worth maintaining. These measures are also useful in determining whether system goals and objectives have properly flowed down through to the various subsystems.

As a product matures through its life cycle, cost estimates should mature as well. At each review, cost estimates need to be presented and compared to the funds likely to be available to complete the project. The cost estimates presented at early reviews should be given special attention since they usually form the basis for the initial cost commitment for the project to NASA management and to external stakeholders such as OMB and Congress. The systems engineer should be able to provide realistic cost estimates to the project manager with the support of a project or independent cost estimator. It is highly advisable for the systems engineer to enlist the support of Agency cost estimating organizations. The NASA Headquarters Cost Analysis Division can provide such contacts at each NASA Center. In the absence of such estimates, overruns are likely to occur, and the credibility of the entire system development process, both internal and external, is threatened.

6.1.2.2.1 Life-Cycle Cost and Other Cost Measures

A number of questions need to be addressed so that life-cycle costs are properly treated in systems analysis and engineering. These questions include:

- What costs should be counted?
- How should costs occurring at different times be treated?
- What about costs that cannot easily be measured in dollars?

6.1.2.2.2 What Costs Should Be Counted

The most comprehensive measure of the cost of an alternative is its life-cycle cost. According to NPR 7120.5, a system's life-cycle cost is:

> ". . . the total of the direct, indirect, recurring, nonrecurring, and other related expenses incurred and estimated to be incurred, in the design, development, verification, production, deployment, prime mission operation, maintenance, support, and disposal of a project including closeout, but not extended operations. The LCC of a project or system can also be defined as the total cost of ownership over the project or system's life cycle from Formulation (excluding Pre-Phase A) through Implementation (excluding extended operations). The LCC includes the cost of the launch vehicle."

Cost may be monetary or actions converted to monetary worth (such as training time, crew time for operations and maintenance, system logistics, etc.).

6.1.2.2.3 Costs Occurring Over Time

The life-cycle cost combines costs that typically occur over the entire project life, including both acquisition costs and operations costs. Operations costs typically have a multiplier (annual or per sortie) and as a result, are likely to be the primary contributor to multiyear program costs. Efforts to minimize operational costs will have the greatest impact in reducing overall life-cycle costs.

To facilitate engineering trades and comparison of system costs, these real year costs are typically normalized to constant year values or often estimated to begin with in some year constant dollars. This removes the impact of inflation from all estimates and allows ready comparison of alternative approaches.

In those instances where major portfolio architectural trades are being conducted, it may be necessary to perform formal cost benefit analyses (also sometimes called "discounted cash flow analyses" or "net present value analyses") or evaluate leasing versus purchase alternatives. In those trades, engineers and cost analysts should follow the guidance provided in Office of Management and Budget (OMB) Circular A-94 on rate of return and net present value calculation in comparing alternatives.

6.1.2.2.4 Difficult-to-Measure Costs

In practice, estimating some costs poses special problems. These special problems, which are not unique to NASA systems, usually occur in two areas: (1) when alternatives have differences in the irreducible chances of loss of life, and (2) when externalities are present. Two examples of externalities that impose costs are pollution caused by some launch systems and the creation of orbital debris. Because it is difficult to place a dollar figure on these resource uses, they are generally called "incommensurable costs." The general treatment of these types of costs in trade studies is not to ignore them, but instead to keep track of them along with other costs. If these elements are part of the trade space, it is generally advisable to apply OMB Circular A-94 approaches to those trades.

6.1.2.2.5 Controlling Life-Cycle Costs

The project manager/systems engineer should ensure that the probabilistic life-cycle cost estimate is compatible with NASA's budget and strategic priorities. The current policy is that projects are to submit budgets sufficient to ensure a 70 percent probability of achieving the objectives within the proposed resources (see Section 6.1.2.2.8 on Joint Confidence Level). Project managers and systems engineers should establish processes to estimate, assess, monitor, and control the project's life-cycle cost through every phase of the project.

Early decisions in the systems engineering process tend to have the greatest effect on the resultant system life-cycle cost. Typically, by the time the preferred system architecture is selected, between 50 and 70 percent of the system's life-cycle cost has been locked in (see Figure 2.5-3). By the time a preliminary system design is selected, this figure may be as high as 90 percent. This means that very little design modification can be conducted after preliminary system design without major cost impacts and presents a major dilemma to the systems engineer, who should lead this selection process. Just at the time when decisions are most critical, the state of information about the alternatives is least certain. Uncertainty about costs is a fact of systems engineering, and that uncertainty should be accommodated by complete and careful analysis of the project risks and provision of sufficient margins (cost, technical, and schedule) to ensure success. There are a number of estimating techniques to assist the systems engineer and project manager in providing for uncertainty and unknown requirements. Additional information on these techniques can be found in the *NASA Cost Estimating Handbook*.

This suggests that efforts to acquire better information about the life-cycle cost of each alternative early in the project life cycle (Phases Pre A, A, and B) potentially have very high payoffs. The systems engineer needs to identify the principal life-cycle cost drivers and the risks associated with the system design, manufacturing, and operations. Consequently, it is particularly important with such a system to bring in the crosscutting and specialty engineering disciplines and subject matter experts such as reliability, maintainability, supportability, and operations engineering early in the systems engineering process, as they are essential to proper life-cycle cost estimation.

One mechanism for controlling life-cycle cost is to establish a life-cycle cost management program as part of the project's management approach. Life-cycle cost management has sometimes been called "design-to-life-cycle cost." Such a program establishes life-cycle cost as a design goal, perhaps with subgoals for acquisition costs or operations and support costs. More specifically, the objectives of a life-cycle cost management program are to:

- Identify a common set of ground rules and assumptions for life-cycle cost estimation;
- Manage to a cost baseline and maintain traceability to the technical baseline with documentation for subsequent cost changes;
- Ensure that best-practice methods, tools, and models are used for life-cycle cost analysis;
- Use Earned Value Management (EVM) techniques to track the estimated life-cycle cost throughout the project life cycle (the NASA Headquarters Cost Analysis Division and Office of Chief Engineer can provide contacts for EVM expertise at each NASA Center).
- *And, most importantly*, integrate life-cycle cost considerations into the design and development process using trade studies and formal change request assessments.

Trade studies and formal change request assessments provide the means to balance the effectiveness and life-cycle cost of the system. The complexity of integrating life-cycle cost considerations into the design and development process should not be underestimated, but neither should the benefits, which can be measured in terms of greater cost-effectiveness. The existence of a rich set of potential life-cycle cost trades makes this complexity even greater. Sections 2.5, 2.6 and 7.9 particularly emphasize the importance of trades that address the life-cycle cost assessment of the human element(s) in the operation and maintenance of the system. Functionality and efficiency should not be deferred to human operators/maintainers during design and development without performing trade studies to evaluate near-term cost savings versus life-cycle cost growth.

6.1.2.2.6 Cost Estimating Methods

Various cost-estimating methodologies are utilized throughout a program's life cycle. These include parametric, analogous, and engineering (grassroots).

- **Parametric:** Parametric cost models are used in the early stages of project development when there is limited program and technical definition. Such models involve collecting relevant historical data at an aggregated level of detail and relating it to the area to be estimated through the use of mathematical techniques to create cost-estimating relationships. Normally, less detail is required for this approach than for other methods.

- **Analogous:** This is based on the approach that most new programs originated or evolved from existing programs or simply represent a new combination of existing components. It uses actual costs of similar existing or past programs and adjusts for complexity, technical, or physical differences to derive the new system estimate. This method would be used when there is insufficient actual cost data to use as a basis for a detailed approach, but there is a sufficient amount of program and technical definition.
- **Engineering (Grassroots):** These bottom-up estimates are the result of rolling up the costs estimated by each organization performing work described in the WBS. Properly done, grassroots estimates can be quite accurate, but each time a "what if" question is raised, a new estimate needs to be made. Each change of assumptions voids at least part of the old estimate. Because the process of obtaining grassroots estimates is typically time-consuming and labor-intensive, the number of such estimates that can be prepared during trade studies is in reality severely limited.

The type of cost-estimating method used depends on the adequacy of program definition, level of detail required, availability of data, and time constraints. For example, during the early stages of a program, a conceptual study considering several options might need an estimating method that requires no actual cost data and limited program definition on the systems being estimated. A parametric model would be a sound approach at this point. Once a design is baselined and the program is more adequately defined, an analogy approach may become appropriate. As detailed actual cost data are accumulated, a grassroots methodology could be used. However, other projects may use any one or combination of these cost estimating tools at any phase of the life cycle.

More information on cost-estimating methods and the development of cost estimates can be found in the *NASA Cost Estimating Handbook*.

6.1.2.2.7 Integrating Cost Model Results for a Complete Life-Cycle Cost Estimate

A number of parametric cost models are available for costing NASA systems. A list of the models currently in use may be found in an appendix in the *NASA Cost Estimating Handbook* or by contacting the NASA Headquarters Cost Analysis Division. Unfortunately, no one alone is sufficient to estimate life-cycle cost. Assembling an estimate of life-cycle cost often requires that several different models (along with the other two techniques) be used together. Whether generated by parametric models, analogous, or grass roots methods, the estimated cost of the hardware element should frequently be "wrapped" or have factors applied to estimate the costs associated with management, systems engineering, test, etc., of the systems being estimated. The NASA full-cost factors also should be applied separately; however, applying full cost can be complicated. For example, if some of the work is being performed by NASA in-house and that cost is separately estimated *and* if the model or data to estimate the contracted part is based on past projects in which all the work was done using contracts, this would result in double counting because the model is estimating the total content unaware that some of the work is being done in-house by NASA. The opposite erroneous effect could occur if the model or data being used in the estimate is based on past projects that enjoyed some of the work being done "off-book" by NASA in-house. In this case, using only the model or data to estimate cost would erroneously leave out the cost of that work previously done in-house. The NASA Headquarters Cost Analysis

Division can provide guidance on full cost adjustments. To integrate the costs being estimated by these different models, the systems engineer should ensure that the inputs to and assumptions of the models are consistent, that all relevant life-cycle cost components are covered, and that the phasing of costs is correct. Estimates from different sources are often expressed in different year *constant* dollars, which should be combined. For budget inputs, appropriate inflation factors should be applied to enable construction of a total life-cycle cost estimate in real year dollars. Guidance on the use of inflation rates for new projects and for budget submissions for ongoing projects can be found in the annual NASA strategic guidance. Also, the NASA Headquarters Cost Analysis Division maintains an inflation table that can be used for converting costs to any price level ("year dollars") desired.

Cost models frequently produce a cost estimate for the first unit of a hardware item, but where the project requires multiple units, a learning curve can be applied to the first unit cost to obtain the required multiple-unit estimate. Learning curves are based on the concept that resources required to produce each additional unit decline as the total number of units produced increases. The learning curve concept is used primarily for uninterrupted manufacturing and assembly tasks, which are highly repetitive and labor intensive. The major premise of learning curves is that each time the product quantity doubles, the resources (labor hours) required to produce the product will reduce by a determined percentage of the prior quantity resource requirements. The two types of learning curve approaches are unit curve and cumulative average curve. The systems engineer can learn more about the calculation and use of learning curves in the *NASA Cost Estimating Handbook*.

Models frequently provide a cost estimate of the total acquisition effort without providing a recommended phasing of costs over the life cycle. The systems engineer can use a set of phasing algorithms based on the typical ramping-up and subsequent ramping-down of acquisition costs for that type of project if a detailed project schedule is not available to form a basis for the phasing of the effort. A normal distribution curve, or beta curve, is one type of function used for spreading parametrically derived cost estimates and for R&D contracts where costs build up slowly during the initial phases and then escalate as the midpoint of the contract approaches. A beta curve is a combination of percentage spent against percentage time elapsed between two points in time. More about beta curves can be found in an appendix of the *NASA Cost Estimating Handbook*. If the cost estimate is being performed as part of a Joint Confidence Level (JCL) analysis, the project schedule tool (e.g. Microsoft Project, Primavera, etc.) can be used to resource-load the schedule with cost, which will then be phased consistent with the project schedule. (See Section 6.1.1.2 and Section 6.1.2.2.8.)

Although parametric cost models for space systems are already available, their proper use usually requires a considerable investment in learning how to appropriately utilize the models. For projects outside of the domains of these existing cost models, new cost models may be needed to support trade studies. Efforts to develop these models need to begin early in the project life cycle to ensure their timely application during the systems engineering process. Again, the systems engineer is advised to enlist the assistance of the Agency cost-estimating organizations and the NASA Headquarters Cost Analysis Division. Whether existing models or newly created ones are used, the SEMP and its associated life-cycle cost management plan should identify which (and how) models are to be used during each phase of the project life cycle.

6.1.2.2.8 Joint Confidence Level

What is JCL?

A relatively new initiative in NASA project cost analysis is the adoption of joint confidence level estimating. Joint Confidence Level (JCL) is an integrated uncertainty analysis of cost and schedule (See Figure 6.1-6). The result of a JCL indicates the probability that a project's cost will be equal to or less than the targeted cost *and* that the schedule will be equal to or less than the targeted finish.

Why Do A JCL?

JCL analysis provides a cohesive and holistic picture of the project's ability to achieve cost and schedule goals by systematically integrating technical, cost, schedule, and risk data. As an integrating framework, a JCL can show the impacts of risk to a project as well as highlight the relationship between cost and schedule. This relations hip can be extremely important in situations with constrained budgets. A complete JCL analysis can also facilitate transparency with stakeholders on expectations and the probability of meeting those expectations.

When Is A JCL Required?

NASA requires that a JCL analysis be completed and submitted at Key Decision Point (KDP)-C for all projects above $250 million. However, projects should start planning for their JCL when producing their probabilistic range estimates. In the Formulation Phase, specifically for KDP-B, NASA calls for programs and projects to provide probabilistic range analysis on both their cost and schedule estimates. This analysis is then used to determine a high and a low estimate for cost, and for schedule. The community has identified two candidate methodologies for producing the risk estimates and associated results: (1) complete parametric estimates of cost and schedule, or (2) complete a JCL. Conducting a JCL at KDP-B is not required primarily because projects typically do not have detailed plans available to support an in-depth JCL analysis, and by design, the probabilistic range estimate requirement at KDP-B is intended to "bound-the-problem." Conducting a parametric estimate of schedule and cost utilizes the historical data and performance of the agency and provides a valuable estimate of the range of possibilities.

JCL Nuts and Bolts

To calculate a JCL, the project uses a process that combines its cost, schedule, and risk into a single model that can generate a probabilistic assessment of the level of confidence of achieving a specific cost-schedule goal.

Once a baseline is approved, NASA policy does not require a project to maintain the artifacts used to calculate the JCL. However, the Agency does utilize a variety of performance metrics to assess how well the project is performing against its plan. If these metrics show that a project's performance varies significantly from its plan, the project may need to replan, but Agency policy only requires a repeat calculation of the JCL in the event the project requires a rebaseline. JCL analysis can provide valuable insights as a management tool; however, the only Agency requirement for JCL is at KDP-C.

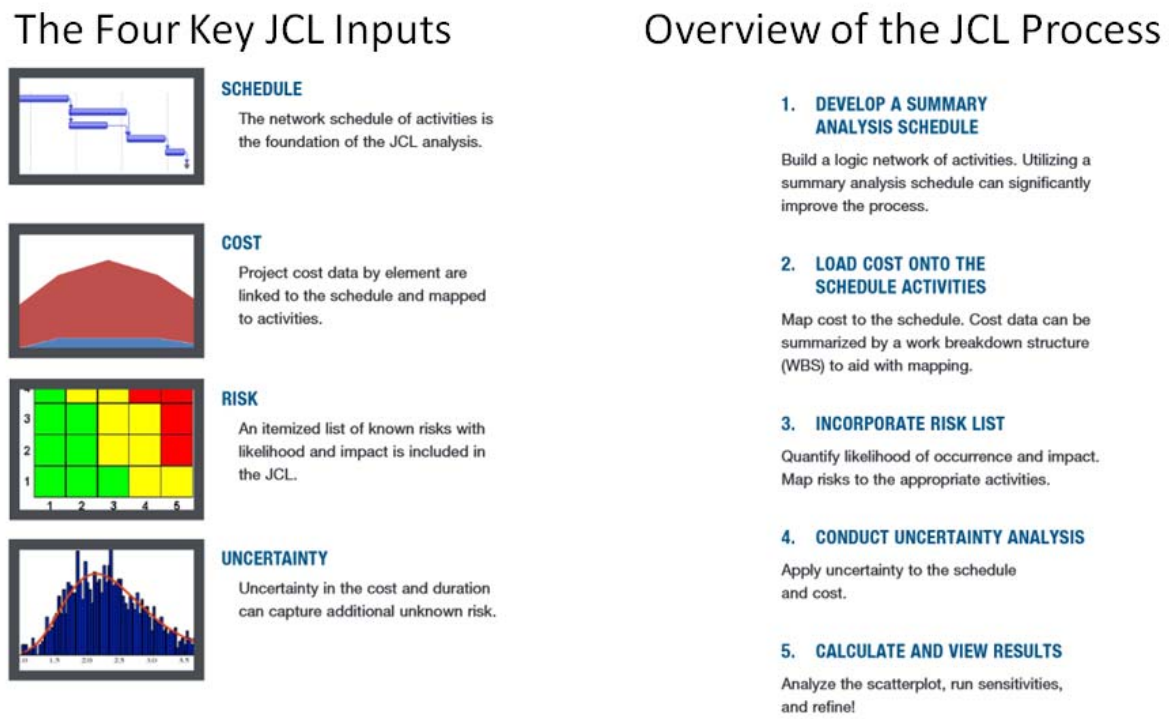
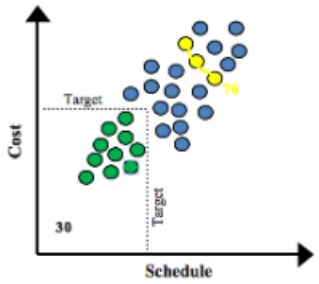


Figure 6.1-6 JCL Process Overview

Understanding the JCL Scatterplot

The JCL scatterplot is a standard XY chart with schedule on the X-axis and cost on the Y-axis pair (see Figure 6.1-7). The JCL calculation is based on the number of results in the target area for both cost and schedule and is expressed as a percentage of all simulation results displayed on the scatterplot. Establishing a cost and schedule target (black dotted lines) divides the scatterplot into two areas. One area contains results that are at or beneath the target (shown in green). The other area contains results that exceed the target (shown in blue). The yellow points and frontier line represent all results from the simulation that meet a desired joint level. Multiple points from the simulation may meet the JCL target. Each of the yellow points would establish a new target area large enough to meet the desired JCL.



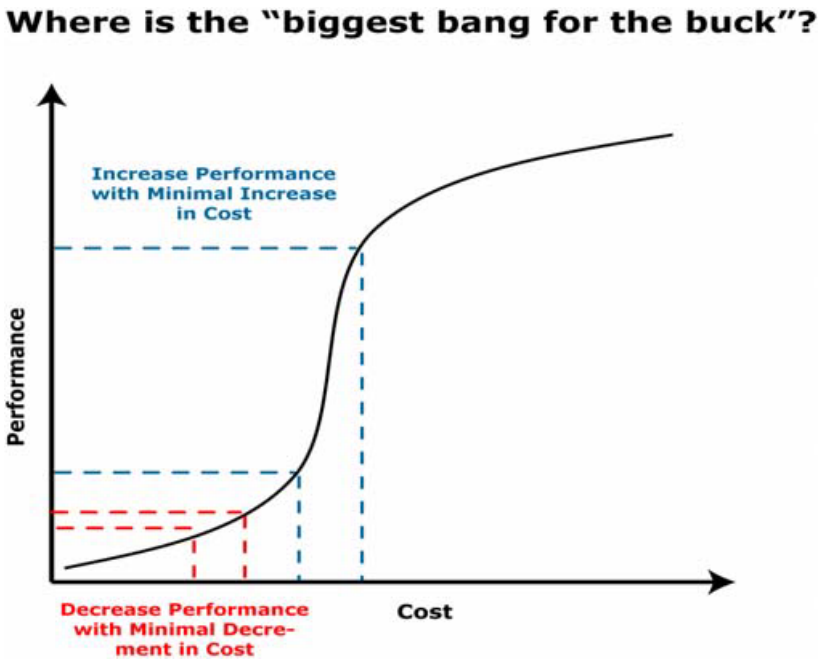


Figure 6.1-7 Simplified JCL Scatterplot

Frequently asked questions about JCL can be seen at the following link: http://www.nasa.gov/pdf/394931main_JCL_FAQ_10_12_09.pdf

6.1.2.2.9 Trade Studies: Balancing Performance, Cost, Schedule, and Risk

Trade studies are at the heart of the affordability process, and their solutions are often in a multi-dimensional trade space bounded by a cost element and by one or more performance parameters (for trade studies technical guidance and process description, refer to Section 6.8.2.1). Figure 6.1-8 illustrates a simplified, two dimensional trade space with a plot connecting candidate design alternatives. A multi-dimensional trade space may be substituted to show the interaction of multiple cost drivers, including performance, schedule, and risk. Solutions (data points) at the far left of the trade space may show alternatives that look attractive from a cost perspective but that may not satisfy even the threshold (minimal required) performance requirements. Similarly, data points at the far right may be alternatives that exceed the threshold cost boundary, only to provide performance beyond the requirement, which may not justified.

For additional information on performing trade studies, see Section 6.6.2.1.

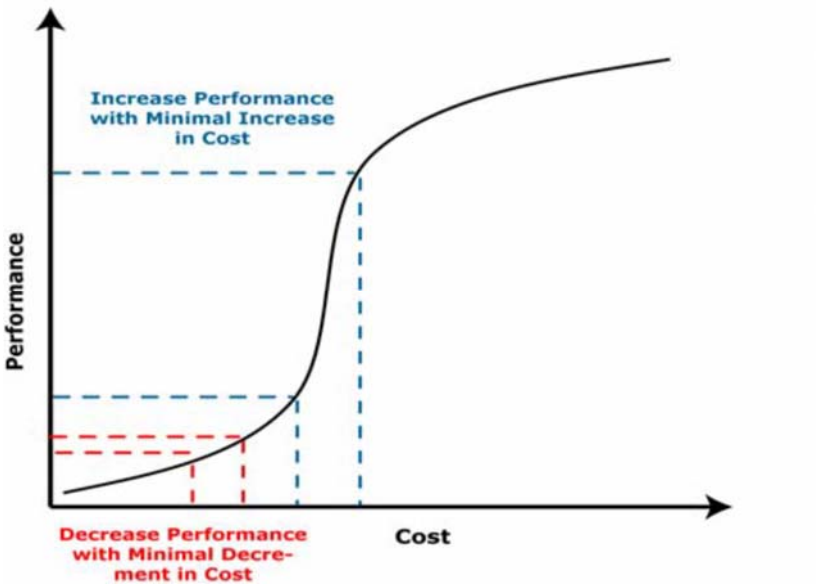


Figure 6.1-8 Cost versus Performance

6.1.2.3 Lessons Learned

No section on technical planning guidance would be complete without the effective integration and incorporation of the lessons learned relevant to the project.

6.1.2.3.1 Systems Engineering Role in Lessons Learned

Systems engineers are one of the main users and contributors to lessons learned systems. A lesson learned is knowledge or understanding gained by experience—either a successful test or mission or a mishap or failure. Systems engineers compile lessons learned to serve as historical documents, requirements' rationales, and other supporting data analysis. Systems engineering practitioners collect lessons learned during program and project plans, key decision points, life-cycle phases, systems engineering processes, and technical reviews. Systems engineers' responsibilities include knowing how to utilize, manage, create, and store lessons learned and knowledge management best practices.

6.1.2.3.2 Utilization of Lessons Learned Best Practice

Lessons learned are important to future programs, projects, and processes because they show hypotheses and conclusive insights from previous projects or processes. Practitioners determine how previous lessons from processes or tasks impact risks to current projects and implement those lessons learned that improve design and/ or performance.

To pull in lessons learned at the start of a project or task:

- Search the NASA Lessons Learned Information System (LLIS) database using keywords of interest to the new program or project. In addition, other organizations doing similar work may have publicly available databases with lessons learned. For example, the Chemical Safety Board has a good series of case study reports on mishaps.

- Supporting lessons from each engineering discipline should be reflected in the program and project plans. Even if little information was found, the search for lessons learned can be documented.
- Compile lessons by topic and/or discipline throughout the entire system life cycle, including operations.
- Include metrics as part of lessons learned.
- Review and select knowledge gained from particular lessons learned.
- Determine how these lessons learned may represent potential risk to the current program or project.
- Incorporate knowledge gained into the project database for risk management, cost estimate, and any other supporting data analysis.

As an example, a systems engineer working on the concept for an instrument for a spacecraft might search the lessons learned database using the keywords "environment," "mishap," or "configuration management." One of the lessons learned that search would bring up is #1514, a lesson from Chandra. A rebaseline of the program in 1992 removed two instruments, changed Chandra's orbit from low Earth to high elliptical, and simplified the thermal control concept from the active control required by one of the descoped instruments to passive "cold-biased" surface plus heaters. This change in thermal control concept mandated silver Teflon thermal control surfaces. The event driving the lesson was a severe spacecraft charging and an electrostatic discharge environment. The event necessitated an aggressive electrostatic discharge test and circuit protection effort that cost over $1 million, according to the database. The Teflon thermal control surfaces plus the high elliptical orbit created the electrostatic problem. Design solutions for one environment were inappropriate in another environment. The lesson learned was that any orbit modifications should trigger a complete new iteration of the systems engineering processes starting from requirements definition. Rebaselining a program should take into account change in the natural environment before new design decisions are made. This lesson would be valuable to keep in mind when changes occur to baselines on the program currently being worked on.

6.1.2.3.3 Management of Lessons Learned Best Practice

Capturing lessons learned is a function of good management practice and discipline. Too often lessons learned are missed because they should have been developed and managed within, across, or between life-cycle phases. There is a tendency to wait until resolution of a situation to document a lesson learned, but the unfolding of a problem at the beginning is valuable information and hard to recreate later. It is important to document a lesson learned as it unfolds, particularly as resolution may not be reached until a later phase. Since detailed lessons are often hard for the human mind to recover, waiting until a technical review or the end of a project to collect the lessons learned hinders the use of lessons and the evolution of practice. A mechanism for managing and leveraging lessons as they occur, such as monthly lessons learned briefings or some periodic sharing forums, facilitates incorporating lessons into practice and carrying lessons into the next phase.

At the end of each life-cycle phase, practitioners should use systems engineering processes and procedural tasks as control-gate cues. All information passed across control gates should be managed in order to successfully enter the next phase, process, or task.

The systems engineering practitioner should make sure all lessons learned in the present phase are concise and conclusive. Conclusive lessons learned contain series of events that formulate abstracts and driving events. Irresolute lessons learned may be rolled into the next phase to await proper supporting evidence. Project managers and the project technical team should make sure lessons learned are recorded in the Agency database at the end of all life-cycle phases, major systems engineering processes, key decision points, and technical reviews. For additional information on the lessons learned process refer to NPD 7120.6, Knowledge Policy on Programs and Projects.

6.2 Requirements Management

Requirements management activities apply to the management of all stakeholder expectations, customer requirements, and technical product requirements down to the lowest level product component requirements (hereafter referred to as expectations and requirements). This includes physical functional and operational requirements, including those that result from interfaces between the systems in question and other external entities and environments. The Requirements Management Process is used to:

- Identify, control, decompose, and allocate requirements across all levels of the WBS.
- Provide bidirectional traceability.
- Manage the changes to established requirement baselines over the life cycle of the system products.

Definitions

Traceability: A discernible association between two or more logical entities such as requirements, system elements, verifications, or tasks.

Bidirectional traceability: The ability to trace any given requirement/expectation to its parent requirement/expectation and to its allocated children requirements / expectation..

6.2.1 Process Description

Figure 6.2-1 provides a typical flow diagram for the Requirements Management Process and identifies typical inputs, outputs, and activities to consider in addressing requirements management.

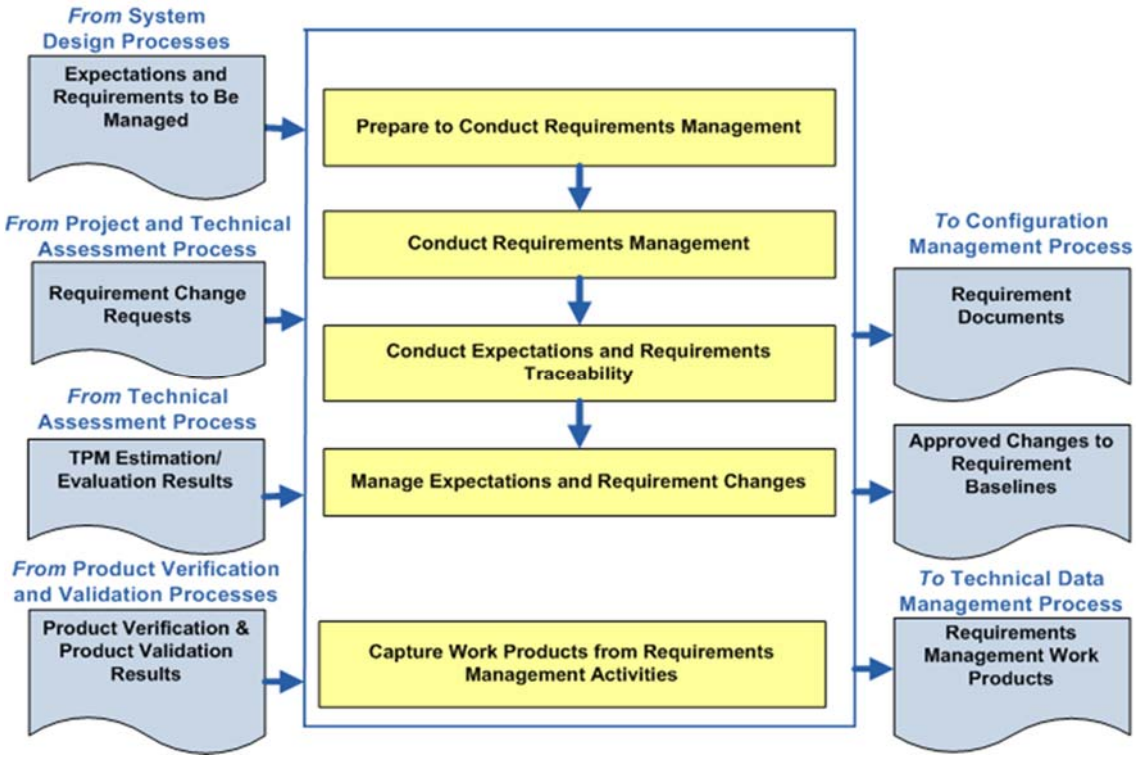


Figure 6.2-1 Requirements Management Process

6.2.1.1 Inputs

There are several fundamental inputs to the Requirements Management Process.

- **Expectations and requirements to be managed**: Requirements and stakeholder expectations are identified during the system design processes, primarily from the Stakeholder Expectations Definition Process and the Technical Requirements Definition Process.
- **Requirement change requests:** The Requirements Management Process should be prepared to deal with requirement change requests that can be generated at any time during the project life cycle or as a result of reviews and assessments as part of the Technical Assessment Process.
- **TPM estimation/evaluation results**: TPM estimation/evaluation results from the Technical Assessment Process provide an early warning of the adequacy of a design in satisfying selected critical technical parameter requirements. Variances from expected values of product performance may trigger changes to requirements.
- **Product verification and validation results:** Product verification and product validation results from the Product Verification and Product Validation Processes are mapped into the requirements database with the goal of verifying and validating all requirements.

6.2.1.2 Process Activities

6.2.1.2.1 Prepare to Conduct Requirements Management

Preparing to conduct requirements management includes gathering the requirements that were defined and baselined during the Requirements Definition Process. Identification of the sources/owners of each requirement should be checked for currency. The organization (e.g., change board) and procedures to perform requirements management are established.

6.2.1.2.2 Conduct Requirements Management

The Requirements Management Process involves managing all changes to expectations and requirements baselines over the life of the product and maintaining bidirectional traceability between stakeholder expectations, customer requirements, technical product requirements, product component requirements, design documents, and test plans and procedures. The successful management of requirements involves several key activities:

- Establish a plan for executing requirements management.
- Receive requirements from the system design processes and organize them in a hierarchical tree structure.
- Maintain bidirectional traceability between requirements.
- Evaluate all change requests to the requirements baseline over the life of the project and make changes if approved by change board.
- Maintain consistency between the requirements, the ConOps, and the architecture/design, and initiate corrective actions to eliminate inconsistencies.

6.2.1.2.3 Conduct Expectations and Requirements Traceability

As each requirement is documented, its bidirectional traceability should be recorded. Each requirement should be traced back to a parent/source requirement or expectation in a baselined document or identified as *self-derived* and concurrence on it sought from the next higher level requirements sources. Examples of self-derived requirements are requirements that are locally adopted as good practices or are the result of design decisions made while performing the activities of the Logical Decomposition and Design Solution Processes.

The requirements should be evaluated, independently if possible, to ensure that the requirements trace is correct and that it fully addresses its parent requirements. If it does not, some other requirement(s) should complete fulfillment of the parent requirement and be included in the traceability matrix. In addition, ensure that all top-level parent document requirements have been allocated to the lower level requirements. If there is no parent for a particular requirement and it is not an acceptable self-derived requirement, it should be assumed either that the traceability process is flawed and should be redone or that the requirement is “gold plating” and should be eliminated. Duplication between levels should be resolved. If a requirement is simply repeated at a lower level and it is not an externally imposed constraint, it may not belong at the higher level. Requirements traceability is usually recorded in a requirements matrix or through the use of a requirements modeling application.

6.2.1.2.4 Managing Expectations and Requirement Changes

Throughout early Phase A, changes in requirements and constraints will occur as they are initially defined and matured. It is imperative that all changes be thoroughly evaluated to determine the impacts on the cost, schedule, architecture, design, interfaces, ConOps, and higher and lower level requirements. Performing functional and sensitivity analyses will ensure that the requirements are realistic and evenly allocated. Rigorous requirements verification and validation will ensure that the requirements can be satisfied and conform to mission objectives. All changes should be subjected to a review and approval cycle to maintain traceability and to ensure that the impacts are fully assessed for all parts of the system.

Once the requirements have been validated and reviewed in the System Requirements Review (SRR) in late Phase A, they are placed under formal configuration control. Thereafter, any changes to the requirements should be approved by a Configuration Control Board (CCB) or equivalent authority. The systems engineer, project manager, and other key engineers usually participate in the CCB approval processes to assess the impact of the change including cost, performance, programmatic, and safety.

Requirement changes during Phase B and C are more likely to cause significant adverse impacts to the project cost and schedule. It is even more important that these late changes are carefully evaluated to fully understand their impact on cost, schedule, and technical designs.

The technical team should also ensure that the approved requirements are communicated in a timely manner to all relevant people. Each project should have already established the mechanism to track and disseminate the latest project information. Further information on Configuration Management (CM) can be found in Section 6.5.

6.2.1.2.5 Key Issues for Requirements Management

Requirements Changes

Effective management of requirements changes requires a process that assesses the impact of the proposed changes prior to approval and implementation of the change. This is normally accomplished through the use of the Configuration Management Process. In order for CM to perform this function, a baseline configuration should be documented and tools used to assess impacts to the baseline. Typical tools used to analyze the change impacts are as follows:

- **Performance Margins:** This tool is a list of key performance margins for the system and the current status of the margin. For example, the propellant performance margin will provide the necessary propellant available versus the propellant necessary to complete the mission. Changes should be assessed for their impact on performance margins.
- **CM Topic Evaluators List:** This list is developed by the project office to ensure that the appropriate persons are evaluating the changes and providing impacts to the change. All changes need to be routed to the appropriate individuals to ensure that the change has had all impacts identified. This list will need to be updated periodically.
- **Risk System and Threats List:** The risk system can be used to identify risks to the project and the cost, schedule, and technical aspects of the risk. Changes to the baseline can affect the consequences and likelihood of identified risk or can introduce new risk to the project. A

threats list is normally used to identify the costs associated with all the risks for the project. Project reserves are used to mitigate the appropriate risk. Analyses of the reserves available versus the needs identified by the threats list assist in the prioritization for reserve use.

The process for managing requirements changes needs to take into account the distribution of information related to the decisions made during the change process. The Configuration Management Process needs to communicate the requirements change decisions to the affected organizations. During a board meeting to approve a change, actions to update documentation need to be included as part of the change package. These actions should be tracked to ensure that affected documentation is updated in a timely manner.

Requirements Creep

"Requirements creep" is the term used to describe the subtle way that requirements grow imperceptibly during the course of a project. The tendency for the set of requirements is to relentlessly increase in size during the course of development, resulting in a system that is more expensive and complex than originally intended. Often the changes are quite innocent and what appear to be changes to a system are really enhancements in disguise.

However, some of the requirements creep involves truly new requirements that did not exist, and could not have been anticipated, during the Technical Requirements Definition Process. These new requirements are the result of evolution, and if we are to build a relevant system, we cannot ignore them.

There are several techniques for avoiding or at least minimizing requirements creep:

- The first line of defense is a good ConOps that has been thoroughly discussed and agreed-to by the customer and relevant stakeholders.
- In the early requirements definition phase, flush out the conscious, unconscious, and undreamt-of requirements that might otherwise not be stated.
- Establish a strict process for assessing requirement changes as part of the Configuration Management Process.
- Establish official channels for submitting change requests. This will determine who has the authority to generate requirement changes and submit them formally to the CCB (e.g., a contractor-designated representative, project technical leads, customer/science team lead, or user).
- Measure the functionality of each requirement change request and assess its impact on the rest of the system. Compare this impact with the consequences of not approving the change. What is the risk if the change is not approved?
- Determine if the proposed change can be accommodated within the fiscal and technical resource budgets. If it cannot be accommodated within the established resource margins, then the change most likely should be denied.

6.2.1.2.6 Capture Work Products

These products include maintaining and reporting information on the rationale for and disposition and implementation of change actions, current requirement compliance status and expectation, and requirement baselines.

6.2.1.3 Outputs

Typical outputs from the requirements management activities are:

- **Requirements Documents:** Requirements documents are submitted to the Configuration Management Process when the requirements are baselined. The official controlled versions of these documents are generally maintained in electronic format within the requirements management tool that has been selected by the project. In this way, they are linked to the requirements matrix with all of its traceable relationships.
- **Approved Changes to the Requirements Baselines:** Approved changes to the requirements baselines are issued as an output of the Requirements Management Process after careful assessment of all the impacts of the requirements change across the entire product or system. A single change can have a far-reaching ripple effect, which may result in several requirement changes in a number of documents.
- **Various Requirements Management Work Products:** Requirements management work products are any reports, records, and undeliverable outcomes of the Requirements Management Process. For example, the bidirectional traceability status would be one of the work products that would be used in the verification and validation reports.

6.2.2 Requirements Management Guidance

6.2.2.1 Requirements Management Plan

The technical team should prepare a plan for performing the requirements management activities. This plan is normally part of the SEMP but can also stand alone. The plan should:

- Identify the relevant stakeholders who will be involved in the Requirements Management Process (e.g., those who may be affected by, or may affect, the product as well as the processes).
- Provide a schedule for performing the requirements management procedures and activities.
- Assign responsibility, authority, and adequate resources for performing the requirements management activities, developing the requirements management work products, and providing the requirements management services defined in the activities (e.g., staff, requirements management database tool, etc.). Define the level of configuration management/data management control for all requirements management work products.
- Identify the training for those who will be performing the requirements management activities.

6.2.2.2 Requirements Management Tools

For small projects and products, the requirements can usually be managed using a spreadsheet program. However, the larger programs and projects require the use of one of the available requirements management tools.

In selecting a tool, it is important to define the project's procedure for specifying how the requirements will be organized in the requirements management database tool and how the tool will be used. It is possible, given modern requirements management tools, to create a requirements management database that can store and sort requirements data in multiple ways according to the particular needs of the technical team. The organization of the database is not a trivial exercise and has consequences on how the requirements data can be viewed for the life of the project. It is important to organize the database so that it has all the views into the requirements information that the technical team is likely to need. Careful consideration should be given to how the flowdown of requirements and bidirectional traceability will be represented in the database.

Sophisticated requirements management database tools also have the ability to capture numerous requirement attributes in the tools' requirements matrix, including the requirements traceability and allocation links. For each requirement in the requirements matrix, the verification method(s), level, and phase are documented in the verification requirements matrix housed in the requirements management database tool (e.g., the tool associates the attributes of method, level, and phase with each requirement). It is important to make sure that the requirements management database tool is compatible with the verification and validation tools chosen for the project.

6.3 Interface Management

The definition, management, and control of interfaces are crucial to successful programs or projects. Interface management is a process to assist in controlling product development when efforts are divided among parties (e.g., Government, contractors, geographically diverse technical teams, etc.) and/or to define and maintain compliance among the products that should interoperate.

The basic tasks that need to be established involve the management of internal and external interfaces of the various levels of products and operator tasks to support product integration. These basic tasks are as follows:

- Define interfaces;
- Identify the characteristics of the interfaces (physical, electrical, mechanical, human, etc.);
- Ensure interface compatibility at all defined interfaces by using a process documented and approved by the project;
- Strictly control all of the interface processes during design, construction, operation, etc.;
- Identify lower level products to be assembled and integrated (from the Product Transition Process);
- Identify assembly drawings or other documentation that show the complete configuration of the product being integrated, a parts list, and any assembly instructions (e.g., torque requirements for fasteners);
- Identify end-product, design-definition-specified requirements (specifications), and configuration documentation for the applicable work breakdown structure model, including interface specifications, in the form appropriate to satisfy the product life-cycle phase success criteria (from the Configuration Management Process); and
- Identify product integration-enabling products (from existing resources or the Product Transition Process for enabling product realization).

6.3.1 Process Description

Figure 6.3-1 provides a typical flow diagram for the Interface Management Process and identifies typical inputs, outputs, and activities to consider in addressing interface management.

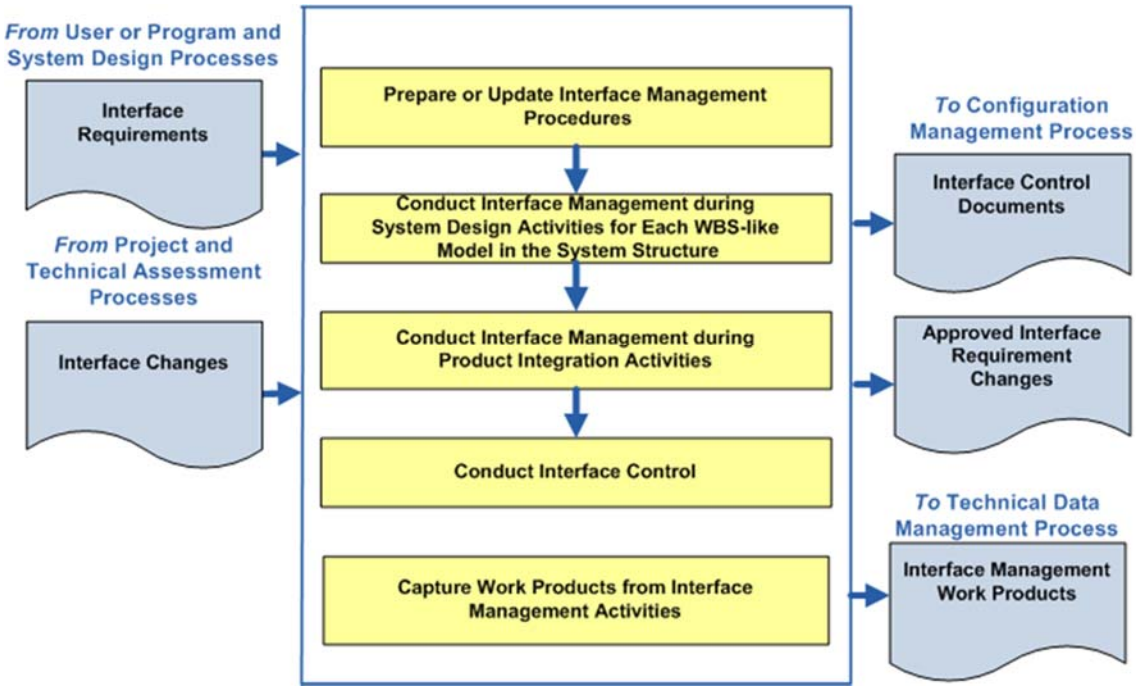


Figure 6.3-1 Interface Management Process

6.3.1.1 Inputs

Typical inputs needed to understand and address interface management would include the following:

- **Interface Requirements:** These include the internal and external functional, physical, and performance interface requirements developed as part of the Technical Requirements Definition Process for the product(s).
- **Interface Change Requests:** These include changes resulting from program or project agreements or changes on the part of the technical team as part of the Technical Assessment Process.

Other inputs that might be useful are:

- **System Description:** This allows the design of the system to be explored and examined to determine where system interfaces exist. Contractor arrangements will also dictate where interfaces are needed.
- **System Boundaries:** Documented physical boundaries, components, and/or subsystems, which are all drivers for determining where interfaces exist.
- **Organizational Structure:** Decisions on which organization will dictate interfaces, particularly when there is the need to jointly agree on shared interface parameters of a system. The program and project WBS will also provide organizational interface boundaries.
- **Boards Structure:** Defined board structure that identifies organizational interface responsibilities.

6.3.1.2 Process Activities

6.3.1.2.1 Prepare or Update Interface Management Procedures

These procedures establish the interface management responsibilities, what process will be used to maintain and control the internal and external functional and physical interfaces (including human), and how the change process will be conducted. Training of the technical teams or other support may also be required and planned.

6.3.1.2.2 Conduct Interface Management during System Design Activities

During project Formulation, the ConOps of the product is analyzed to identify both external and internal interfaces. This analysis will establish the origin, destination, stimuli, and special characteristics of the interfaces that need to be documented and maintained. As the system structure and architecture emerges, interfaces will be added and existing interfaces will be changed and should be maintained. Thus, the Interface Management Process has a close relationship to other areas, such as requirements definition and configuration management, during this period

6.3.1.2.3 Conduct Interface Management during Product Integration

During product integration, interface management activities would support the review of integration and assembly procedures to ensure interfaces are properly marked and compatible with specifications and interface control documents. The interface management process has a close relationship to verification and validation. Interface control documentation and approved interface requirement changes are used as inputs to the Product Verification Process and the Product Validation Process, particularly where verification test constraints and interface parameters are needed to set the test objectives and test plans. Interface requirements verification is a critical aspect of the overall system verification.

6.3.1.2.4 Conduct Interface Control

Typically, an Interface Working Group (IWG) establishes communication links between those responsible for interfacing systems, end products, enabling products, and subsystems. The IWG has the responsibility to ensure accomplishment of the planning, scheduling, and execution of all interface activities. An IWG is typically a technical team with appropriate technical membership from the interfacing parties (e.g., the project, the contractor, etc.). The IWG may work independently or as a part of a larger change control board.

6.3.1.2.5 Capture Work Products

Work products include the strategy and procedures for conducting interface management, rationale for interface decisions made, assumptions made in approving or denying an interface change, actions taken to correct identified interface anomalies, lessons learned and updated support and interface agreement documentation.

6.3.1.3 Outputs

Typical outputs needed to capture interface management would include:

- **Interface control documentation**. This is the documentation that identifies and captures the interface information and the approved interface change requests. Types of interface documentation include the Interface Requirements Document (IRD), Interface Control Document/Drawing (ICD), Interface Definition Document (IDD), and Interface Control Plan (ICP). These outputs will then be maintained and approved using the Configuration Management Process and become a part of the overall technical data package for the project.
- **Approved interface requirement changes.** After the interface requirements have been baselined, the Requirements Management Process should be used to identify the need for changes, evaluate the impact of the proposed change, document the final approval/disapproval, and update the requirements documentation/tool/database. For interfaces that require approval from all sides, unanimous approval is required. Changing interface requirements late in the design or implementation life cycle is more likely to have a significant impact on the cost, schedule, or technical design/operations.
- **Other work products.** These work products include the strategy and procedures for conducting interface management, the rationale for interface decisions made, the assumption made in approving or denying an interface change, the actions taken to correct identified interface anomalies, the lessons learned in performing the interface management activities, and the updated support and interface agreement documentation.

6.3.2 Interface Management Guidance

6.3.2.1 Interface Requirements Document

An interface requirement defines the functional, performance, environmental, human, and physical requirements and constraints that exist at a common boundary between two or more functions, system elements, configuration items, or systems. Interface requirements include logical, cognitive, and physical interfaces. They include, as necessary, physical measurements, definitions of sequences of energy or information transfer, and all other significant interactions between items. For example, communication interfaces involve the movement and transfer of data and information within the system, and between the system and its environment. Proper evaluation of communications requirements involves definition of the technical characteristics (e.g., bandwidth, data rate, distribution, etc.) and definition of how humans will interface with the communications and content requirements (what data/information is being communicated, what is being moved among the system components, and the criticality of this information to system functionality).

Interface requirements can be derived from the functional allocation if function inputs and outputs have been defined. For example, as shown in Figure 6.3-2:

- If function F1 outputs item A to function F2, and
- Function F1 is allocated to component C1, and
- Function F2 is allocated to component C2,

- Then there is an implicit requirement that the interface between components C1 and C2 pass item A, whether item A is a liquid, a solid, or a message containing data, etc.

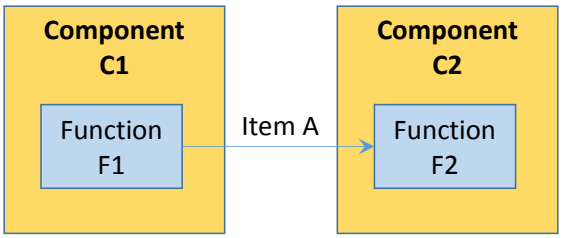


Figure 6.3-2 Deriving Interface Requirements from Functional Allocations

The IRD is a document or model that defines all physical, functional, performance and environmental interface requirements (written in "shall" statements) between two or more end items, elements, or components of a system and ensures project hardware and software compatibility. An IRD is composed of physical and functional requirements and constraints imposed on hardware configuration items and/or software configuration items. The purpose of the IRD is to control the interfaces between interrelated components of the system under development, as well as between the system under development and any external systems (either existing or under development), including humans, that comprise a total architecture. Interface requirements may be contained in the SRD until the point in the development process where the individual interfaces are determined. IRDs are useful when separate organizations are developing components of the system or when the system should levy requirements on other systems outside program/project control. During both Phase A and Phase B, multiple IRDs are drafted for different levels of interfaces. By SRR, draft IRDs would be complete for system-to-external-system interfaces (e.g., the shuttle to the International Space Station), and segment-to-segment interfaces (e.g., the shuttle to the launch pad). An IRD generic outline is described in appendix L.

6.3.2.2 Interface Control Document or Interface Control Drawing

An interface control document, model, or drawing details the physical interface between two system elements, including the number and types of connectors, electrical parameters, mechanical properties, and environmental constraints. The ICD identifies the design solution to the interface requirement. ICDs are useful when separate organizations are developing design solutions to be adhered to at a particular interface. ICDs are usually agreements between two or more organizations / entities, as opposed to an IDD (described in the next section), which is a one-sided description of an interface. An ICD usually does not contain "shall" statements, rather it is the detailed implementation of the requirements defined in the applicable IRD or other requirements document. For small projects, the IRD and ICD may be combined as a two-part document—requirements ("shall" statements) as defined during the early phases, and later as updated with a description of how they were implemented into the design.

6.3.2.3 Interface Definition Document

An IDD is a unilateral document controlled by the end-item provider, and it basically provides the details of the interface for a design solution that is already established. This document is sometimes referred to as a "one-sided ICD." The user of the IDD is provided information on

connectors, electrical parameters, mechanical properties, environmental constraints, etc., of the existing design. The user should then design the interface of the system to be compatible with the already existing design interface.

6.3.2.4 Interface Control Plan

An ICP should be developed to address the process for controlling identified interfaces and the related interface documentation. Key content for the ICP is the organizations involved and their responsibilities relative to defining or controlling interfaces. The ICP should also address the configuration control forum and mechanisms to implement the change process (e.g., Preliminary Interface Revision Notice (PIRN)/Interface Revision Notice (IRN)) for the documents/models. The ICP should also address how issues that arise will be resolved and how to account for the risks associated with the interfaces. For space flight projects that fall under NPR 7120.5, the approach to controlling interfaces is required at MCR, a preliminary version should be ready at SDR and is baselined at PDR. As the design and interfaces mature after PDR, the ICP may need to be updated. The actual list to the ICDs and who is responsible for them may be kept in the ICP as it matures, or kept in other project documentation or databases.

Typical Interface Management Checklist

- Use the generic outline provided when developing the IRD. Define a “reserved” placeholder if a paragraph or section is not applicable.
- Ensure that there are two or more specifications that are being used to serve as the parent for the IRD specific requirements.
- Ensure that “shall” statements are used to define specific requirements.
- Each organization needs to approve and sign the IRD.
- A control process needs to be established to manage changes to the IRD.
- Corresponding ICDs are developed based upon the requirements in the IRD.
- Confirm connectivity between the interface requirements and the Product Verification and Product Validation Processes.
- Define the SEMP content to address interface management.
- Each major program or project should include an ICP to describe the how and what of interface management products.

6.3.2.5 Interface Management Tasks

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

verification activities, multiple system elements are checked out as integrated subsystems or systems. The following provides more details on these tasks.

6.3.2.5.1 Defining Interfaces

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

6.3.2.5.2 Verifying Interfaces

During the verification process, the systems engineer should ensure that the interfaces of each element of the system or subsystem are controlled and known to the developers. When changes to the interfaces are needed, the changes should at least be evaluated for possible impact on other interfacing elements and then communicated to the affected developers. Although all affected developers are part of the group that makes changes, such changes need to be captured in a readily accessible place so that the current state of the interfaces can be known to all. Typical products include ICDs and exception reports.

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

6.3.2.5.3 Inspecting and Acknowledging System and Subsystem Element Receipt

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

6.3.2.5.4 Verifying Element Interfaces

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

6.3.2.5.5 Final Integration and Verification

The elements of the system should be assembled in accordance with the established integration strategy to ensure that the assembly of the system elements into larger or more complex

assemblies is conducted in accordance with the planned strategy. To ensure that the integration has been completed, the integrated system interfaces should be verified and validated. Typical products include integration reports, exception reports, and an integrated system.

6.4 Technical Risk Management

The Technical Risk Management Process is one of the crosscutting technical management processes. Risk is the potential for performance shortfalls, which may be realized in the future, with respect to achieving explicitly established and stated performance requirements. The performance shortfalls may be related to institutional support for mission execution or related to any one or more of the following mission execution domains:

- Safety
- Technical
- Cost
- Schedule

Systems engineers are involved in this process to help identify potential technical risks, develop mitigation plans, monitor progress of the technical effort to determine if new risks arise or old risks can be retired, and to be available to answer questions and resolve issues. The following is guidance in implementation of risk management in general. Thus, when implementing risk management on any given program/project, the responsible systems engineer should direct the effort accordingly. This may involve more or less rigor and formality than that specified in governing documents such as NPRs. Of course, if deviating from NPR "requirements," the responsible engineer must follow the deviation approval process. The idea is to tailor the risk management process so that it meets the needs of the individual program/project being executed while working within the bounds of the governing documentation (e.g., NPRs). For detailed information on the Risk Management Process, refer to the *NASA Risk Management Handbook (NASA/SP-2011-3422).*

Risk is characterized by three basic components:

1. The scenario(s) leading to degraded performance with respect to one or more performance measures (e.g., scenarios leading to injury, fatality, destruction of key assets; scenarios leading to exceedance of mass limits; scenarios leading to cost overruns; scenarios leading to schedule slippage);
2. The likelihood(s) (qualitative or quantitative) of those scenario(s); and
3. The consequence(s) (qualitative or quantitative severity of the performance degradation) that would result if the scenario(s) was (were) to occur.

Uncertainties are included in the evaluation of likelihoods and consequences.

Scenarios begin with a set of initiating events that cause the activity to depart from its intended state. For each initiating event, other events that are relevant to the evolution of the scenario may (or may not) occur and may have either a mitigating or exacerbating effect on the scenario progression. The frequencies of scenarios with undesired consequences are determined. Finally, the multitude of such scenarios is put together, with an understanding of the uncertainties, to create the risk profile of the system.

This “risk triplet” conceptualization of risk is illustrated in Figures 6.4-1 and 6.4-2.

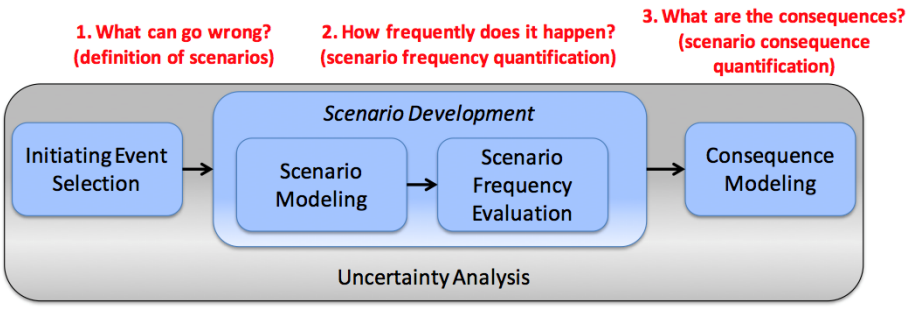


Figure 6.4-1 Risk Scenario Development *(Source: NASA/SP-2011-3421)*

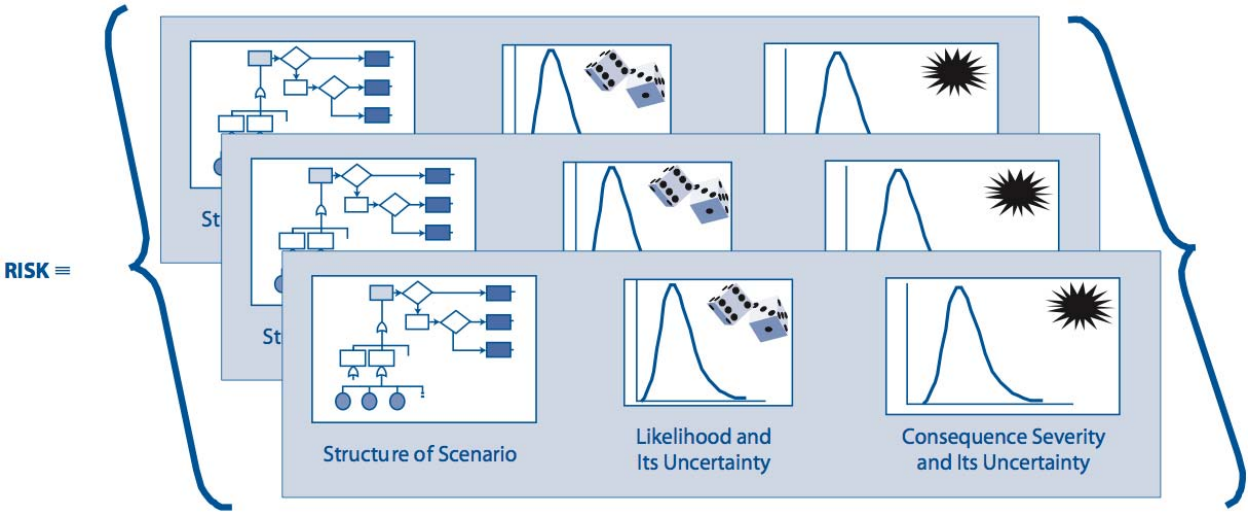


Figure 6.4-2 Risk as an Aggregate Set of Risk Triplets

Undesired scenario(s) might come from technical or programmatic sources (e.g., a cost overrun, schedule slippage, safety mishap, health problem, malicious activities, environmental impact, or failure to achieve a needed scientific or technological objective or success criterion). Both the likelihood and consequences may have associated uncertainties.

- Key Concepts in Risk Management **Risk:** Risk is the potential for shortfalls, which may be realized in the future with respect to achieving explicitly-stated requirements. The performance shortfalls may be related to institutional support for mission execution, or related to any one or more of the following mission execution domains: safety, technical, cost, schedule. Risk is characterized as a set of triplets:
  - The scenario(s) leading to degraded performance in one or more performance measures.
  - The likelihood(s) of those scenarios.
  - The consequence(s), impact, or severity of the impact on performance that would result if those scenarios were to occur.

  Uncertainties are included in the evaluation of likelihoods and consequences.
- **Cost Risk:** This is the risk associated with the ability of the program/project to achieve its life-cycle cost objectives and secure appropriate funding. Two risk areas bearing on cost are (1) the risk that the cost estimates and objectives are not accurate and reasonable; and (2) the risk that program execution will not meet the cost objectives as a result of a failure to handle cost, schedule, and performance risks.
- **Schedule Risk:** Schedule risks are those associated with the adequacy of the time estimated and allocated for the development, production, implementation, and operation of the system. Two risk areas bearing on schedule risk are (1) the risk that the schedule estimates and objectives are not realistic and reasonable; and (2) the risk that program execution will fall short of the schedule objectives as a result of failure to handle cost, schedule, or performance risks.
- **Technical Risk:** This is the risk associated with the evolution of the design and the production of the system of interest affecting the level of performance necessary to meet the stakeholder expectations and technical requirements. The design, test, and production processes (process risk) influence the technical risk and the nature of the product as depicted in the various levels of the PBS (product risk).
- **Programmatic Risk:** This is the risk associated with action or inaction from outside the project, over which the project manager has no control, but which may have significant impact on the project. These impacts may manifest themselves in terms of technical, cost, and/or schedule. This includes such activities as: International Traffic in Arms Regulations (ITAR), import/export control, partner agreements with other domestic or foreign organizations, congressional direction or earmarks, Office of Management and Budget (OMB) direction, industrial contractor restructuring, external organizational changes, etc.
- **Scenario:** A sequence of credible events that specifies the evolution of a system or process from a given state to a future state. In the context of risk management, scenarios are used to identify the ways in which a system or process in its current state can evolve to an undesirable state.

6.4.1 Risk Management Process Description

Figure 6.4-3 provides a typical flow diagram for the Risk Management Process and identifies typical inputs, activities, and outputs to consider in addressing risk management.

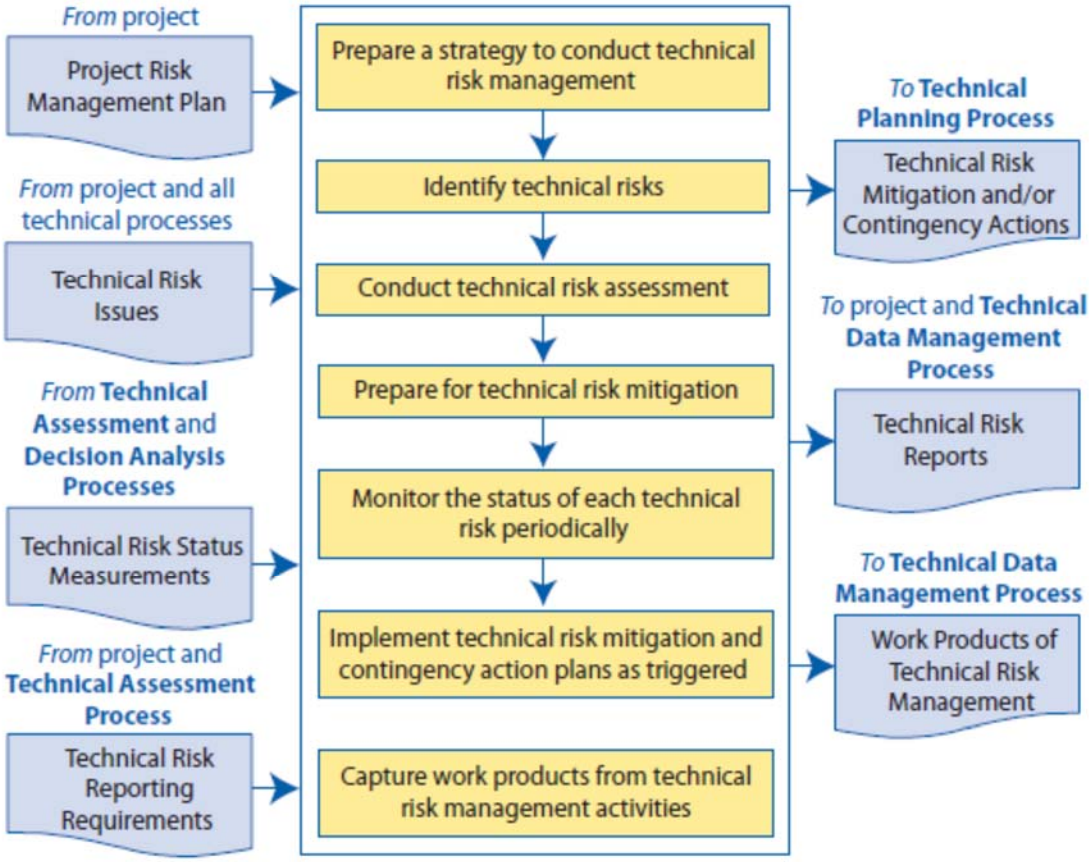


Figure 6.4-3 Risk Management Process

6.4.1.1 Inputs

The following are typical inputs to risk management:

- **Project Risk Management Plan:** The Risk Management Plan is developed under the Technical Planning Process and defines how risk will be identified, mitigated, monitored, and controlled within the project.
- **Technical Risk Issues:** These will be the technical issues identified as the project progresses that pose a risk to the successful accomplishment of the project mission/goals.
- **Technical Risk Status Measurements:** These are any measures that are established that help to monitor and report the status of project technical risks.
- **Technical Risk Reporting Requirements:** Includes requirements of how technical risks will be reported, how often, and to whom.

Additional inputs that may be useful:

- **Other Plans and Policies:** Systems Engineering Management Plan, form of technical data products, and policy input to metrics and thresholds.
- **Technical Inputs:** Stakeholder expectations, concept of operations, imposed constraints, tracked observables, current program baseline, performance requirements, and relevant experience data.

6.4.1.2 Activities

6.4.1.2.1 Prepare a Strategy to Conduct Technical Risk Management

This strategy would include documenting how the program/project risk management plan (as developed during the Technical Planning Process) will be implemented, identifying any additional technical risk sources and categories not captured in the plan, identifying what will trigger actions and how these activities will be communicated to the internal and external teams.

6.4.1.2.2 Identify Technical Risks

On a continuing basis, the technical team will identify technical risks including their source, analyze the potential consequence and likelihood of the risks occurring, and prepare clear risk statements for entry into the program/project risk management system. Coordination with the relevant stakeholders for the identified risks is included. For more information on identifying technical risks, see Section 6.4.2.1.

6.4.1.2.3 Conduct Technical Risk Assessment

Until recently, NASA's Risk Management (RM) approach was based almost exclusively on Continuous Risk Management (CRM)**,** which stresses the management of individual risk issues during implementation. In December of 2008, NASA revised its RM approach in order to more effectively foster proactive risk management. The new approach, which is outlined in NPR 8000.4, Agency Risk Management Procedural Requirements and further developed in *NASA/SP-2011-3422, NASA Risk Management Handbook*, evolves NASA's risk management to entail two complementary processes: Risk-Informed Decision Making (RIDM) and CRM. RIDM is intended to inform direction-setting systems engineering (SE) decisions (e.g., design decisions) through better use of risk and uncertainty information in selecting alternatives and establishing baseline performance requirements (for additional RIDM technical information, guidance, and process description, see *NASA/SP-2010-576 Version 1, NASA Risk-Informed Decision Making Handbook*).

CRM is then used to manage risks over the course of the development and implementation phases of the life cycle to assure that requirements related to safety, technical, cost, and schedule are met. In the past, RM was considered equivalent to the CRM process; now, RM is defined as comprising both the RIDM and CRM processes, which work together to assure proactive risk management as NASA programs and projects are conceived, developed, and executed. Figure 6.4-4 illustrates the concept.

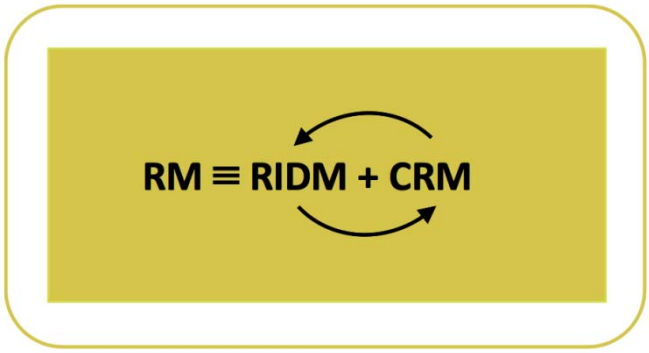


Figure 6.4-4 Risk Management as the Interaction of Risk-Informed Decision Making and Continuous Risk Management (*Source: NASA/SP-2011-3422*)

6.4.1.2.4 Prepare for Technical Risk Mitigation

This includes selecting the risks that will be mitigated and more closely monitored, identifying the risk level or threshold that will trigger a risk mitigation action plan, and identifying for each risk which stakeholders will need to be informed that a mitigation/contingency action is determined as well as which organizations will need to become involved to perform the mitigation/contingency action.

6.4.1.2.5 Monitor the Status of Each Technical Risk Periodically

Risk status will need to be monitored periodically at a frequency identified in the risk plan. Risks that are approaching the trigger thresholds will be monitored on a more frequent basis. Reports of the status are made to the appropriate program/project management or board for communication and for decisions whether to trigger a mitigation action early. Risk status will also be reported at most life-cycle reviews.

6.4.1.2.6 Implement Technical Risk Mitigation and Contingency Action Plans as Triggered

When the applicable thresholds are triggered, the technical risk mitigation and contingency action plans are implemented. This includes monitoring the results of the action plan implementation and modifying them as necessary, continuing the mitigation until the residual risk and/or consequence impacts are acceptable, and communicating the actions and results to the identified stakeholders. Action plan reports are prepared and results reported at appropriate boards and at life-cycle reviews.

6.4.1.2.7 Capture Work Products

Work products include the strategy and procedures for conducting technical risk management; the rationale for decisions made; assumptions made in prioritizing, handling, and reporting technical risks and action plan effectiveness; actions taken to correct action plan implementation anomalies; and lessons learned.

6.4.1.3 Outputs

Following are key risk outputs from activities:

- **Technical Risk Mitigation and/or Contingency Actions:** Actions taken to mitigate identified risks or contingency actions taken in case risks are realized.
- **Technical Risk Reports:** Reports of the technical risk policies, status, remaining residual risks, actions taken, etc. Output at the agreed-to frequency and recipients.
- **Work Products:** Includes the procedures for conducting technical risk management; rationale for decisions made; selected decision alternatives; assumptions made in prioritizing, handling, and reporting technical risks; and lessons learned.

6.4.2 Risk Management Process Guidance

For additional guidance on risk management, refer to *NASA/SP-2010-576, NASA RIDM Handbook* and *NASA/SP-2011-3422, NASA Risk Management Handbook*.

6.5 Configuration Management

Configuration management is a management discipline applied over the product's life cycle to provide visibility into and to control changes to performance and functional and physical characteristics. Additionally, according to SAE Electronic Industries Alliance (EIA) 649B, improper configuration management may result in incorrect, ineffective, and/or unsafe products being released. Therefore, in order to protect and ensure the integrity of NASA products, NASA has endorsed the implementation of the five configuration management functions and the associated 37 underlying principles defined within SAE/EIA-649-*2 Configuration Management Requirements for NASA Enterprises*.

Together, these standards address what configuration management activities are to be done, when they are to happen in the product life-cycle, and what planning and resources are required. Configuration management is a key systems engineering practice that, when properly implemented, provides visibility of a true representation of a product and attains the product's integrity by controlling the changes made to the baseline configuration and tracking such changes. Configuration management ensures that the configuration of a product is known and reflected in product information, that any product change is beneficial and is effected without adverse consequences, and that changes are managed.

CM reduces technical risks by ensuring correct product configurations, distinguishes among product versions, ensures consistency between the product and information about the product, and avoids the embarrassment cost of stakeholder dissatisfaction and complaint. In general, NASA adopts the CM principles as defined by *SAE/EIA 649B, Configuration Management Standard*, in addition to implementation as defined by NASA CM professionals and as approved by NASA management.

When applied to the design, fabrication/assembly, system/subsystem testing, integration, and operational and sustaining activities of complex technology items, CM represents the "backbone" of the enterprise structure. It instills discipline and keeps the product attributes and documentation consistent. CM enables all stakeholders in the technical effort, at any given time in the life of a product, to use identical data for development activities and decision-making. CM principles are applied to keep the documentation consistent with the approved product, and to ensure that the product conforms to the functional and physical requirements of the approved design.

6.5.1 Process Description

Figure 6.5-1 provides a typical flow diagram for the Configuration Management Process and identifies typical inputs, outputs, and activities to consider in addressing CM.

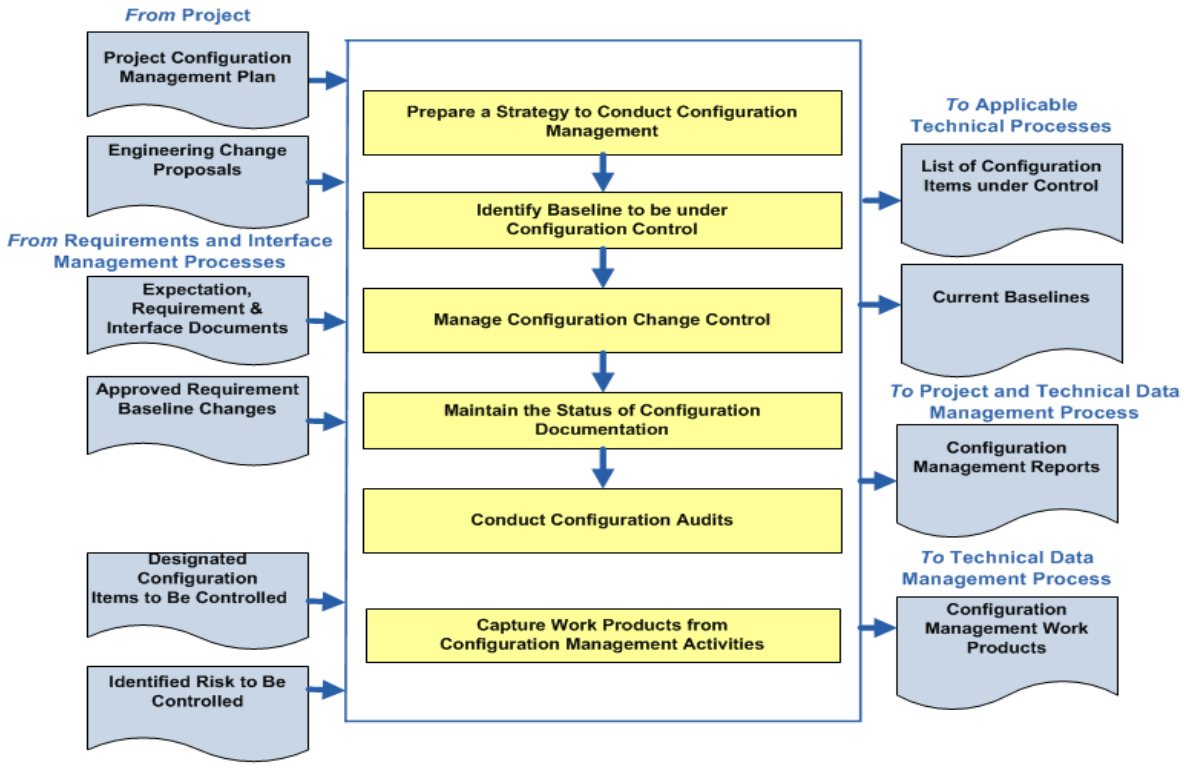


Figure 6.5-1 Configuration Management Process

6.5.1.1 Inputs

The inputs for this process are:

- **CM plan:** This plan would have been developed under the Technical Planning Process and serves as the overall guidance for this process for the program/project
- **Engineering change proposals:** These are the requests for changes to the established baselines in whatever form they may appear throughout the life cycle.
- **Expectation, requirements and interface documents:** These baselined documents or models are key to the design and development of the product.
- **Approved requirements baseline changes:** The approved requests for changes will authorize the update of the associated baselined document or model**.**
- **Designated configuration items to be controlled:** As part of technical planning, a list or philosophy would have been developed that identifies the types of items that will need to be placed under configuration control.

6.5.1.2 Process Activities

There are five elements of CM (see Figure 6.5-2):

- Configuration planning and management
- Configuration identification,

- Configuration change management,
- Configuration Status Accounting (CSA), and
- Configuration verification.

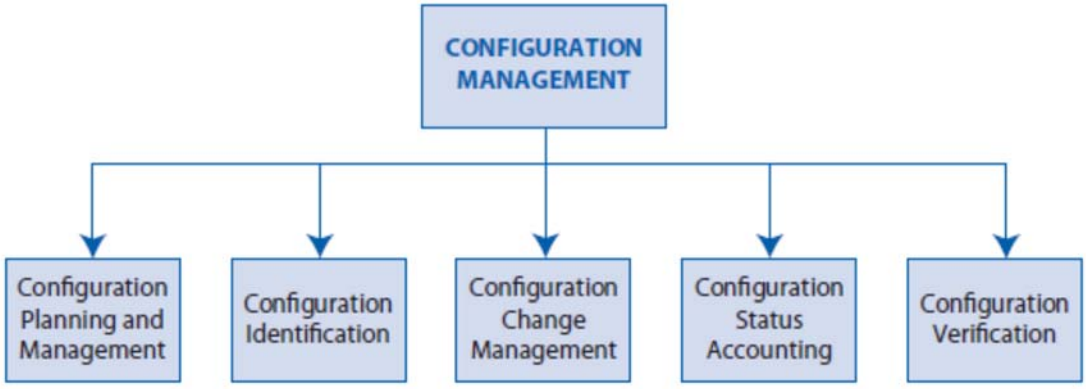


Figure 6.5-2 Five Elements of Configuration Management

6.5.1.2.1 Prepare a Strategy to Conduct CM

CM planning starts at a program's or project's inception. The CM office should carefully weigh the value of prioritizing resources into CM tools or into CM surveillance of the contractors. Reviews by the Center Configuration Management Organization (CMO) are warranted and will cost resources and time, but the correction of systemic CM problems before they erupt into losing configuration control are always preferable to explaining why incorrect or misidentified parts are causing major problems in the program/project.

One of the key inputs to preparing for CM implementation is a strategic plan for the project's complete CM process. This is typically contained in a CM plan. See appendix M for an outline of a typical CM plan.

This plan has both internal and external uses:

- **Internal:** It is used within the program/project office to guide, monitor, and measure the overall CM process. It describes all the CM activities and the schedule for implementing those activities within the program/project.
- **External:** The CM plan is used to communicate the CM process to the contractors involved in the program/project. It establishes consistent CM processes and working relationships.

The CM plan may be a standalone document or it may be combined with other program/project planning documents. It should describe the criteria for each technical baseline creation, technical approvals, and audits.

6.5.1.2.2 Identify Baseline to be Under Configuration Control

Configuration identification is the systematic process of selecting, organizing, and stating the product attributes. Identification requires unique identifiers for a product and its configuration

documentation. The CM activity associated with identification includes selecting the Configuration Items (CIs), determining the CIs' associated configuration documentation, determining the appropriate change control authority, issuing unique identifiers for both CIs and CI documentation, releasing configuration documentation, and establishing configuration baselines.

NASA has four baselines, each of which defines a distinct phase in the evolution of a product design. The baseline identifies an agreed-to description of attributes of a CI at a point in time and provides a known configuration to which changes are addressed. Baselines are established by agreeing to (and documenting) the stated definition of a CI's attributes. The approved "current" baseline defines the basis of the subsequent change. The system specification is typically finalized following the SRR. The functional baseline is established at the SDR and will usually transfer to NASA's control at that time for contracting efforts. For in-house efforts, the baseline is set / controlled by the NASA program/project.

The four baselines (see figure 6.5-3) normally controlled by the program, project, or Center are the following:

- **Functional Baseline:** The functional baseline is the approved configuration documentation that describes a system's or top-level CI's performance requirements (functional, interoperability, and interface characteristics) and the verification required to demonstrate the achievement of those specified characteristics. The functional baseline is established at the SDR by the NASA program/project. The program/project will direct through contractual agreements, how the functional baselines are managed at the different functional levels. (Levels 1-4)
- **Allocated Baseline:** The allocated baseline is the approved performance-oriented configuration documentation for a CI to be developed that describes the functional, performance, and interface characteristics that are allocated from a higher level requirements document or a CI and the verification required to demonstrate achievement of those specified characteristics. The allocated baseline extends the top-level performance requirements of the functional baseline to sufficient detail for defining the functional and performance characteristics and for initiating detailed design for a CI. The allocated baseline is usually controlled by the design organization until all design requirements have been verified. The allocated baseline is typically established at the successful completion of the PDR. Prior to CDR, NASA normally reviews design output for conformance to design requirements through incremental deliveries of engineering data. NASA control of the allocated baseline occurs through review of the engineering deliveries as data items.
- **Product Baseline:** The product baseline is the approved technical documentation that describes the configuration of a CI during the production, fielding/ deployment, and operational support phases of its life cycle. The established product baseline is controlled as described in the configuration management plan that was developed during Phase A. The product baseline is typically established at the completion of the CDR. The product baseline describes:
  - Detailed physical or form, fit, and function characteristics of a CI;
  - The selected functional characteristics designated for production acceptance testing; and

- The production acceptance test requirements.

- **As-Deployed Baseline:** The as-deployed baseline occurs at the ORR. At this point, the design is considered to be functional and ready for flight. All changes will have been incorporated into the documentation.

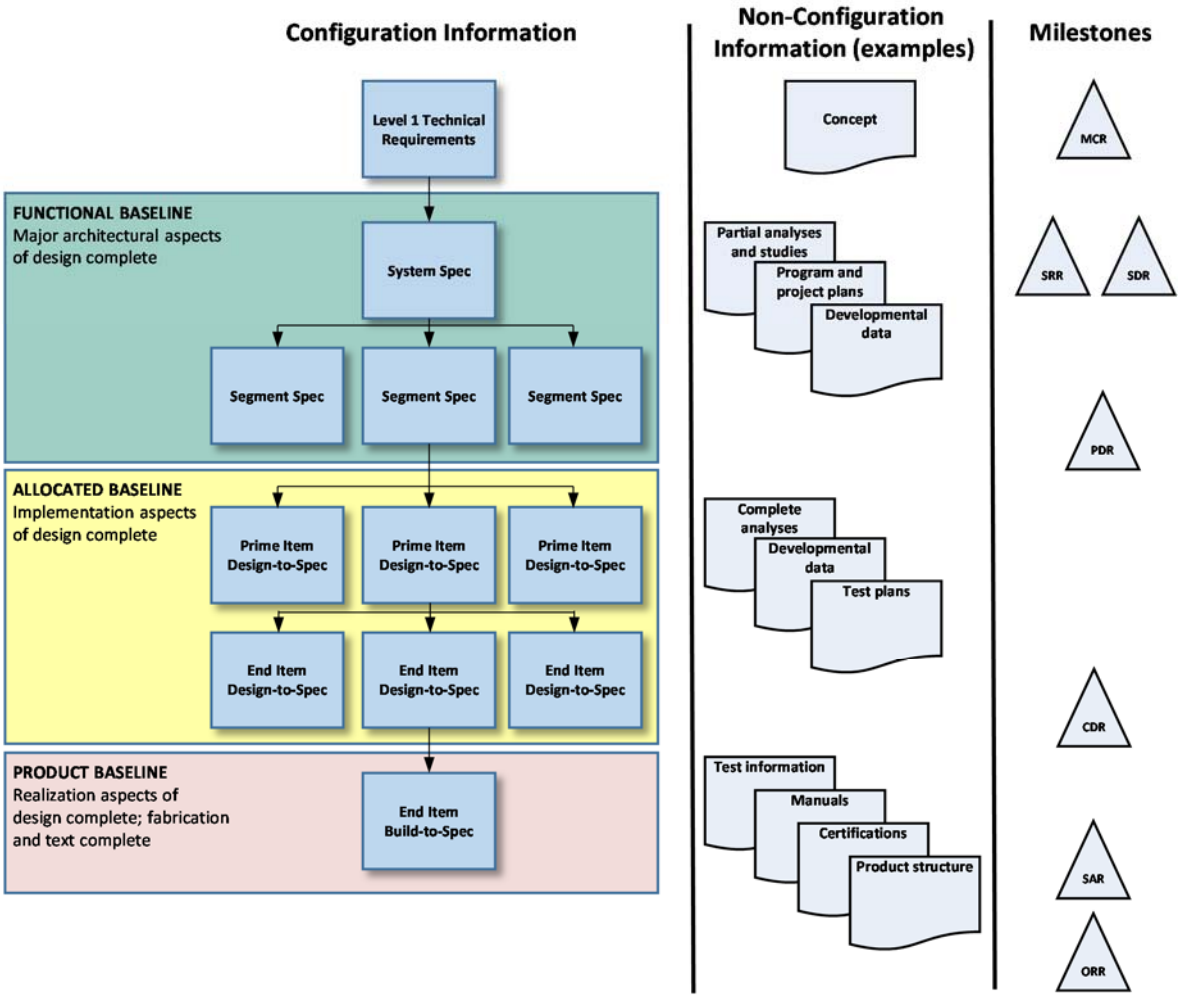


Figure 6.5-3 Evolution of Technical Baseline

6.5.1.2.3 Manage Configuration Change Control

Configuration change management is a process to manage approved designs and the implementation of approved changes. Configuration change management is achieved through the systematic proposal, justification, and evaluation of proposed changes followed by incorporation of approved changes and verification of implementation. Implementing configuration change management in a given program/project requires unique knowledge of the program/project objectives and requirements. The first step establishes a robust and well-disciplined internal NASA Configuration Control Board (CCB) system, which is chaired by someone with program/project change authority. CCB members represent the stakeholders with authority to commit the team they represent. The second step creates configuration change management surveillance of the contractor's activity. The CM office advises the NASA program or project

manager to achieve a balanced configuration change management implementation that suits the unique program/project situation. See Figure 6.5-4 for an example of a typical configuration change management control process.

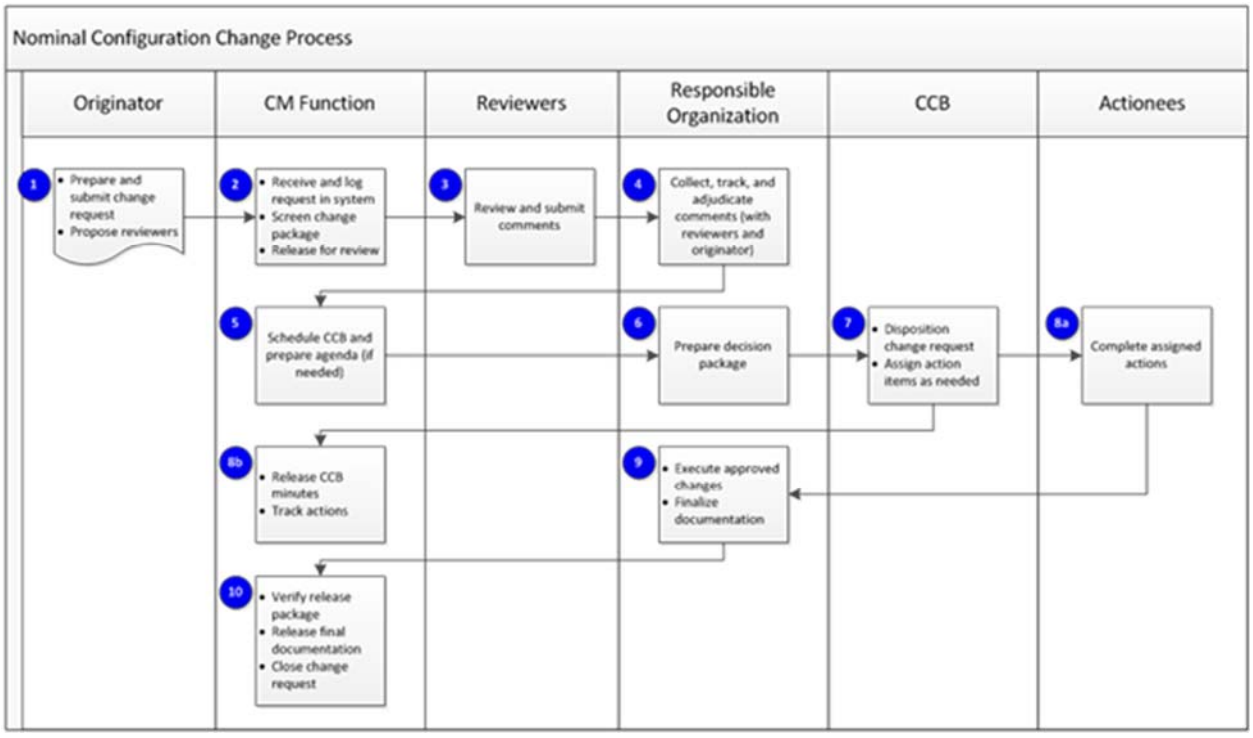


Figure 6.5-4 Typical Change Control Process

Types of Configuration Management Changes

- **Engineering Change:** An engineering change is an iteration in the baseline. Changes can be major or minor. They may or may not include a specification change. Changes affecting an external interface must be coordinated and approved by all stakeholders affected.
  - A "major" change is a change to the baseline configuration documentation that has significant impact (i.e., requires retrofit of delivered products or affects the baseline specification, cost, safety, compatibility with interfacing products, or operator, or maintenance training).
  - A "minor" change corrects or modifies configuration documentation or processes without impact to the interchangeability of products or system elements in the system structure.
- **Waiver:** A waiver is a documented agreement intentionally releasing a program or project from meeting a requirement. (Some Centers use deviations prior to Implementation and waivers

6.5.1.2.4 Maintain the Status of Configuration Documentation

Configuration Status Accounting (CSA) is the recording and reporting of configuration data necessary to manage CIs effectively. An effective CSA system provides timely and accurate configuration information such as:

- Complete current and historical configuration documentation and unique identifiers.
- Status of proposed changes, deviations, and waivers from initiation to implementation.
- Status and final disposition of identified discrepancies and actions identified during each configuration audit.

Some useful purposes of the CSA data include:

- An aid for proposed change evaluations, change decisions, investigations of design problems, warranties, and shelf-life calculations.
- Historical traceability.
- Software trouble reporting.
- Performance measurement data.

The following are critical functions or attributes to consider if designing or purchasing software to assist with the task of managing configuration.

- Ability to share data real time with internal and external stakeholders securely;
- Version control and comparison (track history of an object or product);
- Secure user checkout and check in;
- Tracking capabilities for gathering metrics (i.e., time, date, who, time in phases, etc.);
- Web based;
- Notification capability via e-mail;
- Integration with other databases or legacy systems;
- Compatible with required support contractors and/or suppliers (i.e., can accept data from a third party as required);
- Integration with drafting and modeling programs as required;
- Provide neutral format viewer for users;
- License agreement allows for multiple users within an agreed-to number;
- Workflow and life-cycle management;
- Limited customization;
- Migration support for software upgrades;
- User friendly;
- Consideration for users with limited access;
- Ability to attach standard format files from desktop
- Workflow capability (i.e., route a CI as required based on a specific set of criteria); and
- Capable of acting as the one and only source for released information.

6.4.1.2.5 Conduct Configuration Audits

Configuration verification is accomplished by inspecting documents, products, and records; reviewing procedures, processes, and systems of operations to verify that the product has achieved its required performance requirements and functional attributes; and verifying that the product's design is documented. This is sometimes divided into functional and physical configuration audits. (See Section 6.7.2.4.2 for more on technical reviews.)

6.4.1.2.6 Capture work Products

These include the strategy and procedures for configuration management, the list of identified configuration items, descriptions of the configuration items, change requests, disposition of the requests, rational for dispositions, reports, and audit results.

6.5.1.3 Outputs

NPR 7120.5 defines a project's life cycle in progressive phases. Beginning with Pre-Phase A, these steps in turn are grouped under the headings of Formulation and Implementation. Approval is required to transition between these phases. Key Decision Points (KDPs) define transitions between the phases. CM plays an important role in determining whether a KDP has been met. Major outputs of CM are:

- **List of configuration items under control (Configuration Status Accounting (CSA) reports):** This output is the list of all the items, documents, hardware, software, models, etc., that were identified as needing to be placed under configuration control. CSA reports are updated and maintained throughout the program and project life cycle.
- **Current baselines:** Baselines of the current configurations of all items that are on the CM list are made available to all technical teams and stakeholders.
- **CM reports:** Periodic reports on the status of the CM items should be available to all stakeholders on an agreed-to frequency and at key life-cycle reviews.
- **Other CM work products:** Other work products include the strategy and procedures used for CM; descriptions, drawings and/or models of the CM items; change requests and their disposition and accompanying rationale; reports; audit results as well as any corrective actions needed.

6.5.2 CM Guidance

6.5.2.1 What Is the Impact of Not Doing CM?

The impact of not doing CM may result in a project being plagued by confusion, inaccuracies, low productivity, and unmanageable configuration data. During the Columbia accident investigation, the Columbia Accident Investigation Board (CAIB) found inconsistencies related to the hardware and the documentation with "unincorporated documentation changes" that led to failure. No CM issues were cited as a cause of the accident. The usual impact of not implementing CM can be described as "losing configuration control." Within NASA, this has resulted in program/project delays and engineering issues, especially in fast prototyping developments (X-37 Program) where schedule has priority over recording what is being done to

the hardware. If CM is implemented properly, discrepancies identified during functional and physical configuration audits will be addressed. The following impacts are possible and have occurred in the past:

- Mission failure and loss of property and life due to improperly configured or installed hardware or software,
- Mission failure to gather mission data due to improperly configured or installed hardware or software,
- Significant mission delay incurring additional cost due to improperly configured or installed hardware or software, and
- Significant mission costs or delay due to improperly certified parts or subsystems due to fraudulent verification data.

If CM is not implemented properly, problems may occur in manufacturing, quality, receiving, procurement, etc. The user will also experience problems if ILS data are not maintained. Using a shared software system that can route and track tasks provides the team with the resources necessary for a successful project.

Warning Signs/Red Flags (How Do You Know When You're in Trouble?)

General warning signs of an improper implementation of CM include the following:

- Failure of program to define the "top-level" technical requirement ("We don't need a specification").
- Failure of program to recognize the baseline activities that precede and follow design reviews.
- Program office reduces the time to evaluate changes to one that is impossible for engineering, SMA, or other CCB members to meet.
- Program office declares "there will be no dissent in the record" for CCB documentation.
- Contract is awarded without CM requirements concurred with by CMO supporting the program office.
- Redlines used inappropriately on production floor to keep track of changes to design.
- Material Review Board does not know the difference between critical, major, and minor non-conformances and the appropriate classification of waivers.
- Drawings are not of high quality and do not contain appropriate notes to identify critical engineering items for configuration control or appropriate tolerancing.
- Vendors do not understand the implication of submitting waivers to safety requirements as defined in engineering.
- Subcontractors/vendors change engineering design without approval of integrating contractor, do not know how to coordinate and write an engineering change request, etc.
- Manufacturing tooling engineering does not keep up with engineering changes that affect tooling concepts. Manufacturing tools lose configuration control and acceptability for production.
- Verification data cannot be traced to released part number and specification that apply to verification task.
- Operational manuals and repair instructions cannot be traced to latest released part number and repair drawings that apply to repair/modification task.
- Maintenance and ground support tools and equipment cannot be traced to latest released part number and specification that applies to equipment.
- Parts and items cannot be identified due to improper identification markings.
- Digital closeout photography cannot be correlated to the latest released engineering.
- NASA is unable to verify the latest released engineering through access to the contractor's CM Web site.
- Tools required per installation procedures do not match the fasteners and nuts and bolts used in the design of CIs.
- CIs do not fit into their packing crates and containers due to losing configuration control in the design of the shipping and packing containers.
- Supporting procurement/fabrication change procedures do not adequately involve approval by

6.5.2.2 When Is It Acceptable to Use Redline Drawings?

"Redline" refers to the control process of marking up drawings and documents during design, fabrication, production, and testing that are found to contain errors or inaccuracies. Work stoppages could occur if the documents were corrected through the formal change process.

All redlines require the approval of the responsible hardware manager and quality assurance manager at a minimum. The program/project will determine whether redlines are to be incorporated into the plan or procedure.

The important point is that each project should have a controlled procedure for redlines that specifies redline procedures and approvals.

Redlines Identified as One of the Major Causes of the NOAA N Prime Mishap

Excerpts from the NOAA N-Prime Mishap Investigation Final Report:

"Several elements contributed to the NOAA N-PRIME incident, the most significant of which were the lack of proper TOC [Turn Over Cart] verification, including the lack of proper PA [Product Assurance] witness, the change in schedule and its effect on the crew makeup, the failure of the crew to recognize missing bolts while performing the interface surface wipe down, the failure to notify in a timely fashion or at all the Safety, PA, and Government representatives, and the improper use of procedure redlines leading to a difficult-to-follow sequence of events. The interplay of the several elements allowed a situation to exist where the extensively experienced crew was not focusing on the activity at hand. There were missed opportunities that could have averted this mishap.

"In addition, the operations team was utilizing a heavily redlined procedure that required considerable 'jumping' from step to step, and had not been previously practiced. The poorly written procedure and novel redlines were preconditions to the decision errors made by the RTE [Responsible Test Engineer].

"The I&T [Integration and Test] supervisors allowed routine poor test documentation and routine misuse of procedure redlines.

"Key processes that were found to be inadequate include those that regulate operational tempo, operations planning, procedure development, use of redlines, and GSE [Ground Support Equipment] configurations. For instance, the operation during which the mishap occurred was conducted using extensively redlined procedures. The procedures were essentially new at the time of the operation—that is, they had never been used in that particular instantiation in any prior operation. The rewritten procedure had been approved through the appropriate channels even

6.6 Technical Data Management

The Technical Data Management Process is used to plan for, acquire, access, manage, protect, and use data of a technical nature to support the total life cycle of a system. Data Management (DM) includes the development, deployment, operations and support, eventual retirement, and retention of appropriate technical, to include mission and science, data beyond system retirement as required by NPR 1441.1, NASA Records Retention Schedules.

DM is illustrated in Figure 6.6-1. Key aspects of DM for systems engineering include:

- Application of policies and procedures for data identification and control,
- Timely and economical acquisition of technical data,
- Assurance of the adequacy of data and its protection,
- Facilitating access to and distribution of the data to the point of use,
- Analysis of data use,
- Evaluation of data for future value to other programs/projects, and
- Process access to information written in legacy software.

The Technical Data Management and Configuration Management Processes work side-by-side to ensure all information about the project is safe, known, and accessible. Changes to information under configuration control require a Change Request (CR) and are typically approved by a Configuration Control Board. Changes to information under Technical Data Management do not need a CR but still need to be managed by identifying who can make changes to each type of technical data.

6.6.1 Process Description

Figure 6.6-1 provides a typical flow diagram for the Technical Data Management Process and identifies typical inputs, outputs, and activities to consider in addressing technical data management.

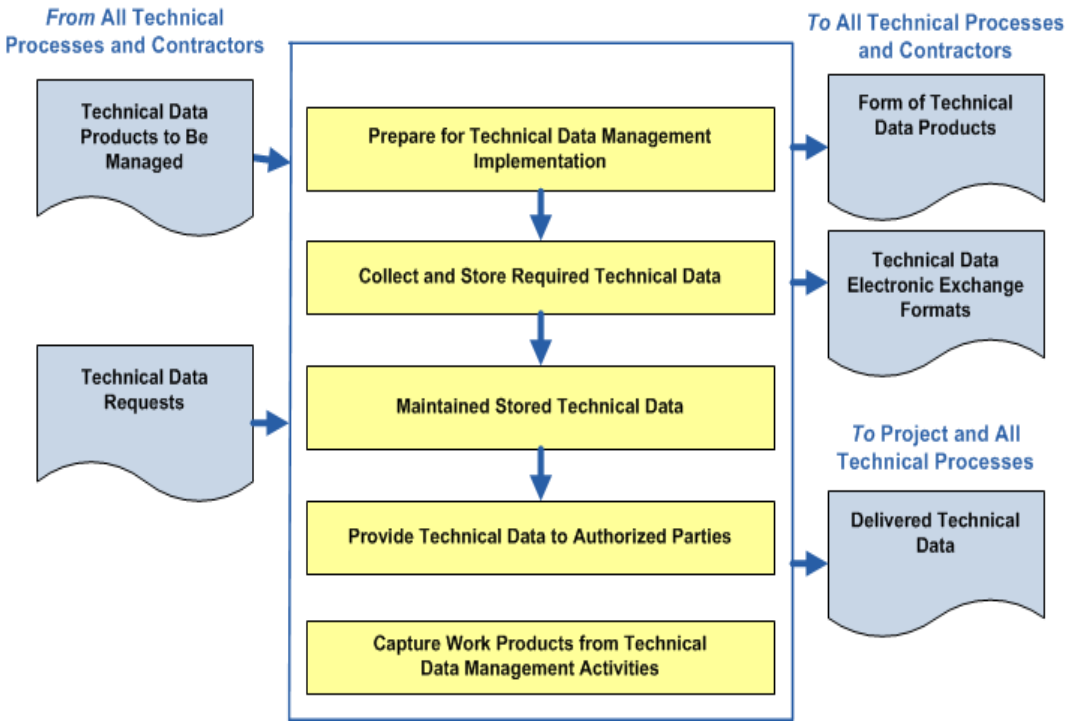


Figure 6.6-1 Technical Data Management Process

6.6.1.1 Inputs

The inputs for this process are:

- **Technical data products to be managed:** Technical data, regardless of the form or method of recording and whether the data are generated by the contractor or Government during the life cycle of the system being developed. (Electronic technical data should be stored with sufficient metadata to enable easy retrieval and sorting.)
- **Technical data requests:** External or internal requests for any of the technical data generated by the program/project.

6.6.1.2 Process Activities

Each Center is responsible for policies and procedures for technical data management. All space flight programs and projects need to manage authoritative data associated with a product throughout its life cycle. NPR 7120.5 and NPR 7123.1 define the need to manage data, but leave specifics to the individual Centers. However, NPR 7120.5 does require that DM planning be provided as either a section in the program/project plan, CM plan, or as a separate document. The program or project manager is responsible for ensuring that the data required are captured and stored, data integrity is maintained, and data are disseminated as required.

Other NASA policies address the acquisition and storage of data and not just the technical data used in the life cycle of a system.

6.6.1.2.1 Role of Data Management Plan

The recommended procedure is that the DM plan be a separate plan apart from the program/project plan. DM issues are usually of sufficient magnitude to justify a separate plan. The plan should cover the following major DM topics:

- Identification/definition/management of data sets.
- Control procedures—receipt, modification, review, and approval.
- Guidance on how to access/search for data for users.
- Data exchange formats that promote data reuse and help to ensure that data can be used consistently throughout the system, family of systems, or system of systems.
- Data rights and distribution limitations such as export-control Sensitive But Unclassified (SBU).
- Storage and maintenance of data, including master lists where documents and records are maintained and managed.

6.6.1.2.2 Technical Data Management Key Considerations

Subsequent activities collect, store, and maintain technical data and provide it to authorized parties as required. Some considerations that impact these activities for implementing Technical Data Management include:

- Requirements relating to the flow/delivery of data to or from a contractor should be specified in the technical data management plan and included in the Request for Proposal (RFP) and contractor agreement.
- NASA should not impose changes on existing contractor data management systems unless the program/project technical data management requirements, including data exchange requirements, cannot otherwise be met.
- Responsibility for data inputs into the technical data management system lies solely with the originator or generator of the data.
- The availability/access of technical data lies with the author, originator, or generator of the data in conjunction with the manager of the technical data management system.
- The established availability/access description and list should be baselined and placed under configuration control.
- For new programs/projects, a digital generation and delivery medium is desired. Existing programs/projects should weigh the cost/benefit trades of digitizing hard copy data.

6.6.1.2.3 General Data Management Roles

The Technical Data Management Process provides the basis for applying the policies and procedures to identify and control data requirements; to responsively and economically acquire, access, and distribute data; and to analyze data use.

Adherence to DM principles/rules enables the sharing, integration, and management of data for performing technical efforts by Government and industry, and ensures that information generated from managed technical data satisfies requests or meets expectations.

The Technical Data Management Process has a leading role in capturing and organizing technical data and providing information for the following uses:

- Identifying, gathering, storing, and maintaining the work products generated by other systems engineering technical and technical management processes as well as the assumptions made in arriving at those work products;
- Enabling collaboration and life-cycle use of system product data;
- Capturing and organizing technical effort inputs, as well as current, intermediate, and final outputs;
- Data correlation and traceability among requirements, designs, solutions, decisions, and rationales;
- Documenting engineering decisions, including procedures, methods, results, and analyses;
- Facilitating technology insertion for affordability improvements during re-procurement and post-production support; and
- Supporting other technical management and technical processes, as needed.

6.6.1.2.4 Data Identification/Definition

Each program/project determines data needs during the life cycle. Data types may be defined in standard documents. Center and Agency directives sometimes specify content of documents and are appropriately used for in-house data preparation. The standard description is modified to suit program/project-specific needs, and appropriate language is included in SOWs to implement actions resulting from the data evaluation. "Data suppliers" may be contractors, academia, or the Government. Procurement of data from an outside supplier is a formal procurement action that requires a procurement document; in-house requirements may be handled using a less formal method. Below are the different types of data that might be utilized within a program/ project:

- **Data**
  - "Data" is defined in general as "recorded information regardless of the form or method of recording." However, the terms "data" and "information" are frequently used interchangeably. To be more precise, data generally should be processed in some manner to generate useful, actionable information.
  - "Data," as used in SE DM, includes technical data; computer software documentation; and representation of facts, numbers, or data of any nature that can be communicated, stored, and processed to form information required by a contract or agreement to be delivered to, or accessed by, the Government.
  - Data include that associated with system development, modeling and simulation used in development or test, test and evaluation, installation, parts, spares, repairs, usage data required for product sustainability, and source and/or supplier data.

- Data specifically not included in Technical Data Management would be data relating to general NASA workforce operations information, communications information (except where related to a specific requirement),financial transactions, personnel data, transactional data, and other data of a purely business nature.

- **Data Call**: Solicitation from Government stakeholders (specifically Integrated Product Team (IPT) leads and functional managers) identifies and justifies their data requirements from a proposed contracted procurement. Since data provided by contractors have a cost to the Government, a data call (or an equivalent activity) is a common control mechanism used to ensure that the requested data are truly needed. If approved by the data call, a description of each data item needed is then developed and placed on contract.
- **Information:** Information is generally considered as processed data. The form of the processed data is dependent on the documentation, report, review formats, or templates that are applicable.
- **Technical Data Package:** A technical data package is a technical description of an item adequate for supporting an acquisition strategy, production, engineering, and logistics support. The package defines the required design configuration and procedures to ensure adequacy of item performance. It consists of all applicable items such as drawings, associated lists, specifications, standards, performance requirements, quality assurance provisions, and packaging details.
- **Technical Data Management System**: The strategies, plans, procedures, tools, people, data formats, data exchange rules, databases, and other entities and descriptions required to manage the technical data of a program/project.

Inappropriate Uses of Technical Data

Examples of inappropriate uses of technical data include:

- Unauthorized disclosure of classified data or data otherwise provided in confidence;
- Faulty interpretation based on incomplete, out-of-context, or otherwise misleading data; and
- Use of data for parts or maintenance procurement for which at least Government purpose rights have not been obtained.

Ways to help prevent inappropriate use of technical data include the following:

- Educate stakeholders on appropriate data use; and
- Control access to sensitive data.

6.6.1.2.5 Initial Data Management System Structure

When setting up a DM system, it is not necessary to acquire (that is, to purchase and take delivery of) all technical data generated on a project. Some data may be stored in other locations with accessibility provided on a need-to-know basis. Data should be purchased only when such access is not sufficient, timely, or secure enough to provide for responsive life-cycle planning and system maintenance. Data calls are a common control mechanism to help address this need.

6.6.1.2.6 Data Management Planning

- Prepare a technical data management strategy. This strategy can document how the program / project data management plan will be implemented by the technical effort or, in the absence of such a program-level plan, be used as the basis for preparing a detailed technical data management plan, including:
  - Items of data that will be managed according to program/project or organizational policy, agreements, or legislation;
  - The data content and format;
  - A framework for data flow within the program/project and to/from contractors including the language(s) to be employed in technical effort information exchanges;
  - Technical data management responsibilities and authorities regarding the origin, generation, capture, archiving, security, privacy, and disposal of data products;
  - Establishing the rights, obligations, and commitments regarding the retention of, transmission of, and access to data items; and
  - Relevant data storage, transformation, transmission, and presentation standards and conventions to be used according to program/project or organizational policy, agreements, or legislative constraints.
- Obtain strategy/plan commitment from relevant stakeholders.
- Prepare procedures for implementing the technical data management strategy for the technical effort and/or for implementing the activities of the technical data management plan.
- Establish a technical database(s) to use for technical data maintenance and storage or work with the program/project staff to arrange use of the program/project database(s) for managing technical data.
- Establish data collection tools, as appropriate to the technical data management scope and available resources. (See Section 7.3.)
- Establish electronic data exchange interfaces in accordance with international standards / agreements and applicable NASA standards.
- Train appropriate stakeholders and other technical personnel in the established technical data management strategy/plan, procedures, and data collection tools, as applicable.
- Expected outcomes:
  - A strategy and/or plan for implementing technical data management;
  - Established procedures for performing planned technical data management activities;
  - Master list of managed data and its classification by category and use;
  - Data collection tools established and available; and
  - Qualified technical personnel capable of conducting established technical data management procedures and using available data collection tools.

6.6.1.2.7 Key Considerations for Planning Data Management and for Tool Selection

- All data entered into the technical data management system or delivered to a requester from the databases of the system should have traceability to the author, originator, or generator of the data.
- All technical data entered into the technical data management system should carry objective evidence of current status (for approval, for agreement, for information, etc.), version/control number, and date.
- The technical data management approach should be included in the SEMP.
- Technical data expected to be used for re-procurement of parts, maintenance services, etc., might need to be reviewed by the Center's legal counsel.

Careful consideration should be given when planning the data access and storage of data that will be generated from a project or program. If a system or tool is needed, many times the CM tool can be used with less formality. If a separate tool is required to manage the data, refer to the section below for some best practices when evaluating a data management tool. Priority should be placed on being able to access the data and on the ease of inputting the data. The second priority should be consideration of the value of the specific data to current projects/programs, future programs / projects, NASA's overall efficiency, and the uniqueness to NASA's engineering knowledge.

The following are critical functions or attributes to consider when designing or purchasing software to assist with the task of managing data:

- Ability to share data with internal and external stakeholders securely;
- Version control and comparison to track the history of an object or product;
- Secure user updating;
- Access control down to the file level;
- Web-based;
- Ability to link data to CM system or elements;
- Compatibility with required support contractors and/or suppliers, i.e., can accept data from a third party as required;
- Ability to integrate with drafting and modeling programs as required;
- Provision of a neutral format viewer for users;
- License agreement that allows for multiuser seats;
- Workflow and life-cycle management is a suggested option;
- Limited customization;
- Migration support between software version upgrades;
- User friendly;

- Straightforward search capabilities; and
- Ability to attach standard format files from desktop.

Data Collection Checklist

- Have the frequency of collection and the points in the technical and technical management processes when data inputs will be available been determined?
- Has the timeline that is required to move data from the point of origin to storage repositories or stakeholders been established?
- Who is responsible for the input of the data?
- Who is responsible for data storage, retrieval, and security?
- Have necessary supporting tools been developed or acquired?

6.6.1.2.8 Provide Data to Authorized Parties

Storage of engineering data needs to be planned at the beginning of a program or project. Some of the data types will fall under the control of NPR 1441.1, *NASA Records Management Program Requirements*, and therefore will have specified retention requirements; those that do not will have to be addressed. It is best to evaluate all data that will be produced and decide how long it is of value to the program or project or to NASA engineering as a whole. There are four basic questions to ask when evaluating data's value:

- Do the data describe the product/system that is being developed or built?
- Are the data required to accurately produce the product/system being developed or built?
- Do the data offer insight for similar future programs or projects?
- Do the data hold key information that needs to be maintained in NASA's knowledge base for future engineers to use or kept as a learning example?

6.6.1.2.9 Technical Data Capture Tasks

Table 6.6-1 defines the tasks required to capture technical data.

6.6.1.2.10 Protection for Data Deliverables

All data deliverables should include distribution statements and procedures to protect all data that contain critical technology information, as well as to ensure that limited distribution data, intellectual property data, or proprietary data are properly handled during systems engineering activities. This injunction applies whether the data are hard copy or digital.

As part of overall asset protection planning, NASA has established special procedures for the protection of Critical Program Information (CPI). CPI may include components; engineering, design, or manufacturing processes; technologies; system capabilities, and vulnerabilities; and any other information that gives a system its distinctive operational capability.

CPI protection should be a key consideration for the technical data management effort and is part of the asset protection planning process.

Table 6.6-1 Technical Data Tasks

Description	Tasks	Expected Outcomes
Technical data capture	Collect and store inputs and technical effort outcomes from the technical and technical management processes, including: results from technical assessments; descriptions of methods, tools, and metrics used; recommendations, decisions, assumptions, and impacts of technical efforts and decisions; lessons learned; deviations from plan; anomalies and out-of-tolerances relative to requirements; and other data for tracking requirements Perform data integrity checks on collected data to ensure compliance with content and format as well as technical data checks to ensure there are no errors in specifying or recording the data. Report integrity check anomalies or variances to the authors or generators of the data for correction. Prioritize, review, and update data collection and storage procedures as part of regularly scheduled maintenance.	Sharable data needed to perform and control the technical and technical management processes is collected and stored. Stored data inventory.
Technical data maintenance	Implement technical management roles and responsibilities with technical data products received. Manage database(s) to ensure that collected data have proper quality and integrity; and are properly retained, secure, and available to those with access authority. Periodically review technical data management activities to ensure consistency and identify anomalies and variances. Review stored data to ensure completeness, integrity, validity, availability, accuracy, currency, and traceability. Perform technical data maintenance, as required. Identify and document significant issues, their impacts, and changes made to technical data to correct issues and mitigate impacts. Maintain, control, and prevent the stored data from being used inappropriately. Store data in a manner that enables easy and speedy retrieval. Maintain stored data in a manner that protects the technical data against foreseeable hazards, e.g., fire, flood, earthquake, etc.	Records of technical data maintenance. Technical effort data, including captured work products, contractor-delivered documents, and acquirer-provided documents are controlled and maintained. Status of data stored is maintained to include: version description, timeline, and security classification.

Description	Tasks	Expected Outcomes
Technical data/ information distribution	Maintain an information library or reference index to provide technical data availability and access instructions. Receive and evaluate requests to determine data requirements and delivery instructions. Process special requests for technical effort data or information according to established procedures for handling such requests. Ensure that required and requested data are appropriately distributed to satisfy the needs of the acquirer and requesters in accordance with the agreement, program/project directives, and technical data management plans and procedures. Ensure that electronic access rules are followed before database access is allowed or any requested data are electronically released / transferred to the requester. Provide proof of correctness, reliability, and security of technical data provided to internal and external recipients.	Access information (e.g., available data, access means, security procedures, time period for availability, and personnel cleared for access) is readily available. Technical data are provided to authorize requesters in the appropriate format, with the appropriate content, and by a secure mode of delivery, as applicable.
Data management system maintenance	Implement safeguards to ensure protection of the technical database and of *en route* technical data from unauthorized access or intrusion. Establish proof of coherence of the overall technical dataset to facilitate effective and efficient use. Maintain, as applicable, backups of each technical database. Evaluate the technical data management system to identify collection and storage performance issues and problems; satisfaction of data users; risks associated with delayed or corrupted data, unauthorized access, or survivability of information from hazards such as fire, flood, earthquake, etc. Review systematically the technical data management system, including the database capacity, to determine its appropriateness for successive phases of the Defense Acquisition Framework. Recommend improvements for discovered risks and problems: Handle risks identified as part of technical risk management. Control recommended changes through established program / project change management activities.	Current technical data management system. Technical data are appropriately and regularly backed up to prevent data loss.

6.6.1.3 Outputs

Outputs include timely, secure availability of needed data in various representations to those authorized to receive it. Major outputs from the Technical Data Management Process include the following (see Figure 6.6-1):

- **Form of Technical Data Products:** How each type of data is held and stored such as textual, graphic, video, etc.
- **Technical Data Electronic Exchange Formats:** Description and perhaps templates, models or other ways to capture the formats used for the various data exchanges.
- **Delivered Technical Data:** The data that were delivered to the requester.

Other work products generated as part of this process include the strategy and procedures used for technical data management, request dispositions, decisions, and assumptions.

6.6.2 Technical Data Management Guidance

6.6.2.1 Data Security and ITAR

NASA generates an enormous amount of information, much of which is unclassified / nonsensitive in nature with few restrictions on its use and dissemination. NASA also generates and maintains Classified National Security Information (CNSI) under a variety of Agency programs, projects, and through partnerships and collaboration with other Federal agencies, academia, and private enterprises. SBU markings require the author, distributor, and receiver to keep control of the sensitive document and data or pass the control to an established control process. Public release is prohibited, and a document/data marked as such should be transmitted by secure means. Secure means are encrypted e-mail, secure fax, or person-to-person tracking. WebEx is a non-secure environment. Standard e-mail is not permitted to transmit SBU documents and data. Per NID 1600.55, Sensitive But Unclassified (SBU) Controlled Information, a secure way to send SBU information via e-mail is using the Public Key Infrastructure (PKI) to transmit the file(s). PKI is a system that manages keys to lock and unlock computer data. The basic purpose of PKI is to enable you to share your data keys with other people in a secure manner. PKI provides desktop security, as well as security for desktop and network applications, including electronic and Internet commerce.

Data items such as detailed design data (models, drawings, presentations, etc.), limited rights data, source selection data, bid and proposal information, financial data, emergency contingency plans, and restricted computer software are all examples of SBU data. Items that are deemed SBU should be clearly marked in accordance with NID 1600.55, Sensitive But Unclassified (SBU) Controlled Information. Data or items that cannot be directly marked, such as computer models and analyses, should have an attached copy of NASA Form 1686 that indicates the entire package is SBU data. Documents are required to have a NASA Form 1686 as a cover sheet. SBU documents and data should be safeguarded. Some examples of ways to safeguard SBU data are: access is limited on a need-to-know basis, items are copy-controlled, items are attended while being used, items are properly marked (document header, footer, and NASA Form 1686), items are stored in locked containers or offices and secure servers, transmitted by secure means, and destroyed by approved methods (shredding, etc.). For more information on SBU data, see NPR 1600.1, NASA Security Program Procedural Requirements.

The International Traffic in Arms Regulation (ITAR) implements the Arms Export Control Act, and contains the United States Munitions List (USML). The USML lists articles, services, and related technical data that are designated as "defense articles" and "defense services," pursuant to Sections 38 and 47(7) of the Arms Export Control Act. The ITAR is administered by the U.S. Department of State. "Technical data" as defined in the ITAR does not include information concerning general scientific, mathematical, or engineering principles commonly taught in schools, colleges, and universities or information in the public domain (as that term is defined in 22 CFR 120.11). It also does not include basic marketing information on function and purpose or general system descriptions. For purposes of the ITAR, the following definitions apply:

- **"Defense Article" (22 CFR 120.6):** A defense article is any item or technical data on the USML. The term includes technical data recorded or stored in any physical form, models, mockups, or other items that reveal technical data directly relating to items designated in the

USML. Examples of defense articles included on the USML are (1) launch vehicles, including their specifically designed or modified components, parts, accessories, attachments, and associated equipment; (2) remote sensing satellite systems, including ground control stations for telemetry, tracking, and control of such satellites, as well as passive ground stations if such stations employ any cryptographic items controlled on the USML or employ any uplink command capability; and (3) all components, parts, accessories, attachments, and associated equipment(including ground support equipment)that is specifically designed, modified, or configured for such systems. (See 22 CFR 121.1 for the complete listing.)

- **"Technical Data" (22 CFR 120.10):** Technical data are information required for the design, development, production, manufacture, assembly, operation, repair, testing, maintenance, or modification of defense articles. This includes information in the form of blueprints, drawings, photographs, plans, instructions, and documentation.
- **Classified Information Relating to Defense Articles and Defense Services:** Classified information is covered by an invention secrecy order (35 U.S.C. 181; 37 CFR Part 5).
- **Software Directly Related to Defense Articles:** Controlled software includes, but is not limited to, system functional design, logic flow, algorithms, application programs, operating systems, and support software for design, implementation, test, operations, diagnosis, and repair related to defense articles.

6.7 Technical Assessment

Technical assessment is the crosscutting process used to help monitor technical progress of a program/project through periodic technical reviews and through monitoring of technical indicators such as MOEs, MOPs, Key Performance Parameters (KPPs), and TPMs. The reviews and metrics also provide status information to support assessing system design, product realization, and technical management decisions.

NASA has multiple review cycle processes for both space flight programs and projects (see NPR 7120.5), and research and technology programs and projects. (See NPR 7120.8, NASA Research and Technology Program and Project Management Requirements.) These different review cycles all support the same basic goals but with differing formats and formalities based on the particular program or project needs.

6.7.1 Process Description

Figure 6.7-1 provides a typical flow diagram for the Technical Assessment Process and identifies typical inputs, outputs, and activities to consider in addressing technical assessment. Technical assessment is focused on providing a periodic assessment of the program/project's technical and programmatic status and health at key points in the life cycle. There are 6 criteria considered in this assessment process: alignment with and contribution to Agency strategic goals; adequacy of management approach; adequacy of technical approach; adequacy of the integrated cost and schedule estimates and funding strategy; adequacy and availability of nonbudgetary resources, and adequacy of the risk management approach.

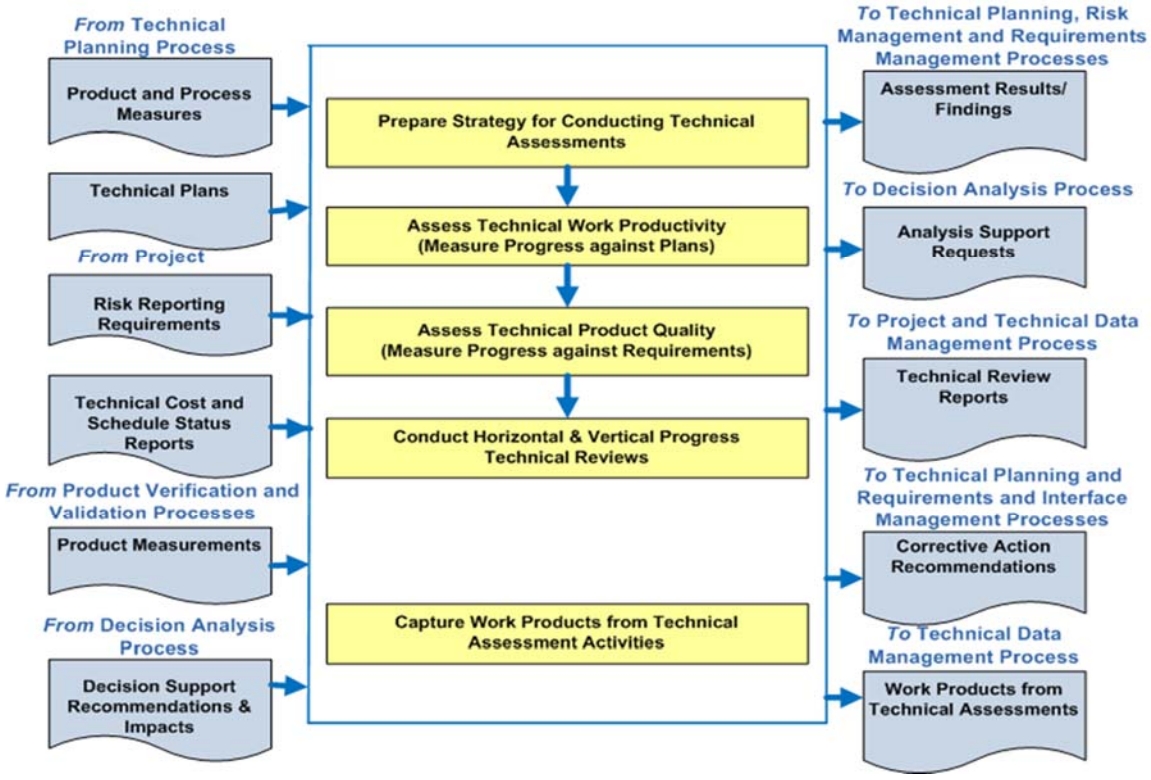


Figure 6.7-1 Technical Assessment Process

6.7.1.1 Inputs

Typical inputs needed for the Technical Assessment Process would include the following:

- **Technical Plans**: These are the planning documents that will outline the technical reviews/assessment process as well as identify the technical product/process measures that will be tracked and assessed to determine technical progress. Examples of these plans are the program (or project) plan, SEMP (if applicable), review plans (which may be part of the program or project plan), ILS plan, and EVM plan (if applicable). These plans contain the information and descriptions of the program/project's alignment with and contribution to Agency strategic goals, its management approach, its technical approach, its integrated cost and schedule, its budget, resource allocations, and its risk management approach.
- **Technical Process and Product Measures**: These are the identified technical measures that will be assessed or tracked to determine technical progress. These measures are also referred to as MOEs, MOPs, KPPs, and TPMs. (See Section 6.7.2.6.2.) They provide indications of the program/project's performance in key management, technical, cost (budget), schedule, and risk areas.
- **Reporting Requirements**: These are the requirements on the methodology in which the status of the technical measures will be reported with regard to management, technical cost (budget), schedule, and risk. The requirements apply internally to the program/project and are used externally by the Centers and Mission Directorates to assess the performance of the program or project. The methodology and tools used for reporting the status will be established on a project-by-project basis.

6.7.1.2 Process Activities

6.7.1.2.1 Prepare Strategy for Conducting Technical Assessments

As outlined in Figure 6.7-1, the technical plans provide the initial inputs into the Technical Assessment Process. These documents outline the technical reviews/assessment approach as well as identify the technical measures that will be tracked and assessed to determine technical progress. An important part of the technical planning is determining what is needed in time, resources, and performance to complete a system that meets desired goals and objectives. Project managers need visibility into the progress of those plans in order to exercise proper management control. Typical activities in determining progress against the identified technical measures include status reporting and assessing the data. Status reporting will identify where the project stands with regard to a particular technical measure. Assessing will analytically convert the output of the status reporting into a more useful form from which trends can be determined and variances from expected results can be understood. Results of the assessment activity then feed into the Decision Analysis Process (see Section 6.8) where potential corrective action may be necessary.

These activities together form the feedback loop depicted in Figure 6.7-2.

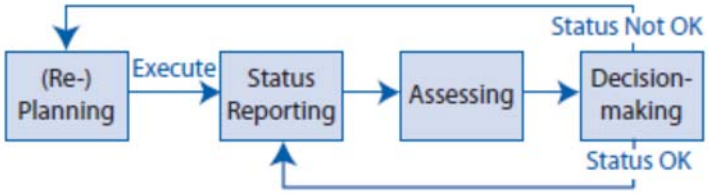


Figure 6.7-2 Planning and Status Reporting Feedback Loop

This loop takes place on a continual basis throughout the project life cycle. This loop is applicable at each level of the project hierarchy. Planning data, status reporting data, and assessments flow up the hierarchy with appropriate aggregation at each level; decisions cause actions to be taken down the hierarchy. Managers at each level determine (consistent with policies established at the next higher level of the project hierarchy) how often and in what form status data should be reported and assessments should be made. In establishing these status reporting and assessment requirements, some principles of good practice are as follows:

- Use an agreed-upon set of well-defined technical measures. (See Section 6.7.2.6.2.)
- Report these technical measures in a consistent format at all project levels.
- Maintain historical data for both trend identification and cross-project analyses.
- Encourage a logical process of rolling up technical measures (e.g., use the WBS or PBS for project progress status).
- Support assessments with quantitative risk measures.
- Summarize the condition of the project by using color-coded (red, yellow, and green) alert zones for all technical measures.

6.7.1.2.2 Assess Technical Work Productivity and Product Quality and Conduct Progress Reviews

Regular, periodic (e.g., monthly) tracking of the technical measures is recommended, although some measures should be tracked more often when there is rapid change or cause for concern. Key reviews, such as PDRs and CDRs, or status reviews are points at which technical measures and their trends should be carefully scrutinized for early warning signs of potential problems. Should there be indications that existing trends, if allowed to continue, will yield an unfavorable outcome, corrective action should begin as soon as practical. Section 6.7.2.6.1 provides additional information on status reporting and assessment techniques for costs and schedules (including EVM), technical performance, and systems engineering process metrics.

The measures are predominantly assessed during the program and project technical reviews. Typical activities performed for technical reviews include (1) identifying, planning, and conducting phase-to-phase technical reviews; (2) establishing each review's purpose, objective, and entry and success criteria; (3) establishing the makeup of the review team; and (4) identifying and resolving action items resulting from the review. Section 6.7.2.3 summarizes the types of technical reviews typically conducted on a program/project and the role of these reviews in supporting management decision processes. This section address the types of technical reviews typically conducted for both space flight and research and technology programs/projects and the role of these reviews in supporting management decision processes. It also identifies some general principles for holding reviews, but leaves explicit direction for executing a review to the program/project team to define.

The process of executing technical assessment has close relationships to other areas, such as risk management, decision analysis, and technical planning. These areas may provide input into the Technical Assessment Process or be the benefactor of outputs from the process.

Table 6.7-1 provides a summary of the types of reviews for a spaceflight project, their purpose, and timing.

Table 6.7-1 Purpose and Results for Life-Cycle Reviews for Spaceflight Projects

Name of Review	Purpose	Timing	Entrance/Success Criteria	Results of Review
Mission Concept Review (MCR)	The MCR will affirm the mission need and evaluates the proposed objectives and the concept for meeting those objectives.	The MCR should be completed prior to entering the concept development phase (Phase A)	The MCR entrance and success criteria are defined in Table G-3 of NPR 7123.1.	A successful MCR supports the determination that the proposed mission meets the customer need and has sufficient quality and merit to support a field Center management decision to propose further study to the cognizant NASA program Associate Administrator as a candidate Phase A effort.
System Requirements Review (SRR)	The SRR evaluates the functional and performance requirements defined for the system and the preliminary program or project plan and ensures that the requirements and selected concept will satisfy the mission.	The SRR is conducted during the concept development phase (Phase A) and before conducting the SDR or MDR.	The SRR entrance and success criteria for a program are defined in Table G-1 of NPR 7123.1. The SRR entrance and success criteria for projects and single-project programs are defined in Table G-4 of NPR 7123.1.	Successful completion of the SRR freezes program / project requirements and leads to a formal decision by the cognizant program Associate Administrator to proceed with proposal request preparations for project implementation

Name of Review	Purpose	Timing	Entrance/Success Criteria	Results of Review
Mission Definition Review (MDR) / System Definition Review (SDR)	Sometimes called the MDR by robotic projects and SDR for human flight projects, this review evaluates whether the proposed architecture is responsive to the functional and performance requirements and that the requirements have been allocated to all functional elements of the mission/system.	The MDR/SDR is conducted during the concept development phase (Phase A) prior to KDP B and the start of preliminary design.	The MDR/SDR entrance and success criteria for a program are defined in Table G-2 of NPR 7123.1. The MDR/SDR entrance and success criteria for projects and single-project programs are defined in Table G-5 of NPR 7123.1.	A successful MDR/SDR supports the decision to further develop the system architecture/design and any technology needed to accomplish the mission. The results reinforce the mission/system's merit and provide a basis for the system acquisition strategy. As a result of successful completion, the mission/system and its operation are well enough understood to warrant design and acquisition of the end items.
Preliminary Design Review (PDR)	The PDR demonstrates that the preliminary design meets all system requirements with acceptable risk and within the cost and schedule constraints and establishes the basis for proceeding with detailed design. It shows that the correct design options have been selected, interfaces have been identified, and verification methods have been described. The PDR should address and resolve critical, system-wide issues and show that work can begin on detailed design.	PDR occurs near the completion of the preliminary design phase (Phase B) as the last review in the Formulation Phase.	The entrance and success criteria for the PDR are defined in Table G-6 of NPR 7123.1.	As a result of successful completion of the PDR, the design-to baseline is approved. A successful review result also authorizes the project to proceed into the Implementation Phase and toward final design.
Critical Design Review (CDR)	The CDR demonstrates that the maturity of the design is appropriate to support proceeding with full scale fabrication, assembly, integration, and test. CDR determines if the technical effort is on track to complete the system development, meeting mission performance requirements within the identified cost and schedule constraints.	CDR occurs during the final design phase (Phase C).	The entrance and success criteria for the CDR are defined in Table G-7 of NPR 7123.1.	As a result of successful completion of the CDR, the build-to baseline, production, and verification plans are approved. A successful review result also authorizes coding of deliverable software (according to the build-to baseline and coding standards presented in the review) and system qualification testing and integration. All open issues should be resolved with closure actions and schedules.

Name of Review	Purpose	Timing	Entrance/Success Criteria	Results of Review
Production Readiness Review (PRR)	A PRR is held for projects developing or acquiring multiple or similar systems greater than three or as determined by the project. The PRR determines the readiness of the system developers to efficiently produce the required number of systems. It ensures that the production plans; fabrication, assembly, and integration-enabling products; and personnel are in place and ready to begin production.	PRR occurs during the final design phase (Phase C).	The entrance and success criteria for the PRR are defined in Table G-8 of NPR 7123.1.	As a result of successful completion of the PRR, the final production build-to baseline, production, and verification plans are approved. Approved drawings are released and authorized for production. A successful review result also authorizes coding of deliverable software (according to the build-to baseline and coding standards presented in the review) and system qualification testing and integration. All open issues should be resolved with closure actions and schedules.
System Integration Review (SIR)	An SIR ensures segments, components, and subsystems are on schedule to be integrated into the system. Integration facilities, support personnel, and integration plans and procedures are on schedule to support integration.	SIR occurs at the end of the final design phase (Phase C) and before the systems assembly, integration, and test phase (Phase D) begins.	The entrance and success criteria for the SIR are defined in Table G-9 of NPR 7123.1.	As a result of successful completion of the SIR, the final as-built baseline and verification plans are approved. Approved drawings are released and authorized to support integration. All open issues should be resolved with closure actions and schedules. The subsystems/systems integration procedures, ground support equipment, facilities, logistical needs, and support personnel are planned for and are ready to support integration.
System Acceptance Review (SAR)	The SAR verifies the completeness of the specific end products in relation to their expected maturity level and assesses compliance to stakeholder expectations. It also ensures that the system has sufficient technical maturity to authorize its shipment to the designated operational facility or launch site.		The entrance and success criteria for the SAR are defined in Table G-11 of NPR 7123.1.	As a result of successful completion of the SAR, the system is accepted by the buyer, and authorization is given to ship the hardware to the launch site or operational facility and to install software and hardware for operational use.

Name of Review	Purpose	Timing	Entrance/Success Criteria	Results of Review
Operational Readiness Review (ORR)	The ORR examines the actual system characteristics and procedures used in the system or end product's operation. It ensures that all system and support (flight and ground) hardware, software, personnel, procedures, and user documentation accurately reflect the deployed state of the system.		The entrance and success criteria for the ORR are defined in Table G-12 of NPR 7123.1.	As a result of successful ORR completion, the system is ready to assume normal operations.
Flight Readiness Review (FRR)	The FRR examines tests, demonstrations, analyses, and audits that determine the system's readiness for a safe and successful flight or launch and for subsequent flight operations. It also ensures that all flight and ground hardware, software, personnel, and procedures are operationally ready.		The entrance and success criteria for the FRR are defined in Table G-13 of NPR 7123.1.	As a result of successful FRR completion, technical and procedural maturity exists for system launch and flight authorization and, in some cases, initiation of system operations.
Post-Launch Assessment Review (PLAR)	A PLAR is a post-deployment evaluation of the readiness of the spacecraft systems to proceed with full, routine operations. The review evaluates the status, performance, and capabilities of the project evident from the flight operations experience since launch. This can also mean assessing readiness to transfer responsibility from the development organization to the operations organization. The review also evaluates the status of the project plans and the capability to conduct the mission with emphasis on near-term operations and mission-critical events.	This review is typically held after the early flight operations and initial checkout.	The entrance and success criteria for the PLAR are defined in Table G-14 of NPR 7123.1.	As a result of successful PLAR completion, the system is ready to assume in-space operations.
Critical Event Readiness Review (CERR)	A CERR confirms the project's readiness to execute the mission's critical activities during flight operation. These include orbital insertion, rendezvous and docking, re-entry, scientific observations / encounters, etc.		The CERR entrance and success criteria for a program are defined in Table G-15 of NPR 7123.1.	As a result of successful CER completion, the system is ready to assume (or resume) in-space operations.

Name of Review	Purpose	Timing	Entrance/Success Criteria	Results of Review
Post-Flight Assessment Review (PFAR)	The PFAR evaluates the activities from the flight after recovery. The review identifies all anomalies that occurred during the flight and mission and determines the actions necessary to mitigate or resolve the anomalies for future flights.		The entrance and success criteria for the PFAR are defined in Table G-16 of NPR 7123.1.	As a result of successful PFAR completion, the report documenting flight performance and recommendations for future missions is complete and all anomalies have been documented and dispositioned.
Decommissioning Review (DR)	The DR confirms the decision to terminate or decommission the system and assesses the readiness of the system for the safe decommissioning and disposal of system assets.	The DR is normally held near the end of routine mission operations upon accomplishment of planned mission objectives. It may be advanced if some unplanned event gives rise to a need to prematurely terminate the mission, or delayed if operational life is extended to permit additional investigations.	The entrance and success criteria for the DR are defined in Table G-17 of NPR 7123.1.	A successful DR completion ensures that the decommissioning and disposal of system items and processes are appropriate and effective.
Disposal Readiness Review (DRR)	A DRR confirms the readiness for the final disposal of the system assets.	The DRR is held as major assets are ready for final disposal.	The DRR entrance and success criteria for a program are defined in Table G-18 of NPR 7123.1.	A successful DRR completion ensures that the disposal of system items and processes are appropriate and effective.

6.7.1.2.3 Capture Work Products

The work products generated during these activities should be captured along with key decisions made, supporting decision rationale and assumptions, and lessons learned in performing the Technical Assessment Process.

6.7.1.3 Outputs

Typical outputs of the Technical Assessment Process would include the following:

- **Assessment Results, Findings, and Recommendations:** This is the collective data on the established measures from which trends can be determined and variances from expected results can be understood. Results then feed into the Decision Analysis Process where corrective action may be necessary.
- **Technical Review Reports/Minutes:** This is the collective information coming out of each review that captures the results, recommendations, and actions with regard to meeting the review's success criteria.
- **Other Work Products:** These would include strategies and procedures for technical assessment, key decisions and associated rationale, assumptions, and lessons learned.

6.7.2 Technical Assessment Guidance

6.7.2.1. Technical Review Basis

Technical reviews are conducted to ensure that three main aspects of system development properly define the system and are making reasonable progression toward the system's intended outcomes. These aspects are: requirements, design, and acceptance of the final system.

Technical review is necessary for any size program, project, or activity to help ensure that the engineering is properly defined and integrated. NASA has three separate technical review tracks:

- Space flight programs and projects are addressed in NPR 7120.5 and described in Section 6.7.2.2.
- Information technology programs and products are covered by NPR 7120.7 and are also described in Section 6.7.2.2.
- Research and technology programs and projects (which include aeronautics research) are coved by NPR 7120.8 and described in Section 6.7.2.5.

These sections address the reviews necessary for the most complex program or project.

For smaller programs, projects, or activities, all of the reviews below may not be necessary to ensure the system has achieved the intended outcomes. The process of tailoring is used (with approval from the appropriate authorities) to define and authorize any variances from the full set of reviews that are prescribed in the applicable program/project life cycle. For the smallest activity (mission types 4 and 5), the minimal technical reviews would be a requirements review, a design review, and an acceptance review. If the project or activity produces systems for space

flight, then a Flight Readiness Review (FRR) would also be necessary. Note, however, for an activity or project that is providing a system, subsystem, experiment, etc., to a larger program or project, a separate FRR may not be necessary. In these cases, the smaller project or activity may provide information to support the FRR conducted by the larger program or project.

As systems gain in complexity and length of development time, the reviews may be expanded and distributed chronologically or by system/element or both. For example, a more complicated program or project (mission type 3) may require both a Preliminary Design Review (PDR) and a Critical Design Review (CDR). Similarly, more complicated systems may need a Mission Concept Review (MCR). The most complicated program or project (mission types 1 and 2) may require a MCR and both a System Requirements Review (SRR) and a System Definition Review (SDR). Acceptance reviews can also be preceded by a Design Certification Review (DCR) to certify the design based on verification and validation results and a System Integration Review (SIR) for complex systems (mission types 1 and 2) that require significant integration prior to final assembly, test, and acceptance. An Operational Readiness Review (ORR) is often conducted prior to FRR for systems that have significant operational aspects in addition to those aspects covered by the flight review. Complex space flight operations may also require a Post-Launch Assessment Review (PLAR), Critical Event Readiness Review (CERR), and a Post-Flight Assessment Review (PFAR). Long-term programs and projects, particularly those with capital assets or orbital hardware, may have a Decommissioning Review (DR) and/or a Disposal Readiness Review (DRR) in support of program/project decommissioning/disposal.

System development or operations that are not meeting defined requirements may be subject to a termination review. These reviews are called at the discretion of the decision authority and are not part of the technical review template. It should be noted that project termination, while usually disappointing to project personnel, may be a proper reaction to changes in external conditions or to an improved understanding of the system's projected cost-effectiveness. Research and technology programs and projects often have defined off-ramps for specific activities that constitute logical points in which a specific research or technology activity can be terminated. At the termination review, the program and the project teams present status, including any material requested by the decision authority. Appropriate support organizations are represented (e.g., procurement, external affairs, legislative affairs, public affairs) as needed. The decision and basis of the decision are fully documented and reviewed with the NASA Associate Administrator or program executive, as appropriate, prior to final implementation.

As can be seen, there is a continuum of program, project, and activity complexity that is mapped to a discrete set of technical reviews. To ensure the right set of technical reviews is established and documented in the program or project plan, a thorough understanding should be obtained of the system to be developed and the mission context. With this understanding, a proper set of technical reviews can be established for the system.

As a general guide to establishing an appropriate implementation approach, risk posture, and the correct set of reviews, the type of mission/project that is being developed needs to be considered. NPR 8705.4, Risk Classification for NASA Payloads, defines the risk classifications for NASA payloads sorting between class A, B, C, or D. There is currently no NASA directive classifying non-payloads. However, the philosophy of NPR 8705.4 can be used by a Center or project to further define types of missions/systems being developed and then to define the types of reviews

or oversight that will be needed for each type. Table 3.11-1 provides an example of how the various types of systems and missions developed and executed by NASA might be further defined. This table includes some example programs/projects as a reference.

Mission types E and F are on the low end of program/project complexity and will tend to have the basic requirements, design, and acceptance reviews. Participation in an FRR or holding of a separate FRR will be necessary for flight programs, projects, and activities. Mission type C will typically need technical reviews ranging from the minimum to a separate PDR and CDR, and perhaps the addition of a physical configuration audit (PCA) and functional configuration audit (FCA) of the system in addition to the acceptance review. Again, these depend on the complexity of the system being developed. Mission type B tends toward having the full set of reviews as specified in Section 6.7.2.3. Mission type A will need to have the full set of reviews and may have these distributed among various elements with a top level program/project review integrating the lower level reviews.

6.7.2.2 Reviews, Audits, and Key Decision Points

To gain a general understanding of the various technical reviews called out in Agency policy, space flight programs and projects follow the life cycle defined in NPR 7120.5 and NPR 7123.1. IT programs and projects follow life cycles in NPR 7120.7. Research and technology programs and projects follow the life cycle defined in NPR 7120.8. The intent of the policy within each of the above-mentioned documents needs to be examined. These reviews inform the decision authority. A primary focus of NPR 7120.5, 7120.7, and NPR 7120.8 is to inform the decision authority as to the readiness of a program/project to proceed into the next phase of the life cycle. This is done for each milestone review and is tied to a KDP throughout the life cycle. For space flight KDP/milestone reviews, external independent reviewers known as Standing Review Board (SRB) members evaluate the program/project and, in the end, report their findings to the decision authority. For a space flight program or project to prepare for the SRB, the technical team should conduct its own internal peer review process. This process typically includes informal or formal peer reviews or life-cycle reviews at the subsystem and system level. (See Section 6.7.2.4.5.) For research and technology programs and projects, independent assessments are conducted periodically as discussed in Section 6.7.2.5.

The intent and policy for reviews, audits, and KDPs should be developed during Phase A and defined in the program/project plan or in a separate review plan. The specific implementation of these activities should be consistent with the types of reviews and audits described in this section and with the NASA program and project life-cycle charts (see NPR 7120.5, NPR 7120.7, or NPR 7120.8). However, the timing of reviews, formality, audits, and KDPs should accommodate the need of each specific project.

6.7.2.2.1 Purpose and Definition

The purpose of a review is to furnish the forum and process to provide NASA management and their contractors assurance that the most satisfactory approach, plan, or design has been selected; that a configuration item has been produced to meet the specified requirements; or a research and technology effort has been completed or is making adequate progress to justify continuation. Reviews help to develop a better understanding among task or project participants, open communication channels, alert participants and management to problems, and open avenues for

solutions. Reviews are intended to add value to the project and enhance project quality and the likelihood of success. This is aided by inviting outside experts to confirm the viability of the presented approach, concept, or baseline or to recommend alternatives. Reviews may be program life-cycle reviews, project life-cycle reviews, or internal reviews.

The purpose of an audit is to provide NASA management and its contractors with a thorough examination of adherence to program/project policies, plans, requirements, and specifications. Audits are the systematic examination of tangible evidence to determine adequacy, validity, and effectiveness of the activity or documentation under review. An audit may examine documentation of policies and procedures, as well as verify adherence to them.

The purpose of a KDP is to provide a scheduled event at which the decision authority determines the readiness of a program/project to progress to the next phase of the life cycle (e.g., B to C, C to D) or to the next KDP. KDPs are part of NASA's oversight and approval process for programs / projects. For a detailed description of the process and management oversight teams, see NPR 7120.5, NPR 7120.7, or NPR 7120.8. Essentially, KDPs serve as gates through which programs and projects should pass. Within each phase, a KDP is preceded by one or more reviews, including the governing Program Management Council (PMC) review. Allowances are made within a phase for the differences between human and robotic space flight programs and projects and research and technology programs, technology development projects, and research and technology portfolio projects, but phases always end with the KDP. The potential outcomes at a KDP include the following:

- Approval for continuation to the next KDP.
- Approval for continuation to the next KDP, pending resolution of actions.
- Disapproval for continuation to the next KDP. In such cases, follow-up actions may include a request for more information and/or a delta independent review; a request for a termination review (described below) for the program or the project (Phases B, C, D, and E only); direction to continue in the current phase; or redirection of the program/project.

The decision authority reviews materials submitted by the governing PMC, SRB, Program Manager (PM), project manager, and Center Management Council (CMC) in addition to agreements and program/project documentation to support the decision process. The decision authority makes decisions by considering a number of factors, including continued relevance to Agency strategic needs, goals, and objectives; continued cost affordability with respect to the Agency's resources; the viability and the readiness to proceed to the next phase; and remaining program or project risk (cost, schedule, technical, safety). Appeals against the final decision of the decision authority go to the next higher decision authority.

6.7.2.2.2 General Principles for Reviews

Several factors can affect the implementation plan for any given review, such as design complexity, schedule, cost, visibility, NASA Center practices, the review itself, etc. As such, there is no set standard for conducting a review across the Agency; however, there are key elements, or principles, that should be included in a review plan. These include definition of review scope, objectives, entrance criteria, success criteria (consistent with NPR 7123.1), and process. Definition of the review process should include identification of schedule, including

duration of the face-to-face meeting (and draft agenda), definition of roles and responsibilities of participants, identification of presentation material and data package contents, and a copy of the form to be used for Review Item Disposition (RID) and/or Request For Action (RFA) and/or comment. The review process for screening and processing discrepancies/requests/comments should also be included in the plan. The review plan should be agreed to by the technical team lead, project manager, and for SRB-type reviews, the SRB chair prior to the review.

All reviews should consist of oral presentations of the applicable project requirements and the approaches, plans, or designs that satisfy those requirements. These presentations are normally provided by the cognizant design engineers or their immediate supervisors. In addition to the SRB, the review audience should include key stakeholders, such as the science community, program executive, etc. This ensures that the project obtains buy-in from the personnel who have control over the project as well as those who benefit from a successful mission. It is also beneficial to have project personnel in attendance that are not directly associated with the design being reviewed (e.g., electric power system attending a thermal discussion). This gives the project an additional opportunity to utilize cross-discipline expertise to identify design shortfalls or recommend improvements. The audience should include non-project specialists, such as safety, quality and mission assurance, reliability, HSI, verification, and testing disciplines. Project planning and control personnel should also be included to evaluate cost and schedule aspects.

6.7.2.2.3 Program Technical Life-Cycle Reviews

Within NASA there are various types of programs:

- Single-project programs (e.g., James Webb Space Telescope Program) tend to have long development and/or operational lifetimes, represent a large investment of Agency resources in one program/project, and have contributions to that program/project from multiple organizations or agencies.
- Uncoupled programs (e.g., Discovery Program, Explorer) are implemented under a broad scientific theme and/or a common program implementation concept, such as providing frequent flight opportunities for cost-capped projects selected through AOs or NASA research announcements. Each such project is independent of the other projects within the program.
- Loosely coupled programs (e.g., Mars Exploration Program or Lunar Precursor and Robotic Program) address specific scientific or exploration objectives through multiple space flight projects of varied scope. While each individual project has an assigned set of mission objectives, architectural and technological synergies and strategies that benefit the program as a whole are explored during the Formulation process. For instance, all orbiters designed for more than one year in Mars orbit are required to carry a communication system to support present and future landers.
- Tightly coupled programs (e.g., Constellation Program) have multiple projects that execute portions of a mission or missions. No single project is capable of implementing a complete mission. Typically, multiple NASA Centers contribute to the program. Individual projects may be managed at different Centers. The program may also include other Agency or international partner contributions.

Regardless of the type, all programs are required to undergo the two technical reviews listed in Table 6.7-2. The main difference lies between uncoupled/loosely coupled programs that tend to conduct "status-type" reviews on their projects after KDP I and single-project/tightly coupled programs that tend to follow the project technical life-cycle review process post-KDP I.

Table 6.7-2 Program Technical Reviews

Review	Purpose
Program/System Requirements Review	The SRR examines the functional and performance requirements defined for the program (and its constituent projects) and ensures that the requirements and the selected concept will satisfy the program and higher level requirements. It is an internal review. Rough order of magnitude budgets and schedules are presented.
Program/System Definition Review	The SDR examines the proposed program architecture and the flowdown to the functional elements of the system.

After KDP I, single-project/tightly coupled programs are responsible for conducting the system-level reviews. These reviews bring the projects together and help ensure the flowdown of requirements and that the overall system/subsystem design solution satisfies the program requirements. The program/project reviews also help resolve interface/integration issues between projects. For the sake of this guide, single-project programs and tightly coupled programs will follow the project life-cycle review process defined after this table. Best practices and lessons learned drive programs to conduct their "concept and requirements-type" reviews prior to project concept and requirements reviews and "program design and acceptance-type" reviews after project design and acceptance reviews.

6.7.2.2.4 Project Technical Life-Cycle Reviews

The phrase "project life cycle/project milestone reviews" has, over the years, come to mean different things to various Centers. Some use it to mean the project's controlled formal review using RIDs and pre-boards/ boards, while others use it to mean the activity tied to RFAs and the SRB/KDP process. This document will use the latter process to define the term. Project life-cycle reviews are mandatory reviews as defined and convened by the decision authority, which summarize the results of internal technical processes (peer reviews) throughout the project life cycle to NASA management and/or an independent review team, such as an SRB (see NPR 7120.5). These reviews are used to assess the progress and health of a project by providing NASA management with assurance that the most satisfactory approach, plan, or design has been selected, that a configuration item has been produced to meet the specified requirements, or that a configuration item is ready for launch/operation. Some examples of life-cycle reviews include the Mission Concept Review, System Requirements Review, Preliminary Design Review, and Critical Design Review.

Specified life-cycle reviews are followed by a KDP in which the decision authority for the project determines, based on results and recommendations from the life-cycle review teams, whether or not the project can proceed to the next life-cycle phase.

6.7.2.2.5 Standing Review Boards

The SRB's role is advisory to the program/project and the convening authorities; the SRB does not have authority over any program/project content. Its review provides expert assessment of the technical and programmatic approach, risk posture, and progress against the program / project baseline. When appropriate, it may offer recommendations to improve performance and/or reduce risk.

6.7.2.2.6 Internal Reviews

During the course of a project or task, it is necessary to conduct internal reviews that present technical approaches, trade studies, analyses, and problem areas to a peer group for evaluation and comment. The timing, participants, and content of these reviews are normally defined by the project manager, lead systems engineer, or the manager of the performing organization with support from the technical team. In preparation for the life-cycle reviews, a project initiates an internal review process as defined in the project plan. These reviews are not just meetings to share ideas and resolve issues, but are internal reviews that allow the project and systems engineering to establish baseline requirements, plans, or design through the review of technical approaches, trade studies, and analyses.

Internal peer reviews provide an excellent means for controlling the technical progress of the project and are sometimes conducted by the lead systems engineer. They should also be used to ensure that all interested parties are involved in the development early on and throughout the process. Thus, representatives from areas such as manufacturing and quality assurance should attend the internal reviews as active participants. It is also a good practice to include representatives from other Centers and outside organizations providing support or developing systems or subsystems that may interface to your system/subsystem. They can then, for example, ensure that the design can be produced and integrated, and that quality is managed through the project life cycle.

Since internal peer reviews will be at a much greater level of detail than the life-cycle reviews, the team may utilize internal and external experts to help develop and assess approaches and concepts at the internal reviews. Some organizations form a red team to provide an internal, independent, peer review to identify deficiencies and offer recommendations. Projects often refer to their internal reviews as "tabletop" reviews or "interim" design reviews. Whatever the name, the purpose is the same: to ensure the readiness of the baseline for successful project life-cycle review. It should be noted that due to the importance of these reviews, each review should have well-defined entrance and success criteria established prior to the review.

Peer reviews provide the technical insight essential to ensure product and process quality. Peer reviews are focused, in-depth technical reviews that support the evolving design and development of a product, including critical documentation or data packages. They can be formal or informal in nature. They are often, but not always, held as supporting reviews for technical reviews such as PDR and CDR. One purpose of the peer review is to add value and reduce risk through expert knowledge infusion, confirmation of approach, identification of defects, and specific suggestions for product improvements.

Each product peer review should have well-defined objectives, a set of requirements that the product is expected to meet, and a peer review checklist established prior to the review. The results of the engineering peer reviews comprise a key element of the review process. The results and issues that surface during these reviews are documented and reported to the appropriate next higher element level. For formal software peer reviews, refer *to NASA-HDBK-2203, NASA Software Engineering Handbook*.

The peer reviewers should be selected from outside the project, but they should have a similar technical background, and they should be selected for their skill and experience. Peer reviewers should be concerned with only the technical integrity and quality of the product. Peer reviews should be kept simple and informal. They should concentrate on a review of the documentation and minimize the viewgraph presentations. A roundtable format rather than a stand-up presentation is preferred. The peer reviews should give the full technical picture of items being reviewed.

Technical depth should be established at a level that allows the review team to gain insight into the technical risks. Rules need to be established to ensure consistency in the peer review process. At the conclusion of the review, a report on the findings, recommendations, and actions should be distributed to the technical team. The entrance and success criteria for the peer review are defined in Table G-19 of NPR 7123.1.

For those projects where systems engineering is done out-of-house, peer reviews should be part of the contract.

A successful peer review completion ensures that the items under review are proceeding in a manner with acceptable technical, cost, and schedule risk to support the successful completion of the system.

6.7.2.3 Required Technical Reviews for Space Flight Projects

This subsection describes the purpose, timing, objectives, success criteria, and results of the NPR 7123.1 required technical reviews in the NASA space flight program and project life cycles. This information is intended to provide guidance to program/project managers and systems engineers and to illustrate the progressive maturation of review activities and systems engineering products. For flight systems and ground support projects, the NASA life-cycle phases of Formulation and Implementation divide into seven project phases. The checklists provided below aid in the preparation of specific review entry and success criteria but do not take their place. To minimize extra work, review material should be keyed to program/project documentation. Table 6.7-3 summarizes the purpose, timing, and results of each type of review required for space flight projects under the purview of NPR 7120.5.

For each of the reviews, the objectives are as follows:

- Ensure a thorough review of the products supporting the review.
- Ensure the products meet the entrance criteria and success criteria.
- Ensure issues raised during the review are appropriately documented and a plan for resolution is prepared.

Table 6.7-3 Purpose and Results for Life-Cycle Reviews for Spaceflight Projects

Name of Review	Purpose	Timing	Entrance/Success Criteria	Results of Review
Mission Concept Review (MCR)	The MCR will affirm the mission need and evaluates the proposed objectives and the concept for meeting those objectives.	The MCR should be completed prior to entering the concept development phase (Phase A)	The MCR entrance and success criteria are defined in Table G-3 of NPR 7123.1.	A successful MCR supports the determination that the proposed mission meets the customer need and has sufficient quality and merit to support a field Center management decision to propose further study to the cognizant NASA program Associate Administrator as a candidate Phase A effort.
System Requirements Review (SRR)	The SRR evaluates the functional and performance requirements defined for the system and the preliminary program or project plan and ensures that the requirements and selected concept will satisfy the mission.	The SRR is conducted during the concept development phase (Phase A) and before conducting the SDR or MDR.	The SRR entrance and success criteria for a program are defined in Table G-1 of NPR 7123.1. The SRR entrance and success criteria for projects and single-project programs are defined in Table G-4 of NPR 7123.1.	Successful completion of the SRR freezes program / project requirements and leads to a formal decision by the cognizant program Associate Administrator to proceed with proposal request preparations for project implementation

Name of Review	Purpose	Timing	Entrance/Success Criteria	Results of Review
Mission Definition Review (MDR) / System Definition Review (SDR)	Sometimes called the MDR by robotic projects and SDR for human flight projects, this review evaluates whether the proposed architecture is responsive to the functional and performance requirements and that the requirements have been allocated to all functional elements of the mission/system.	The MDR/SDR is conducted during the concept development phase (Phase A) prior to KDP B and the start of preliminary design.	The MDR/SDR entrance and success criteria for a program are defined in Table G-2 of NPR 7123.1. The MDR/SDR entrance and success criteria for projects and single-project programs are defined in Table G-5 of NPR 7123.1.	A successful MDR/SDR supports the decision to further develop the system architecture/design and any technology needed to accomplish the mission. The results reinforce the mission/system's merit and provide a basis for the system acquisition strategy. As a result of successful completion, the mission/system and its operation are well enough understood to warrant design and acquisition of the end items.
Preliminary Design Review (PDR)	The PDR demonstrates that the preliminary design meets all system requirements with acceptable risk and within the cost and schedule constraints and establishes the basis for proceeding with detailed design. It shows that the correct design options have been selected, interfaces have been identified, and verification methods have been described. The PDR should address and resolve critical, system-wide issues and show that work can begin on detailed design.	PDR occurs near the completion of the preliminary design phase (Phase B) as the last review in the Formulation Phase.	The entrance and success criteria for the PDR are defined in Table G-6 of NPR 7123.1.	As a result of successful completion of the PDR, the design-to baseline is approved. A successful review result also authorizes the project to proceed into the Implementation Phase and toward final design.
Critical Design Review (CDR)	The CDR demonstrates that the maturity of the design is appropriate to support proceeding with full scale fabrication, assembly, integration, and test. CDR determines if the technical effort is on track to complete the system development, meeting mission performance requirements within the identified cost and schedule constraints.	CDR occurs during the final design phase (Phase C).	The entrance and success criteria for the CDR are defined in Table G-7 of NPR 7123.1.	As a result of successful completion of the CDR, the build-to baseline, production, and verification plans are approved. A successful review result also authorizes coding of deliverable software (according to the build-to baseline and coding standards presented in the review) and system qualification testing and integration. All open issues should be resolved with closure actions and schedules.

Name of Review	Purpose	Timing	Entrance/Success Criteria	Results of Review
Production Readiness Review (PRR)	A PRR is held for projects developing or acquiring multiple or similar systems greater than three or as determined by the project. The PRR determines the readiness of the system developers to efficiently produce the required number of systems. It ensures that the production plans; fabrication, assembly, and integration-enabling products; and personnel are in place and ready to begin production.	PRR occurs during the final design phase (Phase C).	The entrance and success criteria for the PRR are defined in Table G-8 of NPR 7123.1.	As a result of successful completion of the PRR, the final production build-to baseline, production, and verification plans are approved. Approved drawings are released and authorized for production. A successful review result also authorizes coding of deliverable software (according to the build-to baseline and coding standards presented in the review) and system qualification testing and integration. All open issues should be resolved with closure actions and schedules.
System Integration Review (SIR)	An SIR ensures segments, components, and subsystems are on schedule to be integrated into the system. Integration facilities, support personnel, and integration plans and procedures are on schedule to support integration.	SIR occurs at the end of the final design phase (Phase C) and before the systems assembly, integration, and test phase (Phase D) begins.	The entrance and success criteria for the SIR are defined in Table G-9 of NPR 7123.1.	As a result of successful completion of the SIR, the final as-built baseline and verification plans are approved. Approved drawings are released and authorized to support integration. All open issues should be resolved with closure actions and schedules. The subsystems/systems integration procedures, ground support equipment, facilities, logistical needs, and support personnel are planned for and are ready to support integration.
System Acceptance Review (SAR)	The SAR verifies the completeness of the specific end products in relation to their expected maturity level and assesses compliance to stakeholder expectations. It also ensures that the system has sufficient technical maturity to authorize its shipment to the designated operational facility or launch site.		The entrance and success criteria for the SAR are defined in Table G-11 of NPR 7123.1.	As a result of successful completion of the SAR, the system is accepted by the buyer, and authorization is given to ship the hardware to the launch site or operational facility and to install software and hardware for operational use.

Name of Review	Purpose	Timing	Entrance/Success Criteria	Results of Review
Operational Readiness Review (ORR)	The ORR examines the actual system characteristics and procedures used in the system or end product's operation. It ensures that all system and support (flight and ground) hardware, software, personnel, procedures, and user documentation accurately reflect the deployed state of the system.		The entrance and success criteria for the ORR are defined in Table G-12 of NPR 7123.1.	As a result of successful ORR completion, the system is ready to assume normal operations.
Flight Readiness Review (FRR)	The FRR examines tests, demonstrations, analyses, and audits that determine the system's readiness for a safe and successful flight or launch and for subsequent flight operations. It also ensures that all flight and ground hardware, software, personnel, and procedures are operationally ready.		The entrance and success criteria for the FRR are defined in Table G-13 of NPR 7123.1.	As a result of successful FRR completion, technical and procedural maturity exists for system launch and flight authorization and, in some cases, initiation of system operations.
Post-Launch Assessment Review (PLAR)	A PLAR is a post-deployment evaluation of the readiness of the spacecraft systems to proceed with full, routine operations. The review evaluates the status, performance, and capabilities of the project evident from the flight operations experience since launch. This can also mean assessing readiness to transfer responsibility from the development organization to the operations organization. The review also evaluates the status of the project plans and the capability to conduct the mission with emphasis on near-term operations and mission-critical events.	This review is typically held after the early flight operations and initial checkout.	The entrance and success criteria for the PLAR are defined in Table G-14 of NPR 7123.1.	As a result of successful PLAR completion, the system is ready to assume in-space operations.
Critical Event Readiness Review (CERR)	A CERR confirms the project's readiness to execute the mission's critical activities during flight operation. These include orbital insertion, rendezvous and docking, re-entry, scientific observations / encounters, etc.		The CERR entrance and success criteria for a program are defined in Table G-15 of NPR 7123.1.	As a result of successful CER completion, the system is ready to assume (or resume) in-space operations.

Name of Review	Purpose	Timing	Entrance/Success Criteria	Results of Review
Post-Flight Assessment Review (PFAR)	The PFAR evaluates the activities from the flight after recovery. The review identifies all anomalies that occurred during the flight and mission and determines the actions necessary to mitigate or resolve the anomalies for future flights.		The entrance and success criteria for the PFAR are defined in Table G-16 of NPR 7123.1.	As a result of successful PFAR completion, the report documenting flight performance and recommendations for future missions is complete and all anomalies have been documented and dispositioned.
Decommissioning Review (DR)	The DR confirms the decision to terminate or decommission the system and assesses the readiness of the system for the safe decommissioning and disposal of system assets.	The DR is normally held near the end of routine mission operations upon accomplishment of planned mission objectives. It may be advanced if some unplanned event gives rise to a need to prematurely terminate the mission, or delayed if operational life is extended to permit additional investigations.	The entrance and success criteria for the DR are defined in Table G-17 of NPR 7123.1.	A successful DR completion ensures that the decommissioning and disposal of system items and processes are appropriate and effective.
Disposal Readiness Review (DRR)	A DRR confirms the readiness for the final disposal of the system assets.	The DRR is held as major assets are ready for final disposal.	The DRR entrance and success criteria for a program are defined in Table G-18 of NPR 7123.1.	A successful DRR completion ensures that the disposal of system items and processes are appropriate and effective.

6.7.2.4 Other Technical Reviews

These typical technical reviews are some that have been conducted on previous programs and projects but are not required as part of the NPR7123.1 systems engineering process.

6.7.2.4.1 Design Certification Review

Purpose

The Design Certification Review (DCR) ensures that the qualification verifications demonstrate design compliance with functional and performance requirements. A DCR is a form of Safety and Mission Success Review (SMSR) (NPR 8705.6, Safety and Mission Assurance (SMA) Audits, Reviews, and Assessments) and may be required before critical flight activities as defined by the chief safety officer or the chief engineer. The need for these reviews should be documented in the program or project plan. The DCR follows the system CDR, and after qualification tests and all modifications needed to implement qualification-caused corrective actions have been completed.

Objectives

The objectives of the review are as follows:

- Confirm that the verification results met functional and performance requirements, and that test plans and procedures were executed correctly in the specified environments.
- Certify that traceability between test article and production article is correct, including name, identification number, and current listing of all waivers.
- Identify any incremental tests required or conducted due to design or requirements changes made since test initiation, and resolve issues regarding their results.

Criteria for Successful Completion

The following items comprise a checklist to aid in determining the readiness of DCR product preparation:

- Are the pedigrees of the test articles directly traceable to the production units?
- Is the verification plan used for this article current and approved?
- Do the test procedures and environments used comply with those specified in the plan?
- Are there any changes in the test article configuration or design resulting from the as-run tests? Do they require design or specification changes and/or retests?
- Have design and specification documents been audited?
- Do the verification results satisfy functional and performance requirements?
- Do the verification, design, and specification documentation correlate?
- Is the human element adequately integrated, evaluated, and documented; e.g., human/system functional allocation, ConOps, maintenance and logistics plans, etc.?

Results of Review

As a result of a successful DCR, the end item design is approved for production. All open issues should be resolved with closure actions and schedules.

6.7.2.4.2 Functional and Physical Configuration Audits

Configuration audits confirm that the configured product is accurate and complete. The two types of configuration audits are the Functional Configuration Audit (FCA) and the Physical Configuration Audit (PCA). The FCA examines the functional characteristics of the configured product and verifies that the product has met, through test results, the requirements specified in its functional baseline documentation approved at the SRR and PDR. FCAs are conducted on both hardware or software configured products and precede the PCA of the configured product. The PCA (also known as a configuration inspection) examines the physical configuration of the configured product and verifies that the product corresponds to the build-to (or code-to) product baseline documentation previously approved at the CDR. PCAs are conducted on both hardware and software configured products. The representative audit data list for the FCA and PCA is defined in Table 6.7-4. A successful FCA/PCA completion ensures that all of the system functional and physical components in the configured product are accurate and complete.

Table 6.7‑4 Functional and Physical Configuration Audits

Representative Audit Data List	
FCA	PCA
• Design specifications • Design drawings and parts list • Engineering change proposals/engineering change requests • Deviation/waiver approval requests incorporated and pending • Specification and drawing tree • Fracture control plan • Structural dynamics, analyses, loads, and models documentation • Materials usage agreements/materials identification usage list • Verification and validation requirements, plans, procedures, and reports • Software requirements and development documents • Listing of accomplished tests and test results • CDR completion documentation including RIDs/RFAs and disposition reports • Analysis reports • ALERT (Acute Launch Emergency Restraint Tip) tracking log • Hazard analysis/risk assessment • Human element integration (HSI Plan and updates)	• Final version of all specifications • Product drawings and parts list • Configuration accounting and status reports • Final version of all software and software documents • Copy of all FCA findings for each product • List of approved and outstanding engineering change proposals, engineering change requests, and deviation/waiver approval requests • Indentured parts list • As-run test procedures • Drawing and specification tree • Manufacturing and inspection “build” records • Inspection records • As-built discrepancy reports • Product log books • As-built configuration list

6.7.2.4.3 Test Readiness Review

A TRR ensures that the test article (hardware/software), test facility, support personnel, and test procedures are ready for testing and data acquisition, reduction, and control. A TRR can be conducted multiple times during a project before major V&V activities such as a test or a series of tests. The TRR is a form of Safety and Mission Success Review (SMSR) (see NPR 8715.3 and NPR 8705.6) and may be required before hazardous tests involving new or modified hardware as defined by the chief safety officer or the chief engineer. The need for these reviews should be documented in the program or project plan. The entrance and success criteria for the TRR are defined in Table G-10 of NPR 7123.1. A successful TRR signifies that test and safety engineers have certified that preparations are complete, and that the project manager has authorized formal test initiation.

6.7.2.4.4 Program Implementation Reviews/Program Status Reviews

PIRs or PSRs are periodically conducted, as required by the decision authority and documented in the program plan, during the Implementation Phase to evaluate the program's continuing relevance to the Agency's strategic plan. These reviews assess the program performance with respect to expectations and determine the program's ability to execute the implementation plan with acceptable risk within cost and schedule constraints. PIRs/PSRs generally occur periodically during the program/project's operational phase. The PIR/PSR entrance and success criteria for a program are defined in Table G-20 of NPR 7123.1. A successful PIR/PSR completion ensures that the program is proceeding consistent with Agency strategic goals and commitments.

6.7.2.5 Research and Technology Reviews

Within NASA there are various types of research and technology programs:

- Research and Technology (R&T) programs consist of R&T projects that are comprised of strictly R&T investments investigating specific physical phenomena or logic or developing specific technologies. These R&T projects tend to define a cost/schedule structure rather than a Life-Cycle Cost (LCC) and end date.
- A Technology Development (TD) project is a focused development of a technology leading to future implementation in an aeronautics or space flight application. The status reviews for these types of projects may follow the template for specific space flight project reviews depending on the technology development approach. A TD project may be referenced elsewhere in Agency documentation as Advanced Technology Development (ATD).
- An R&T portfolio project may be made up of one or more groups of R&T investigations that address the goals and objectives defined for the R&T portfolio. These projects consist of related but separately managed research and technology investigations. An R&T portfolio project may be referenced elsewhere in Agency documentation as Basic and Applied Research (BAR).

Regardless of the type, all R&T programs/projects follow a status review format that is defined specifically for the program/project.

6.7.2.5.1 Internal Reviews

During the course of a project or task, it is necessary to conduct internal reviews that present technical approaches, trade studies, analyses, and problem areas to a peer group for evaluation and comment. The timing, participants, and content of these reviews are normally defined by the project manager or the manager of the performing organization with support from the technical team. In preparation for the life-cycle reviews, a project initiates an internal review process as defined in the project plan. These reviews are not just meetings to share ideas and resolve issues, but are reviews that allow the project to understand the research or technology development progress and make adjustments to research plans, schedules, and/or funding.

It should be noted that due to the importance of these reviews, each review should have well-defined objectives established prior to the review.

6.7.2.5.2 Required Technical Reviews

This subsection describes the purpose, timing, objectives, success criteria, and results of the required reviews in the NASA R&T program and project life cycles. This information is intended to provide guidance to program/project managers, systems engineers, and principle investigators, and to illustrate the progressive maturation of R&T activities.

In R&T programs and projects the role of the systems engineer varies. Often the principle investigator is responsible for the systems engineering functions, as needed, for research and technology efforts. A research systems engineer may be responsible for systems engineering in technology development programs/projects. Either a research systems engineer or a program/project level Principle Investigator may be responsible for R&T portfolio projects systems engineering, integrating across the various research activities within the portfolio. Principle Investigators are responsible for individual research projects (within a program) or individual research activities (within a project).

The four basic review types associated with R&T programs are formulation review, status review, independent assessment, and peer review. Refer to NPR 7120.8 for a fuller description of the overall requirements for conducting these reviews.

6.7.2.5.3 Formulation Review

Prior to KDP I (programs) or KDP C (projects), a Formulation Review is conducted in order to determine if the R&T program/project is ready to proceed to the Implementation Phase. The review consists of an internal and external component. The internal component is an assessment of the feasibility of the technology concepts; risk; acquisition strategies; high-level requirements and success criteria; plans, budgets, and schedules; and identification of how the program or project supports the Agency's strategic needs, goals, and objectives. The external component is an independent assessment and is performed by a separate organization under the direction of the selecting official and the Terms of Reference (ToR) for the Formulation Review, as described in NPR 7120.8.

6.7.2.5.4 Status Review

Status reviews are mandatory reviews convened by the decision authority to summarize the results of technical progress throughout the project life cycle to NASA management and/or an independent assessment review team. (See NPR 7120.8.) These reviews are used to assess the progress and health of a project by providing NASA management with assurance that the most satisfactory R&T approach and plan have been selected, that the research and technology development are progressing and are still viable, or that the R&T has come to a logical ending point.

6.7.2.5.5 Independent Assessment

An independent assessment is a specific assessment or review conducted by an entity that is outside the advocacy chain of the program or project. There are three types: relevance, quality, and performance. An independent assessment for relevance determines that the program/project is relevant to national priorities, agency missions, relevant fields, and "customer" needs, and can justify its claim on taxpayer resources. An independent assessment for quality determines that a program/project will maximize the quality of the R&D they fund through the use of a clearly stated, defensible method for awarding a significant majority of their funding. Programs/projects should assess and report on the quality of current and past R&D. Lastly, an independent assessment for performance determines that a program or project has met its high priority, multiyear R&D objectives with annual performance outputs and milestones that show how one or more outcomes will be reached. Specific guidance for these reviews is found in NPR 7120.8, NASA Research and Technology Program and Project Requirements, and NPR 1080.1, Requirements for the Conduct of NASA Research and Technology.

6.7.2.5.6 Peer Review

Peer review is a process in which a group of technically knowledgeable people with reputations for integrity and relevant expertise is convened to provide, to the maximum extent possible, unbiased evaluations of the merit and technical validity of proposed work. Specific goals of peer review are as follows:

- Determine the quality, relevance, and value of the work being proposed.
- Identify the work most likely to succeed, or work that might be high risk but would result in high reward.
- Assess the relative merits of the proposed work to the state-of-the-art in both current knowledge and similar work being conducted by other groups.
- Determine the scientific and technical merits of each proposal, consistent with the evaluation factors stated in the solicitation.
- Demonstrate to internal and external communities that excellence and fairness are achieved in arriving at scientific and technical decisions by making the R&T communities themselves participants in the selection process.

Internal peer reviews provide an excellent means for tracking the research progress of the project. They should also be used to ensure that all interested parties are involved in the research early on and throughout the process. Thus, representatives from contributing research disciplines

should attend the internal reviews as active participants. It is also a good practice to include representatives from other Centers and outside organizations providing support or developing systems or subsystems for the R&T effort. They can then, for example, ensure that the research team is coordinated and the R&T is proceeding along the research or technology development plan.

Since internal peer reviews will be at a much greater level of detail than the status reviews, the team may utilize internal and external experts to help develop and assess approaches and concepts at the internal reviews. Some organizations form a red team to provide an internal, independent, peer review to identify deficiencies and offer recommendations. Projects often refer to their internal reviews as "tabletop" reviews. Whatever the name, the purpose is to ensure the coordination and progress of the R&T effort for successful completion.

Each product peer review should have well-defined objectives, a set of requirements that the product is expected to meet, and a peer review checklist established prior to the review.

For additional details on R&T peer reviews, refer to NPR 1080.1, Requirements for the Conduct of NASA Research and Technology (R&T).

6.7.2.6 Status Reporting and Assessment

This subsection provides additional information on status reporting and assessment techniques for costs and schedules (including EVM), technical performance, and systems engineering process metrics.

6.7.2.6.1 Cost and Schedule Control Measures

Status reporting and assessment on costs and schedules provides the project manager and systems engineer visibility into how well the project is tracking against its planned cost and schedule targets. From a management point of view, achieving these targets is on a par with meeting the technical performance requirements of the system. It is useful to think of cost and schedule status reporting and assessment as measuring the performance of the "system that produces the system."

NPR 7120.5 provides specific requirements for the application of EVM to support cost and schedule management. EVM is applicable to both in-house and contracted efforts. The level of EVM system implementation will depend on the dollar value and risk of a project or contract. The standard for EVM systems is ANSI-EIA-748. The project manager/systems engineer uses these guidelines to establish the program and project EVM implementation plan.

NASA's EVM requirements apply to applicable programs/projects/contracts as defined in NPRs 7120.5, 7120.7, and 7120.8 and the NASA Federal Acquisition Regulation Supplement (NFS) 1834.201, Earned Value Management System Policy. Planning for EVM begins early in project Formulation (Phases A and B) and is applied in project Implementation (Phases C and D). The project's preliminary Performance Measurement Baseline (PMB) is established in Phase B in preparation for Key Decision Point (KDP) C. Projects should use the NASA EVM capability to ensure compliance with the EVM requirements.

NASA's EVM capability is located on the NASA Engineering Network (NEN) EVM Community of Practice located at https://nen.nasa.gov/web/pm/evm. Additionally, *NASA's Earned Value Management (EVM) Implementation Handbook, NASA/SP-2012-599,* provides detailed guidance on EVM implementation and can be found at the NASA EVM website http://evm.nasa.gov/index.html along with an EVM tutorial, requirements, guidance, resources, tools and other relevant NASA handbooks.

Assessment Methods – Earned Value Management (EVM)

Performance measurement data are used to assess project cost, schedule, and technical performance and their impacts on the completion cost and schedule of the project. In program control terminology, a difference between actual performance and planned costs or schedule status is called a "variance." Variances should be controlled at the control account level, which is typically at the subsystem WBS level. The person responsible for this activity is frequently called the Control Account Manager (CAM). The CAM develops work and product plans, schedules, and time-phased resource plans. The technical subsystem manager/leads often take on this role as part of their subsystem management responsibilities.

Figure 6.7-3 illustrates two types of variances, cost and schedule, and some related concepts. A product-oriented WBS divides the project work into discrete tasks and products. Associated with each task and product (at any level in the WBS) is a schedule and a budgeted (i.e., planned) cost. The Budgeted Cost for Work Scheduled ($BCWS_t$) for any set of WBS elements is the sum of the budgeted cost of all work on tasks and products in those elements scheduled to be completed by time *t.* The Budgeted Cost for Work Performed ($BCWP_t$), also called Earned Value (EV_t), is the sum of the budgeted cost for tasks and products that have actually been produced at time *t* in the schedule for those WBS elements. The difference, $BCWP_t$ and $BCWS_t$, is called the schedule variance at time *t.* A negative value indicates that the work is behind schedule.

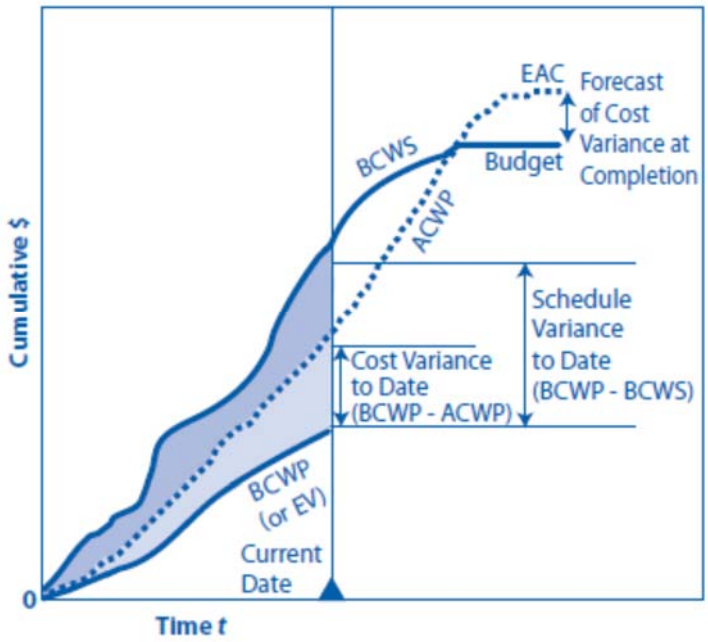


Figure 6.7-3 Cost and Schedule Variances

The Actual Cost of Work Performed ($ACWP_t$) represents the funds that have been expended up to time *t* on those WBS elements. The difference between the budgeted and actual costs, $BCWP_t$ - $ACWP_t$, is called the cost variance at time *t.* A negative value here indicates a cost overrun.

When either schedule variance or cost variance exceeds pre-established control-account-level thresholds that represent significant departures from the baseline plan, the conditions should be analyzed to identify why the variance exists. Once the cause is understood, the CAM can make an informed forecast of the time and resources needed to complete the control account. When corrective actions are feasible (can stay within the BCWS), the plan for implementing them should be included in the analysis. Sometimes no corrective action is feasible; overruns or schedule slips may be unavoidable. However, the earlier a technical problem is identified as a result of schedule or cost variances, the more likely the project team can minimize the impact on completion.

Variances may indicate that the cost Estimate at Completion (EAC) of the project is likely to be different from the Budget at Completion (BAC). The difference between the BAC and the EAC is the Variance at Completion (VAC). A negative VAC is generally unfavorable, while a positive is usually favorable. These variances may also point toward a change in the scheduled completion date of the project. These types of variances enable a program analyst to estimate the EAC at any point in the project life cycle. (See box on analyzing EAC.) These analytically derived estimates should be used only as a "sanity check" against the estimates prepared in the variance analysis process.

If the cost and schedule baselines and the technical scope of the work are not adequately defined and fully integrated, then it is very difficult (or impossible) to estimate the current cost EAC of the project.

Other efficiency factors can be calculated using the performance measurement data. The Schedule Performance Index (SPI) is a measure of work accomplishment in dollars. The SPI is calculated by dividing work accomplished in dollars or BCWP by the dollar value of the work scheduled or BCWS. Just like any other ratio, a value less than one is a sign of a behind-schedule condition, equal to one indicates an on-schedule status, and greater than one denotes that work is ahead of schedule. The Cost Performance Index (CPI) is a measure of cost efficiency and is calculated as the ratio of the earned value or BCWP for a segment of work compared to the cost to complete that same segment of work or ACWP. A CPI will show how much work is being accomplished for every dollar spent on the project. A CPI of less than one reveals negative cost efficiency, a CPI equal to one is right on cost, and a CPI greater than one is positive. Note that traditional measures compare planned cost to actual cost; however, this comparison is never made using earned value data. Comparing planned to actual costs is an indicator only of spending and not of overall project performance.

Analyzing the Estimate at Completion

An EAC can be estimated at any point in the project and should be reviewed at least on a monthly basis. The EAC requires a detailed review by the CAM. A statistical estimate can be used as a cross-check of the CAM's estimate and to develop a range to bound the estimate. The appropriate formula used to calculate the statistical EAC depends upon the reasons associated with any variances that may exist. If a variance exists due to a one-time event, such as an accident, then EAC = ACWP + (BAC – BCWP). The CPI and SPI should also be considered in developing the EAC.

If there is a growing number of liens, action items, or significant problems that will increase the difficulty of future work, the EAC might grow at a greater rate than estimated by the above equation. Such factors could be addressed using risk management methods described in the section 6.4.

6.7.2.6.2 Technical Measures — MOEs, MOPs, and TPMs

Measures of Effectiveness

MOEs are the "operational" measures of success that are closely related to the achievement of mission or operational goals and objectives in the intended operational environment. They are extracted from stakeholder expectations and are stated from the customer/user viewpoint. An MOE represents a stakeholder expectation that is critical to the success of the system/mission, and failure to attain its critical value, attribute, or feature may cause the stakeholder to judge the system/mission a failure. MOEs are developed at the time the NGOs or other representations of the stakeholder expectations are defined and are also validated with the customer/user. MOEs may be either quantitative or qualitative (subjective) in nature. MOEs are typically not directly used as a technical requirement for the system but will be the basis for the concept of operations and requirements definition. MOEs are intended to focus on *how well* mission or operational objectives are achieved, not on *how* they are achieved, i.e., MOEs should be independent of any particular solution. As such, MOEs are the standards against which the "goodness" of each proposed solution may be assessed in trade studies and decision analyses. Measuring or evaluating MOEs not only makes it possible to compare alternative solutions quantitatively, but sensitivities to key assumptions regarding operational environments and to any underlying MOPs can also be investigated. (See Measures of Performance (MOP) discussion below.)

In the systems engineering process, MOEs are used to:

- Define high-level operational requirements from the customer/stakeholder viewpoint.
- Compare and rank alternative solutions in trade studies.
- Investigate the relative sensitivity of the projected mission or operational success to key operational assumptions and performance parameters.
- Determine that the mission or operational success quantitative objectives remain achievable as system development proceeds. (See TPM discussion below.)

Measures of Performance

MOPs are the measures that characterize physical or functional attributes relating to the system end product. They define the key performance characteristics that the system should have when fielded and operated in its intended environment in order to satisfy the associated MOEs. They are expressed in terms of distinctly quantifiable performance features, such as speed, range, or frequency. These attributes are generally measured (verified) under specified conditions or operational environments prior to deployment of the system. They are evaluated during project executions and used as input to management, including as indicators to aid managing technical risks. MOPs are attributes deemed important in achieving mission or operational success. MOPs are generated during requirements definition and provide insight into the performance of the system. MOPs are derived from MOEs and are stated from the supplier's viewpoint. There can be one or multiples of MOPs associated with an MOE. The MOPs, being quantitative, are used to validate a MOE. A MOE is validated when all the associated MOPs are satisfactorily achieved. As described in the next section, some MOPs are key performance parameters.

A distinction between MOEs and MOPs is that they are formulated from different viewpoints. While MOEs are stated from the customer/user viewpoint, MOPs are stated from the supplier's viewpoint. A MOE represents a stakeholder expectation that is critical to the success of the system, and failure to attain its critical value, attribute or feature will cause the stakeholder to judge the system a failure. A MOP is a measure of the desired performance of a supplier's design solution.

Examples of MOEs

- A spacecraft capable of human occupation is established and maintained in lunar orbit
- Ability to support a crewed mission of 180 days
- The operational life of the satellite is at least 10 years
- Valid data is obtained as to the velocity of the particle
- Detect 40 hot (500K – 1000K) Jupiter-sized planets heated by a parent star at distances up to 20 parsecs
- Establish permanent presence for a crew of five people in lunar orbit by 2020

Examples of MOPs

- Provide capability to conduct observations within a wavelength range of 3.0 to 10.0 microns
- Provide capability to maintain crew compartment atmosphere pressure at 1.013×10^5 N/m^2
- The standard deviation of velocities falls within 1 sigma of the shuttle system
- System to provide 100 hours of uninterrupted computing

Key Performance Parameters

For spaceflight programs and projects, Key Performance Parameters (KPPs) are those performance capabilities and characteristics that are considered most essential for the operation of the system to satisfy the mission. Failure to meet a KPP threshold will put the project in cost, schedule, or performance risk status and can be cause for the project to be reevaluated or terminated. KPPs are the minimum number of performance parameters established to characterize the major drivers of operational performance including supportability and interoperability. The program/project defines the KPPs when developing the MOEs, concept of operations, and MOPs and when requirements are being defined. TPMs should be considered for each KPP. Figure 6.7-4 depicts the International Council on Systems Engineering (INCOSE) view of the relationship of the MOEs, KPPs, MOPs, and TPMs. Other organizations may have a different relationship, such as the KPPs being a subset of the MOPs. The important thing is that the program/project has a clear definition of these terms and how they relate to each other. The use of "MOP" elsewhere in this guide should imply that the KPPs have been incorporated, regardless of how they got there.

For research and technology programs, projects, and activities that include aeronautics, KPPs define the technical performance goals of the technology development or research investigation. KPPs are typically defined in measurable engineering terms so that they are relevant and meaningful to future system development engineers. A threshold value is established for the minimal acceptable performance from a system developer standpoint, and a goal value is also specified as the intended value to be achieved for the technology development or research investigation. Many factors are involved in setting R&T KPPs including state of the art performance values and application environment for the research and technology component, subsystem, or system. The KPPs are related to assessment of the development or research

efforts' Technology Readiness Level (TRL), and KPP values are often associated with a specific TRL. In addition, R&T roadmaps and milestones are based on KPP levels. As such, KPPs are the main criteria from which the progress of the R&T effort is measured.

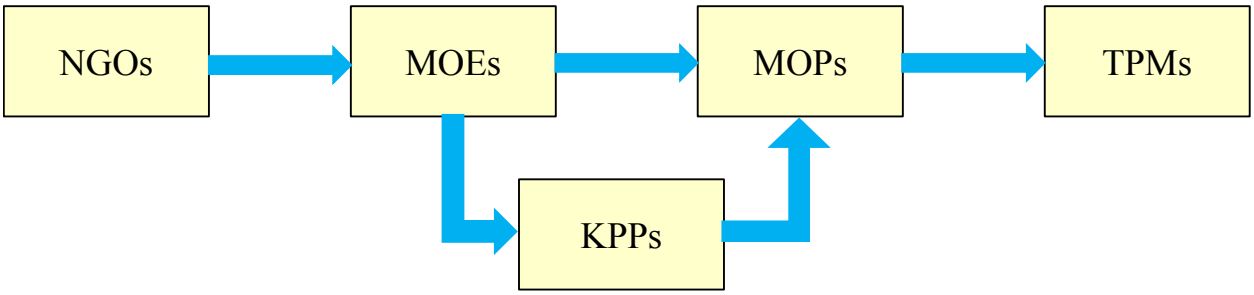


Figure 6.7-4 INCOSE Relationship between MOEs, MOPs, KPPs and TPMs

Technical Performance Measures

TPMs are critical or key mission success or performance parameters that are monitored during implementation by comparing the current actual achievement of the parameters with the values that were anticipated for the current time and projected for future dates. They are used to confirm progress and identify deficiencies that might jeopardize meeting a system requirement or put the project at cost, schedule, or performance risk. TPMs should be considered for each KPP and for selected MOEs or MOPs. The identified technical margins (see Section 7.8) can also be a source for a TPM. When a TPM value falls outside the expected range around the anticipated value, it signals a need for evaluation and corrective action.

In the systems engineering process, TPMs are used to do the following:

- Forecast values to be achieved by critical parameters at major milestones or key events during implementation.
- Identify differences between the actual and planned values for those parameters to contribute to system effectiveness assessments.
- Provide early warning for emerging risks requiring management attention (when negative margins exist).
- Provide early identification of potential opportunities to make design trades that reduce risk or cost, or increase system effectiveness (when positive margins exist).
- Support assessment of proposed design changes.
- Provide a means to assess operational performance against expected (as-designed) performance, including human-in-the-loop total system performance where applicable.

Selecting TPMs

TPMs should be considered for each KPP and for selected MOEs or MOPs. Understanding that TPM tracking requires allocation of resources, care should be exercised in selecting a small set of succinct TPMs that accurately reflect key parameters or risk factors, that are readily measurable, and that can be affected by altering design decisions. In general, TPMs can be

generic (attributes that are meaningful to each PBS element, like mass or reliability) or unique (attributes that are meaningful only to specific PBS elements). The systems engineer needs to decide which generic and unique TPMs are worth tracking at each level of the PBS. (See box for examples of TPMs.) At lower levels of the PBS, TPMs worth tracking can be identified through the functional and performance requirements levied on each individual system, subsystem, etc. The relationship is illustrated in Figure 6.7-5.

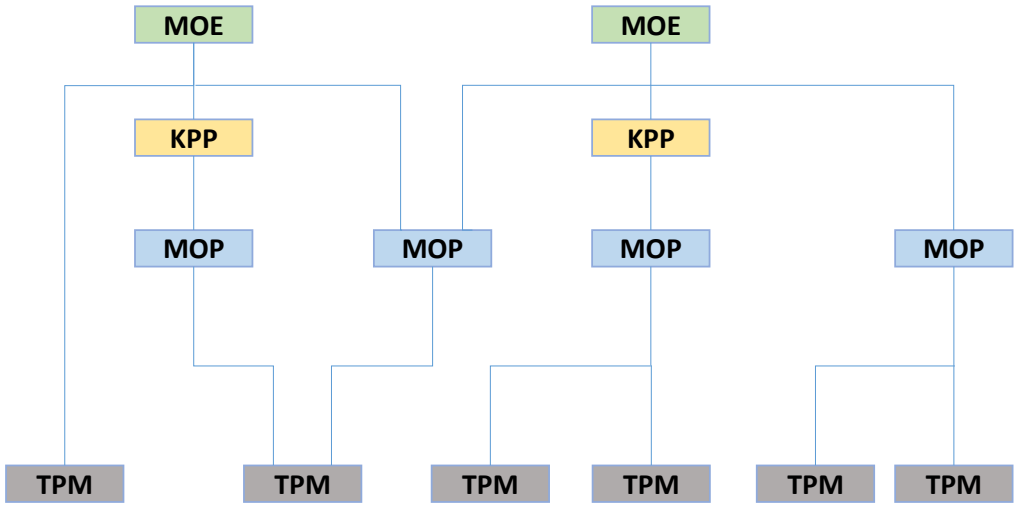


Figure 6.7-5 Example Flow of MOEs, MOPs, and TPMs

As TPMs are intended to provide an early warning of the adequacy of a design in satisfying selected critical technical parameter requirements, the systems engineer should select TPMs that fall within well-defined (quantitative) limits for reasons of system effectiveness or mission feasibility. Usually these limits represent either a firm upper or lower bound constraint. A typical example of such a TPM for a spacecraft is its injected mass, which should not exceed the capability of the selected launch vehicle. Tracking injected mass as a high-level TPM is meant to ensure that this does not happen. A high-level TPM like injected mass should often be "budgeted" and allocated to multiple system elements. Tracking and reporting should be required at these lower levels to gain visibility into the sources of any variances.

Examples of Technical Performance Measures

TPMs from MOEs

- Mission performance (e.g., total science data volume returned)
- Safety (e.g., probability of loss of crew, probability of loss of mission)
- Achieved availability (e.g., (system uptime)/(system uptime + system downtime))
- Ratio of crew time for mission to crew time for maintenance
- System handling qualities during human-in-the-loop operation

TPMs from Selected MOPs or KPPs

- Thrust versus predicted/specified
- Specific Impulse (Isp) versus predicted/specified
- End of Mission (EOM) dry mass
- Injected mass (includes EOM dry mass, baseline mission plus reserve propellant, other consumables and upper stage adaptor mass)
- Propellant margins at EOM
- Other consumables margins at EOM
- Electrical power margins over mission life
- Control system stability margins
- EMI/EMC susceptibility margins
- Onboard data processing memory demand
- Onboard data processing throughput time
- Onboard data bus capacity
- Total pointing error
- Total vehicle mass at launch
- Payload mass (at nominal altitude or orbit)
- Reliability
- Mean time before refurbishment required
- Total crew maintenance time required
- System turnaround time
- Fault detection capability
- Percentage of system designed for on-orbit crew access

In summary, for a TPM to be a valuable status and assessment tool, certain criteria should be met:

- Be a significant descriptor of the system (e.g., weight, range, capacity, response time, safety parameter) that will be monitored at key events (e.g., reviews, audits, planned tests).
- Can be measured (either by test, inspection, demonstration, or analysis).
- Is such that reasonable projected progress profiles can be established (e.g., from historical data or based on test planning).

Example of MOE / KPP / MOP / TPM Flow

- MOE: The data system does not fail when processing mission critical functions
- KPP: Reliability
- MOP: Derived from the MOE—System provides uninterrupted computing for at least 100 hours
- TPM: Failure rate

TPM Assessment and Reporting Methods

Status reporting and assessment of the system's TPMs complement cost and schedule control. There are a number of assessment and reporting methods that have been used on NASA projects, including the planned profile method and the margin management method.

A detailed example of the planned profile method for the Chandra Project weight TPM is illustrated in Figure 6.7-6. This figure depicts the subsystem contributions, various constraints, project limits, and management reserves from project SRR to launch.

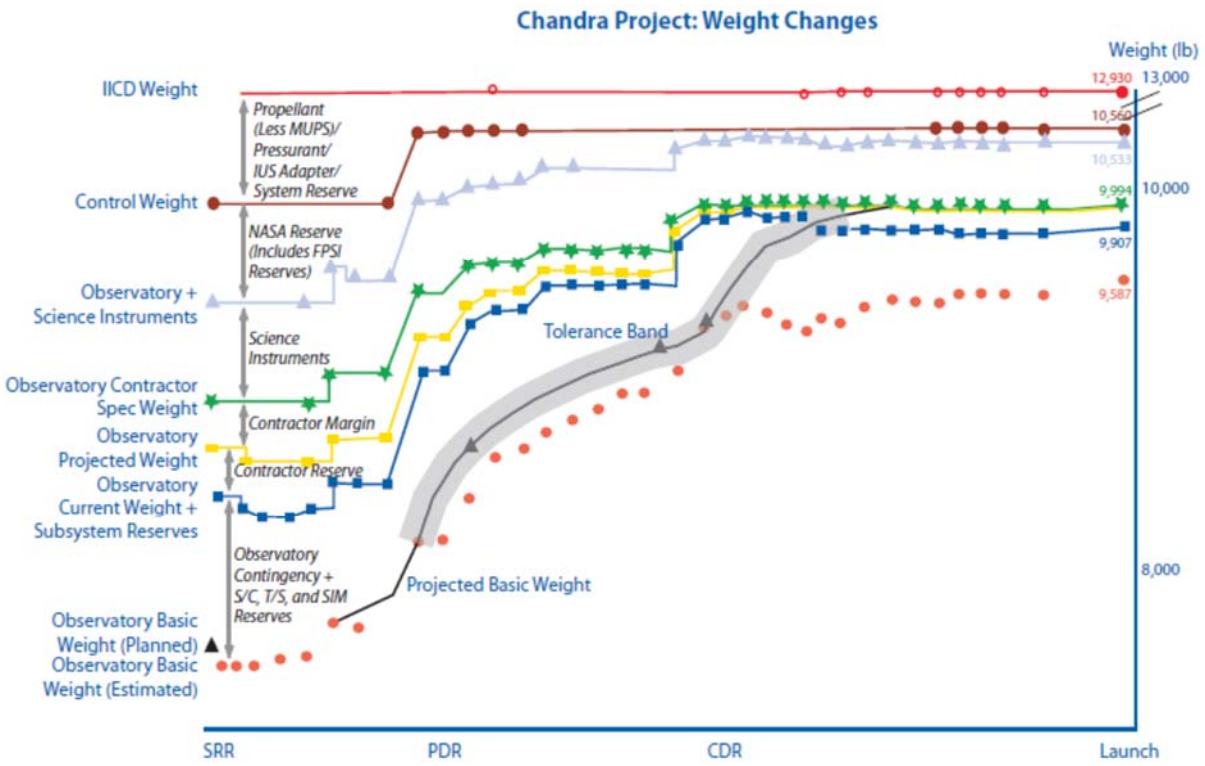


Figure 6.7‑6 Use of the Planned Profile Method for the Weight TPM with Rebaseline in Chandra Project

A detailed example of the margin management method for the Sojourner mass TPM is illustrated in Figure 6.7-7. This figure depicts the margin requirements (horizontal straight lines) and actual mass margins from project SRR to launch. For additional information on technical margins, see Section 7.8 of this document, which also provides additional information on margin management.

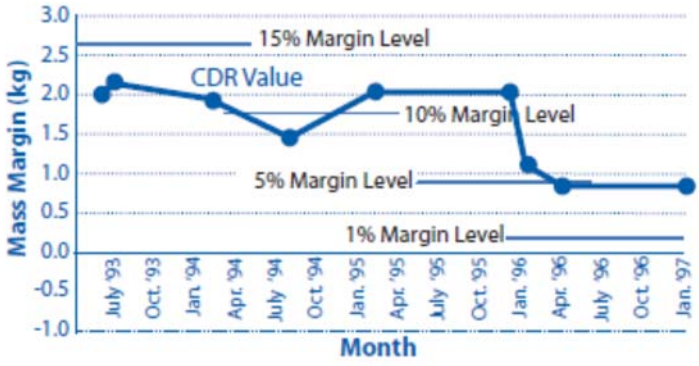


Figure 6.7‑7 Use of the Margin Management Method for the Mass TPM in Sojourner

Note: Current Margin Description: Microrover System (Rover + Lander-Mounted Rover Equipment (LMRE)) Allocation = 16.0 kg; Microrover System (Rover + LMRE) Current Best Estimate = 15.2 kg; Microrover System (Rover + LMRE) Margin = 0.8 kg (5.0%).

Evaluation and Monitoring of TPMs

The SEMP is the usual document for describing the project's TPM assessment program. Larger programs may have a separate technical metrics plan describing and defining the MOEs, MOPs, TPMs, and their relationships. This description should include a master list of those TPMs to be tracked, and the measurement and assessment methods to be employed. If analytical methods and models are used to measure certain high-level TPMs, then these need to be identified. The reporting frequency and timing of assessments should be specified as well. In determining these, the systems engineer should balance the project's needs for accurate, timely, and effective TPM tracking against the cost of the TPM tracking program.

The plan for assessing TPMs, which may be a part of the SEMP or a standalone document for large programs/projects, should specify each TPM's allocation, time-phased planned profile or margin requirement, and alert zones as appropriate to the selected assessment method.

A formal TPM assessment program should be fully planned and baselined with the SEMP. Tracking TPMs should begin as soon as practical in Phase B. Data to support the full set of selected TPMs may, however, not be available until later in the project life cycle. As the project life cycle proceeds through Phases C and D, the measurement of TPMs should become increasingly more accurate with the availability of more actual data about the system.

For the WBS model in the system structure, the following activities are typically performed:

- Analyze stakeholder expectation statements to establish a set of MOEs by which overall system or product effectiveness will be judged and customer satisfaction will be determined.
- Define MOPs for each identified MOE.
- Define appropriate TPMs and document the TPM assessment program in the SEMP.

6.7.2.6.3 Systems Engineering Process Metrics

Status reporting and assessment of systems engineering process metrics provide additional visibility into the performance of the "system that produces the system." As such, these metrics supplement the cost and schedule control measures discussed in Section 6.7.2.6.1.

Systems engineering process metrics try to quantify the effectiveness and productivity of the systems engineering process and organization. Within a single project, tracking these metrics allows the systems engineer to better understand the health and progress of that project. Across projects (and over time), the tracking of systems engineering process metrics allows for better estimation of the cost and time of performing systems engineering functions. It also allows the systems engineering organization to demonstrate its commitment to continuous improvement.

Technical Leading Indicators

This section discusses, in general, what technical leading indicators (TLI) are and how they are used. General guidance on how the three required indicators are gathered and reported is also provided.

What is a Leading Indicator?

A “leading indicator” is an individual measure or collection of measures that has been shown to be predictive with respect to program/project performance, cost, or schedule in subsequent life-cycle phases including process metrics. It is one of, or a subset of, all the parameters that might be monitored on a program/ project. Some leading indicators, such as mass margin, are also often considered Technical Performance Measures (TPMs). TPMs help the program/project keep track of their most critical parameters and may be either leading (predictive of the future state) or lagging (capturing historical data only).

A leading indicator is used to determine both the maturity and stability of the design, development, and operational phases of a program/project. The goal of the indicators is to provide insight into potential future states to allow management to take action before problems are realized. While the program/project may select several leading indicators to monitor their activities, NPR 7123.1 requires three: Mass Margins, Power Margins and RFA/RID/Action Item burndown.

How are they used?

Leading indicators are always viewed against time to see how the measure is progressing so that the products stability and maturity can be determined. This may be depicted in tabular form, or perhaps more usefully in graphical form. The parameters may also be plotted against planned values and/or upper or lower limits. These limits, if used, would likely be defined by the Center, program, or project based on historical information.

By monitoring these trends, the program or project managers, systems engineers, other team members, and management can more accurately assess the health, stability, and maturity of their program/project and predict future problems that might require their attention and mitigation before they become too costly. By combining these periodic trending indicators with the life-cycle review entrance and success criteria, the program/project will have better insight into whether its products are reaching the right maturity levels at the right point in the life cycle, as well as an indication of the stability of those designs. The entrance criteria in particular address the maturity levels of both the end product as well as the project documentation that captures its design. However, just looking at maturity levels is not sufficient. If, for example, the product is appropriately designed to a CDR maturity level, but there are still significant changes in the requirements, the project cannot be considered stable. The indicators will aid in the understanding of both the maturity and stability of the program/project.

Gathering Data

These three leading indicators (Mass Margins, Power Margins, and RFA/RID/Action Item burndown) need to be gathered and reported throughout the program/project life cycle, starting after SRR and continuing through SAR. They are normally gathered by the project at one or more levels within the product hierarchy. The requirement is to report externally at the program/project level, however, it will likely be useful for the project to gather the data one or more levels deeper within the product hierarchy so that the project may determine how best to allocate power and mass resources. Tightly coupled programs will need to gather the parameters from the various projects and provide a rolled-up set of parameters for reporting. The parameters

for single-project programs and loosely coupled programs will remain separate.

Note that the leading indicators may exist first in an estimated form. Then as more information is available, they are refined or converted into actual measured values. For example, early in Phase A, the mass of a product may be estimated through modeling methods whereas later in the life cycle when the product is being built, measured masses can be used. Wherever the program / project is in its life cycle, the goal is to provide the most current and accurate information possible.

Reporting the Data

This information is reported to program/project managers, and may be used for reporting, as required, to internal and external stakeholders (e.g., Associate Administrator, OMB, etc.).

As a minimum, the following characteristics should be provided as part of the report:

1. Data is to be presented as a trend. As such, it will need to be in graphical or tabular form reported against time (month) for multiple time periods (not just the parameters for the current month) as appropriate for the program/project.
2. If in graphical form, a data table needs to be provided as part of the graph.
3. Milestone reviews need to be indicated on the graph or as part of the table for reference.
4. Graphs or tables are to be annotated with informative information as needed to explain key features.

Required Indicators

This section provides detailed guidance on each of the required leading indicators and review trends. Each section will discuss why trends in this area are important and describe the specific leading indicator measurements or review trend that need to be gathered, monitored, and reported. Each measurement will be described and sample plots will be shown. Note that these are only examples. These measurements may be portrayed by the project in the format that is most useful to the project. A spreadsheet template is also provided with an easy way to track and display these parameters.

RID/RFA/Action Item Burndown per Review

This review trend is *required* per NPR 7123.1. During a life-cycle review, comments are usually solicited from the reviewing audience. Depending on the program/project, they may be gathered as Requests for Action (RFAs), Review Item Discrepancies (RIDs), and/or Action Items. How review comments are to be requested and gathered should be determined by the program/project early in Phase A and documented in their Formulation Agreement and/or program or project plan. Typically, if RFAs or RIDs are to be reported, action items are not. If the program/project will only be tracking Action Items at life-cycle reviews, then they are expected to be reported.

The information that needs to be gathered for this review trend is:

1. Total number of RFAs, RIDs or Action Items (whichever the program/project uses).
2. Number of open RFAs, RIDs or Action Items (whichever the program/project uses).

3. Planned burndown rate.

Table 6.7-5 shows an example spreadsheet that is tracking the number of RFAs for a given life-cycle review. Note that the information is gathered monthly, not just at the next life-cycle review. Trending works best if the parameters are taken regularly.

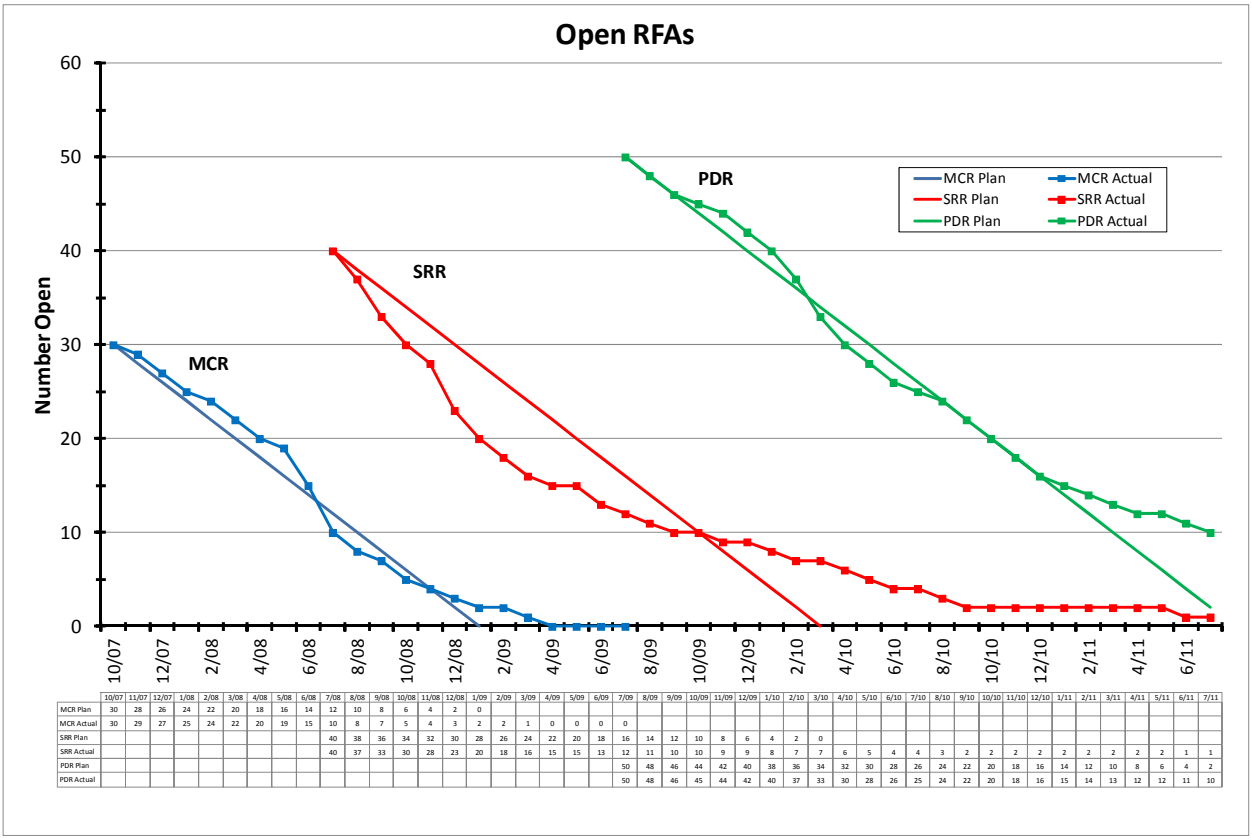
The *planned* burndown rate of the RFAs is included. While it can be difficult to predict how negotiations with submitters will progress, having a plan is important to communicate the desire of the program/project to resolve issues in a timely manner. Having plans and periodically providing status against those plans helps to provide incentives to all parties to address and resolve issues.

Table 6.7-5 Number of Open RFAs per Review for Project

Date	MCR		SRR		PDR	
	Plan	Actual	Plan	Actual	Plan	Actual
Total	35		40		55	
10/07	30	30				
11/07	28	29				
12/07	26	27				
1/08	24	25				
2/08	22	24				
3/08	20	22				
4/08	18	20				
5/08	16	19				
6/08	14	15				
7/08	12	10	40	40		
8/08	10	8	38	37		
9/08	8	7	36	33		
10/08	6	5	34	30		
11/08	4	4	32	28		
12/08	2	3	30	23		
1/09	0	2	28	20		
2/09		2	26	18		
3/09		1	24	16		
4/09		0	22	15		
5/09		0	20	15		
6/09		0	18	13		
7/09		0	16	12	50	50
8/09			14	11	48	48
9/09			12	10	46	46
10/09			10	10	44	45

This information can perhaps be better seen graphically. Figure 6.7-8 is an example. The appropriate number of reporting months on any given graph should be agreed to by the

program/project manager, OCE, and other decision authorities and documented in the Formulation Agreement and/or program/project plan.



	10/07	11/07	12/07	1/08	2/08	3/08	4/08	5/08	6/08	7/08	8/08	9/08	10/08	11/08	12/08	1/09	2/09	3/09	4/09	5/09	6/09	7/09	8/09	9/09	10/09	11/09	12/09	1/10	2/10	3/10	4/10	5/10	6/10	7/10	8/10	9/10	10/10	11/10	12/10	1/11	2/11	3/11	4/11	5/11	6/11	7/11
MCR Plan	30	28	26	24	22	20	18	16	14	12	10	8	6	4	2	0																																
MCR Actual	30	29	27	25	24	22	20	19	15	10	8	7	5	4	3	2	2	1	0	0	0	0																										
SRR Plan										40	38	36	34	32	30	28	26	24	22	20	18	16	14	12	10	8	6	4	2	0																				
SRR Actual										40	37	33	30	28	23	20	18	16	15	15	13	12	11	10	10	9	9	8	7	7	6	5	4	4	3	2	2	2	2	2	2	2	2	1	1
PDR Plan																						50	48	46	44	42	40	38	36	34	32	30	28	26	24	22	20	18	16	14	12	10	8	6	4	2
PDR Actual																						50	48	46	45	44	42	40	37	33	30	28	26	25	24	22	20	18	16	15	14	13	12	12	11	10

Figure 6.7-8 Number of Open RFAs per Review

Mass Margin Leading Indicator

For program/projects that contain hardware, mass margin is a required leading indicator to be gathered and reported. Mass margin data is usually shown along with upper and/or lower tolerance bands, requirement levels, and perhaps stretch goal indications. When the indicator goes outside of the tolerance bands, more attention may be required by the program/project to understand what is happening with that indicator and to determine if corrective action is warranted.

For virtually every program/project that produces a product intended to fly into space, mass is a critical parameter. It is also critical for many projects that do not fly into space. Programs/projects may choose to track raw mass, but perhaps a more informative indication is that of mass margins. As the concepts are analyzed, a determination of how much a new launch vehicle can lift to the desired orbit/destinations will be determined. This in turn will place limits on the products that the program/projects may be providing. Mass parameters will be flowed down and allocated to the subsequent systems, subsystems, and components within the program/project's product. Tracking the ability of the overall product and its lower-level components to accomplish mass goals is critical to ensuring the success of the program/project. It is left to the program/project to determine how far down into the product hierarchy the mass

margin indicator is to be tracked. As a minimum, reporting at the rolled-up program/project level will be required.

The information that needs to be gathered for this leading indicator is as follows:

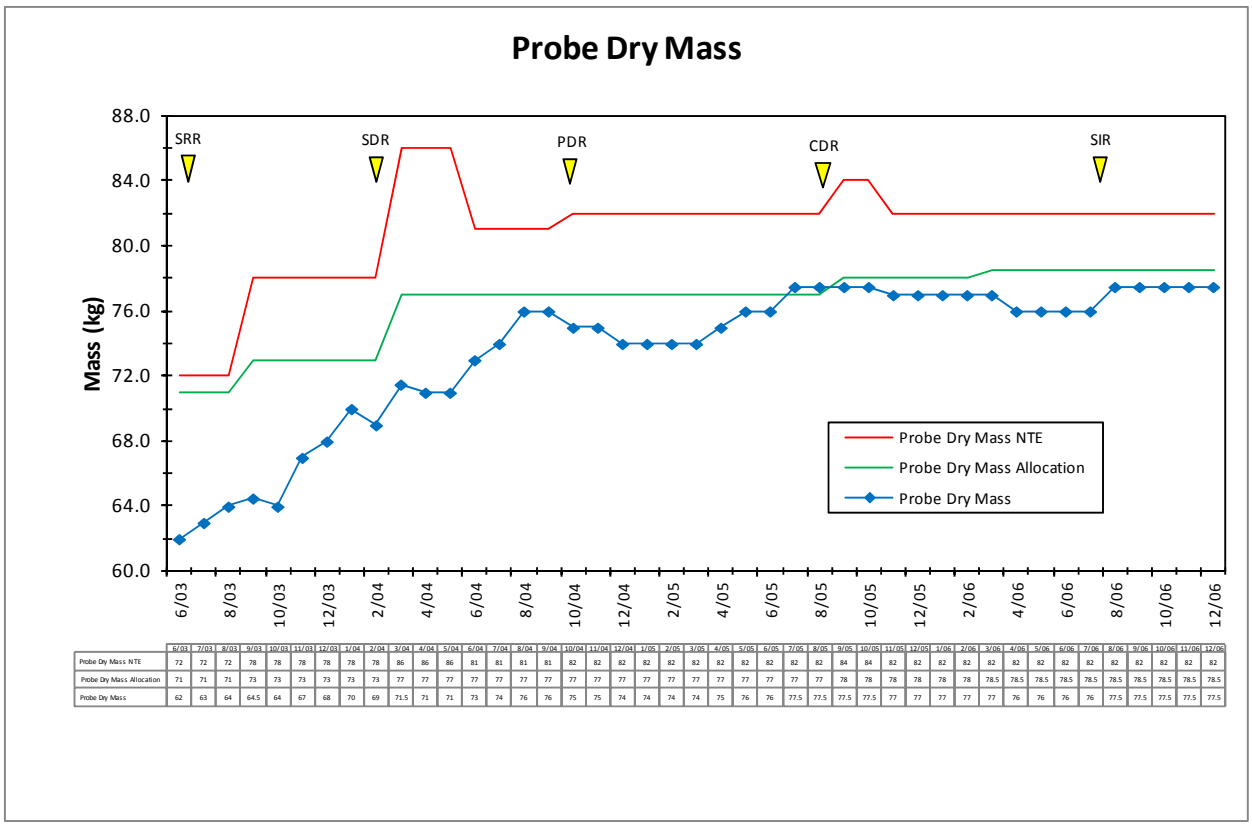
1. Current estimated and/or measured mass;
2. Not-to-exceed mass; and
3. Mass allocation.

Table 6.7-6 is an example of a spreadsheet that is tracking the mass margin of a project. The margins are estimated prior to the actual manufacturing of the product and measured after it is produced. When reporting at the overall program/project level, the parameter may be a combination of estimates for subsystems/components that have not yet been produced and measured mass of those that have been produced. Note that "not-to-exceed" mass and/or allocations can be adjusted during the program/project life cycle as the program/project balances their overall mass reserves.

Table 6.7-6 Example Spreadsheet for Tracking Mass Margin

Date	NTE (kg)	Allocation (kg)	Current Mass (kg)
6/03	72	71	62
7/03	72	71	63
8/03	72	71	64
9/03	78	73	64.5
10/03	78	73	64
11/03	78	73	67
12/03	78	73	68
1/04	78	73	70
2/04	78	73	69
3/04	86	77	71.5
4/04	86	77	71
5/04	86	77	71
6/04	81	77	73
7/04	81	77	74
8/04	81	77	76

Perhaps a more effective way of displaying the mass margin indicator is graphically. Figure 6.7-9 is an example of displaying the mass margins for a particular program/project. Note that symbols are included on the graph to indicate where the life-cycle reviews were held. This helps put the information into the proper context. Also, enough information needs to be on the graph or in the legend to properly identify what mass is being tracked (dry mass, wet mass, mass of just the instrument, or mass of the entire vehicle, etc.).



	6/03	7/03	8/03	9/03	10/03	11/03	12/03	1/04	2/04	3/04	4/04	5/04	6/04	7/04	8/04	9/04	10/04	11/04	12/04	1/05	2/05	3/05	4/05	5/05	6/05	7/05	8/05	9/05	10/05	11/05	12/05	1/06	2/06	3/06	4/06	5/06	6/06	7/06	8/06	9/06	10/06	11/06	12/06
Probe Dry Mass NTE	72	72	72	78	78	78	78	78	78	86	86	86	81	81	81	81	82	82	82	82	82	82	82	82	82	82	82	84	84	82	82	82	82	82	82	82	82	82	82	82	82	82	82
Probe Dry Mass Allocation	71	71	71	73	73	73	73	73	73	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	78	78	78	78	78	78	78.5	78.5	78.5	78.5	78.5	78.5	78.5	78.5	78.5	78.5
Probe Dry Mass	62	63	64	64.5	64	67	68	70	69	71.5	71	71	73	74	76	76	75	75	74	74	74	74	75	76	76	77.5	77.5	77.5	77.5	77	77	77	77	77	76	76	76	76	77.5	77.5	77.5	77.5	77.5

Figure 6.7-9 Example Plot for Mass Margin Indicator

Power Margin Leading Indicator

As with the mass margin indicator, tracking the power margins of a program/project that uses power is usually also a critical factor. Availability of solar, nuclear, battery, or other power sources will also place a limit on how much power the various systems may consume. The program/project will determine how deep within their product hierarchy these parameters will need to be allocated and tracked. As a minimum, the rolled-up power margin indicator will be reported as part of the status. For consistency through the product life cycle, the program/project may choose to define what the total power consumption includes (e.g., all systems operating at full power level simultaneously plus emergency and contingency systems) and utilize it as a base to derive the power profile used in the periodic power margin reports.

The information that needs to be gathered for this leading indicator is as follows:

1. Current estimated power consumption;
2. Not-to-exceed power consumption (source limit); and
3. Power allocation.

Table 6.7-7 is an example of a spreadsheet that is tracking the power margin of a project. The margins are estimated prior to the actual manufacturing of the product, and measured after it is produced. When reporting at the overall program/project level, the parameter may be a

combination of estimates for subsystems/components that have not yet been produced and measured mass of those that have been produced.

Table 6.7-7 Example Spreadsheet for Tracking Power Margin

Date	NTE (W)	Allocation (W)	Current Consumption (W)
6/03	43	32	28
7/03	43	32	28
8/03	43	32	28
9/03	43	32	27
10/03	43	32	28
11/03	43	32	27.5
12/03	43	32	30
1/04	39	32	29
2/04	39	32	29
3/04	39	32	29
4/04	39	32	29
5/04	39	32	31
6/04	39	32	31

Perhaps a more effective way of displaying the power margin indicator is graphically. Figure 6.7-10 is an example of displaying the power margins for a particular program/project. Note that symbols are included on the graph to indicate when the life-cycle reviews were held. This helps put the information into the proper context. Also, enough information needs to be on the graph or in the legend to properly identify what power is being tracked (e.g., total capacity, instrument power, vehicle power, orbit average, etc.).

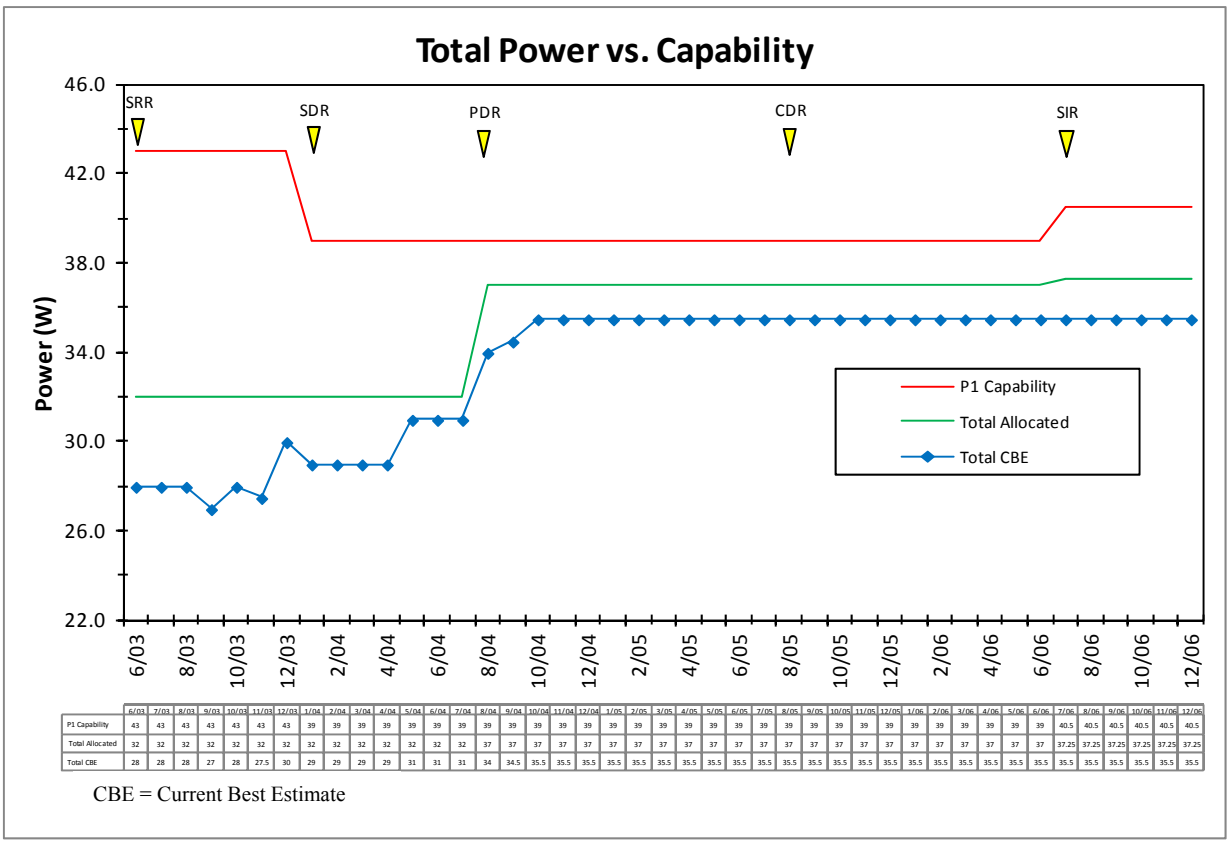


Figure 6.7-10 Example Plot for Power Margin Indicator

An Excel spreadsheet is provided on the NASA Engineering Network (NEN) Systems Engineering Community of Practice (SECoP) under Tools and Methods at https://nen.nasa.gov/web/se/tools/ and then NASA Tools & Methods. The spreadsheet can be used to gather and display these parameters if desired.

Selecting Other Systems Engineering Process Metrics

Generally, systems engineering process metrics fall into three categories: those that measure the progress of the systems engineering effort, those that measure the quality of that process, and those that measure its productivity. Different levels of systems engineering management are generally interested in different metrics. For example, a project manager or lead systems engineer may focus on metrics dealing with systems engineering staffing, project risk management progress, and major trade study progress. A subsystem systems engineer may focus on subsystem requirements and interface definition progress and verification procedures progress. It is useful for each systems engineer to focus on just a few process metrics. Which metrics should be tracked depends on the systems engineer's role in the total systems engineering effort. The systems engineering process metrics worth tracking also change as the project moves through its life cycle.

Collecting and maintaining data on the systems engineering process is not without cost. Status reporting and assessment of systems engineering process metrics divert time and effort from the activity itself. The systems engineer should balance the value of each systems engineering

process metric against its collection cost. The value of these metrics arises from the insights they provide into the activities that cannot be obtained from cost and schedule control measures alone. Over time, these metrics can also be a source of hard productivity data, which are invaluable in demonstrating the potential returns from investment in systems engineering tools and training.

Examples and Assessment Methods

Table 6.7-8 lists some systems engineering process metrics to be considered. This list is not intended to be exhaustive. Additional information can be found in INCOSE-TP-2003-020-01, *Technical Measurement*. Because some of these metrics allow for different interpretations, each NASA Center needs to define them in a commonsense way that fits its own processes. For example, each field Center needs to determine what is meant by a "completed" versus an "approved" requirement, or whether these terms are even relevant. As part of this definition, it is important to recognize that not all requirements, for example, need be lumped together. It may be more useful to track the same metric separately for each of several different types of requirements.

Table 6.7-8 Systems Engineering Process Metrics

Function	Metric	Category
Requirements development and management	Requirements identified versus completed versus approved	S
	Requirements volatility	Q
	Trade studies planned versus completed	S
	Requirements approved per systems engineering hour	P
	Tracking of TBAs, TBDs, and TBRs (to be announced, determined, or resolved) resolved versus remaining	S
Design and development	Specifications planned versus completed	S
	Processing of engineering change proposals (ECPs)/engineering change requests (ECRs)	Q
	Engineering drawings planned versus released	S
Verification and validation	Verification and validation plans identified versus approved	S
	Verification and validation procedures planned versus completed	S
	Functional requirements approved versus verified	S
	Verification and validation plans approved per systems engineering hour	P
	Processing of problem/failure reports	Q
	Percentage of design validated against ConOps	S
Reviews	Processing of RIDs	Q
	Processing of action items	Q

S = progress- or schedule-related; Q = quality-related; P = productivity-related

Quality-related metrics should serve to indicate when a part of the systems engineering process is overloaded and/or breaking down. These metrics can be defined and tracked in several different ways. For example, requirements volatility can be quantified as the number of newly identified requirements or as the number of changes to already approved requirements. As another

example, Engineering Change Request (ECR) processing could be tracked by comparing cumulative ECRs opened versus cumulative ECRs closed, or by plotting the age profile of open ECRs, or by examining the number of ECRs opened last month versus the total number open. The systems engineer should apply his or her own judgment in picking the status reporting and assessment method.

System productivity metrics provide an indication of the throughput of the system through production and operations or the progress of the system through development. Production and operations metrics are system-specific and depend on the complexity of the program, project, or activity. The number of units produced per year is one possible measure of system progress. Operationally, the number of experiments conducted per month or year is another type. During development, productivity is related to the progression of the system through development. This is reflected in the decision board throughput (e.g., number of board decisions per month) and the RID/RFAs accepted and resolved for each technical review as discussed above.

Schedule-related metrics can be depicted in a table or graph of planned quantities versus actuals, for example, comparing planned number of verification closure notices against actual. This metric should not be confused with EVM described in Section 6.7.2.6.1. EVM is focused on integrated cost and schedule at the desired level, whereas this metric focuses on an individual process or product within a subsystem, system, or project itself.

The combination of quality, productivity, and schedule metrics can provide trends that are generally more important than isolated snapshots. The most useful kind of assessment method allows comparisons of the trend on a current project with that for a successfully completed project of the same type. The latter provides a benchmark against which the systems engineer can judge his or her own efforts.

Earned Value Management (EVM)

EVM is a disciplined project management process that integrates a project's technical requirements with schedule and cost elements. For more information on EVM, refer to Section 6.7.2.6.1 of this guide.

6.8 Decision Analysis

The purpose of this section is to provide an overview of the Decision Analysis Process, highlighting selected tools and methodologies. Decision Analysis is a framework within which analyses of diverse types are applied to the formulation and characterization of decision alternatives that best implement the decision-maker's priorities given the decision-maker's state of knowledge.

The Decision Analysis Process is used in support of decision making bodies to help evaluate technical, cost, and schedule issues, alternatives, and their uncertainties. Decision models have the capacity for accepting and quantifying human subjective inputs: judgments of experts and preferences of decision makers.

Early in the project life cycle, high-level decisions are made regarding which technology could be used, such as solid or liquid rockets for propulsion. Operational scenarios, probabilities, and consequences are determined and the design decision is made without specifying the component-level detail of each design alternative. Once high-level design decisions are made, nested systems engineering processes occur at progressively more detailed design levels flowed down through the entire system. Each progressively more detailed decision is affected by the assumptions made at the previous levels. For example, the solid rocket design is constrained by the operational assumptions made during the decision process that selected that design. This is an iterative process among elements of the system**.**

Typical processes that use decision analysis are as follows:

- Determining how to allocate limited resources (e.g., budget, mass, power) among competing subsystem interests to favor the overall outcome of the project;
- Select and test evaluation methods and tools against sample data;
- Configuration management processes for major change requests or problem reports;
- Design processes for making major design decisions and selecting design approaches;
- Key decision point reviews or technical review decisions (e.g., PDR, CDR) as defined in NPR 7120.5 and NPR 7123.1;
- Go or no-go decisions (e.g., FRR):
  - Go—authorization to proceed; or
  - No-go—repeat some specific aspects of development or conduct further development or test.
- Project management of major issues, schedule delays, or budget increases;
- Procurement of major items;
- Technology decisions;
- Risk management of major risks (e.g., red or yellow);
- SMA decisions; and

- Miscellaneous decisions (e.g., whether to intervene in the project to address an emergent performance issue).

Decision Analysis applies across the full system life cycle from pre-formulation through decommissioning. Decision-making can be simple or complex. For programs, large projects, and large activities, a variety of tools and methods are available to support integration of relevant information for complex decision-making. These tools and methods are not limited to larger projects but will be more widely applied in these programs, projects, and activities. In addition, for small projects or activities, decisions may be effectively made through discussion among a small number of informed discipline engineers and project management. There are many factors that go into decision-making including the function, operations, and physics of the problem, budgetary constraints, schedule constraints, as well as policy and law constraints. All of these factors should be accounted for in the decision-making process.

Much of the academic literature of decision analysis is written as if the analysis were to be done by the decision-maker or by analysts who are essentially working for the decision-maker. However, the situation is more complicated when analysis is furnished by one organizational unit to a decision-maker in another organizational unit. This occurs whenever an organization is providing an analysis or a system to an acquiring organization that must then decide whether to accept it, which is in part a risk-acceptance decision. In such a case, that risk-acceptance decision must be predicated not only on the acquiring entity's priorities but also on its state of knowledge; i.e., its uncertainty. Moreover, in a case where risks to humans are involved, NASA policy calls for the actual risk-takers to accept the risk. These circumstances shape the present recommendations for documentation of the technical basis for important decisions and for a considered approach to the presentation of information about uncertainty.

Decision-making structure. The decision maker can range from the program/project manager, chief engineer, or line manager to formal decision bodies such as control boards or working groups. These include both formal decision authorities and delegated decision-makers. The decision-making structure must be properly organized to be effective and efficient. When a hierarchy of decision authorities is used, these bodies (i.e., boards, working groups) must have clear scopes that fit with the system under development. If too many decision-making bodies are employed, decision-making can become extremely slow and limited by available expertise to represent all required disciplines in these decisions. The number and relationship of decision-making bodies should follow the program or project type for the system being developed. For loosely coupled programs, each project may require a separate decision body that feeds a higher level body at the program level. For tightly coupled projects, a single decision body at the program level may be optimal. Small projects or activities may not need large decision bodies and can have decision-making structures where all relevant disciplines meet with the project manager and chief engineer to make a decision.

Having the right expertise represented in the decision-making body is essential. The systems engineer must ensure that all affected or contributing expertise are members of the decision-making body or in attendance at the decision meeting. Each engineering discipline, Safety and Mission Assurance, procurement office (as applicable), and the program or project business office should be represented in the decision body.

Information theory provides a good understanding of decision-making processes and a set of rules in organizing decision-making bodies. The decision-maker and each board member (or decision-making participant) is a source of information to the board. The communication between these members during the board discussion is the communication channel between the members. All information necessary for the decision must be represented in the membership. A range of information that is not complete cannot be mapped properly to the full domain of solutions and the information provided will not fully support the decision. This leads to decisions based on assumed understanding (a source of uncertainty or noise) of the question being decided. This uncertainty is a result of bias that is an element leading to unintended consequences as a result of partially informed decisions. There are many instances where information not presented or withheld for some reason has led to failures of some type in the mission.

Central to considering the distribution of boards is the scope of each board. Board scopes must comprise separate sets of information in which they work. Each decision-making body should have a clear and non-overlapping scope with other decision-making bodies in the program, project, or activity. Scopes that overlap will generate much more uncertainty and effort in decision-making leading to confusion, disagreement, delays, poor decision quality, or the inability to make a decision. Note that these overlapping structures also create a work force skill drain where multiple representatives are needed to cover all of the boards discussing the same information.

On the other hand, scopes which are separate, operating on different information sets, can be distributed to different boards. Generally, a top level decision body may still be needed with representation from each of the lower boards (such as the case of a Program Control Board) to integrate decisions. This is illustrated in loosely coupled projects where each functions somewhat separately from the other and the program provides the integrating function. Again, understanding the set of information or scope that each board has, the relationship between the scopes is essential to setting an efficient decision-making structure.

These basic rules governing the structuring of decision bodies are as follows:

- Understand and define the scope of each needed decision body.
- Ensure that each decision body has all affected or contributing disciplines represented, including understanding of the types and magnitudes of uncertainties affecting decisions within that decision body's scope.
- Minimize the number of decision bodies based on scope. The efficiency of the structure decreases with distributed and overlapping scopes.

The methodology should always be adapted to the system context and issue under consideration. The problem is structured by first stating the problem on a clear physical, logical, financial, and chronological basis. Program, project, or activity goals are identified as guidance to the decision-making process. Once the problem is understood within the context of the physical system and program/project/activity structure, alternatives are defined to meet these decision criteria. In systems where the decision criteria are complex and interact with each other, various decision analysis techniques discussed in the following subsections can be applied.

An important aspect of the Decision Analysis Process is to consider and understand when it is appropriate or required for a decision to be made or not made. When considering a decision, it is important to ask questions such as: Why is a decision required at this time? For how long can a decision be delayed? What is the impact of delaying a decision? Is all of the necessary information available to make a decision? Are there other key drivers or dependent factors and criteria that should be in place before a decision can be made? What kind of evaluation will be needed to gather sufficient data to make the decision?

The outputs from this process support the decision authority's difficult task of deciding among competing alternatives without complete knowledge; therefore, it is critical to understand and document the assumptions and limitation of any tool or methodology and integrate them with other factors when deciding among viable options.

6.8.1 Process Description

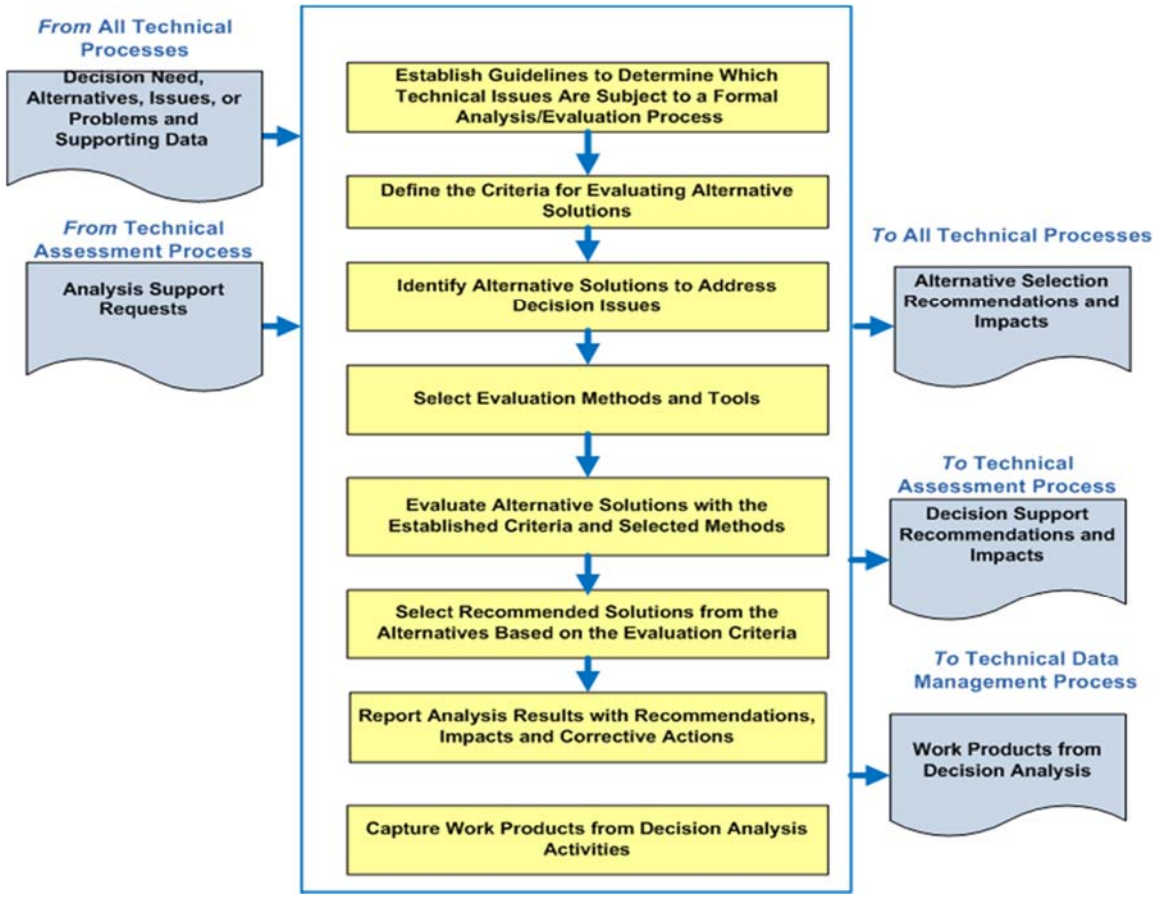


Figure 6.8-1 Decision Analysis Process

A typical process flow diagram is provided in Figure 6.8-1, including inputs, activities, and outputs. The first step in the process is understanding the decision to be made in the context of the system/mission. Understanding the decision needed requires knowledge of the intended outcome in terms of technical performance, cost, and schedule. For an issue that follows the

decision analysis process, the definition of the decision criteria or the measures that are important to characterize the options for making a decision should be the next step in the process. With this defined, a set of alternative solutions can be defined for evaluation. These solutions should cover the full decision space as defined by the understanding of the decision and definition of the decision criteria. The need for specific decision analysis tools (defined in Section 6.8.3 below) can then be determined and employed to support the formulation of a solution. Following completion of the analysis, a description of how each alternative compares with the decision criteria can be captured for submission to the decision-making body or authority. A recommendation is typically provided from the decision analysis, but is not always required depending on the discretion of the decision-making body. A decision analysis report should be generated including: decision to be made, decision criteria, alternatives, evaluation methods, evaluation process and results, recommendation, and final decision.

Decision analysis covers a wide range of timeframes. Complex, strategic decisions may require weeks or months to fully assess all alternatives and potential outcomes. Decisions can also be made in hours or in a few days, especially for smaller projects or activities. Decisions are also made in emergency situations. Under such conditions, process steps, procedures, and meetings may be combined. In these cases, the focus of the systems engineer is on obtaining accurate decisions quickly. Once the decision is made, the report can be generated. The report is usually generated in an ongoing fashion during the decision analysis process. However, for quick or emergency decisions, the report information may be captured after the decision has been made.

Not all decisions require the same amount of analysis effort. The level and rigor required in a specific situation depend essentially on how clear-cut the decision is. If there is enough uncertainty in the alternatives' performance that the decision might change if that uncertainty were to be reduced, then consideration needs to be given to reducing that uncertainty. A robust decision is one that is based on sufficient technical evidence and characterization of uncertainties to determine that the selected alternative best reflects decision-maker preferences and values given the state of knowledge at the time of the decision. This is suggested in Figure 6.8-2 below.

Note that in Figure 6.8-2, the phrase "net beneficial" in the decision node "Net beneficial to reduce uncertainty?" is meant to imply consideration of all factors, including whether the project can afford any schedule slip that might be caused by additional information collection and additional analysis.

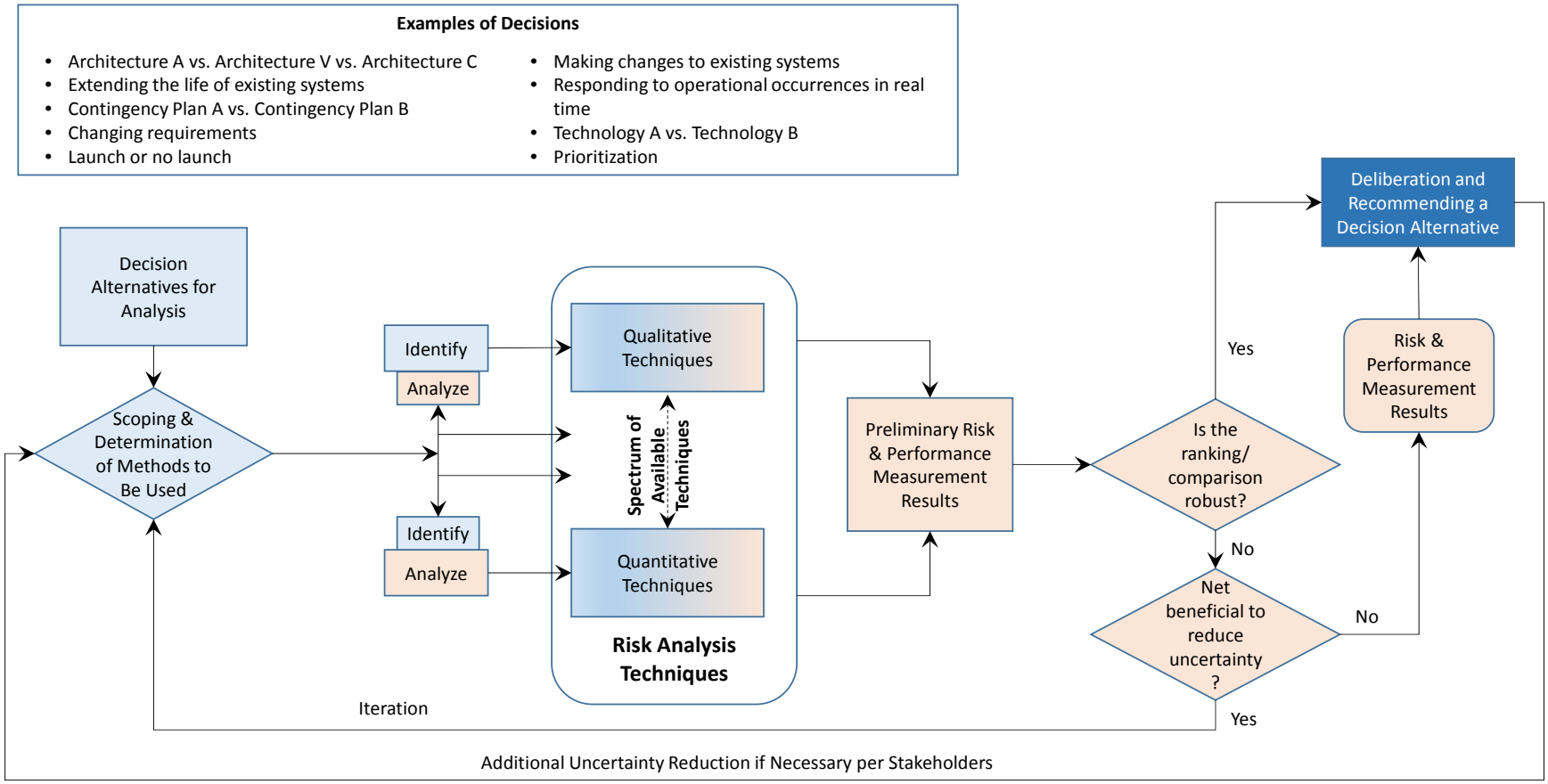


Figure 6.8-2 Risk Analysis of Decision Alternatives

6.8.1.1 Inputs

The technical, cost, and schedule inputs need to be comprehensively understood as part of the general decision definition. Based on this understanding, decision making can be addressed from a simple meeting to a formal analytical analysis. As illustrated in figure 6.8-2, many decisions do not require extensive analysis and can be readily made with clear input from the responsible engineering and programmatic disciplines. Complex decisions may require more formal decision analysis when contributing factors have complicated or not well defined relationships. Due to this complexity, formal decision analysis has the potential to consume significant resources and time. Typically, its application to a specific decision is warranted only when some of the following conditions are met:

- **Complexity:** The actual ramifications of alternatives are difficult to understand without detailed analysis;
- **Uncertainty:** Uncertainty in key inputs creates substantial uncertainty in the ranking of alternatives and points to risks that may need to be managed;
- **Multiple Attributes:** Greater numbers of attributes cause a greater need for formal analysis; and
- **Diversity of Stakeholders:** Extra attention is warranted to clarify objectives and formulate TPMs when the set of stakeholders reflects a diversity of values, preferences, and perspectives.

Satisfaction of all of these conditions is not a requirement for initiating decision analysis. The point is, rather, that the need for decision analysis increases as a function of the above conditions. In addition, often these decisions have the potential to result in high stakes impacts to cost, safety, or mission success criteria, which should be identified and addressed in the process. When the Decision Analysis Process is triggered, the following are inputs:

- **Decision need, identified alternatives, issues, or problems and supporting data:** This information would come from all technical, cost, and schedule management processes. It may also include high-level objectives and constraints (from the program/project).
- **Analysis support requests**: Requests will arise from the technical, cost, and schedule assessment processes.

6.8.1.2 Process Activities

For the Decision Analysis Process, the following activities are typically performed.

It is important to understand the decision needed in the context of the mission and system, which requires knowledge of the intended outcome in terms of technical performance, cost, and schedule. A part of this understanding is the definition of the decision criteria, or the measures that are important to characterize the options for making a decision. The specific decision-making body, whether the program/project manager, chief engineer, line management, or control board should also be well defined. Based on this understanding, then the specific approach to decision-making can be defined.

Decisions are based on facts, qualitative and quantitative data, engineering judgment, and open communications to facilitate the flow of information throughout the hierarchy of forums where technical analyses and evaluations are presented and assessed and where decisions are made. The extent of technical analysis and evaluation required should be commensurate with the consequences of the issue requiring a decision. The work required to conduct a formal evaluation is significant and applicability should be based on the nature of the problem to be resolved. Guidelines for use can be determined by the magnitude of the possible consequences of the decision to be made.

6.8.1.2.1 Define the Criteria for Evaluating Alternative Solutions

This step includes identifying the following:

- The types of criteria to consider, such as customer expectations and requirements, technology limitations, environmental impact, safety, risks, total ownership and life-cycle costs, and schedule impact;
- The acceptable range and scale of the criteria; and
- The rank of each criterion by its importance.

Decision criteria are requirements for individually assessing the options and alternatives being considered. Typical decision criteria include cost, schedule, risk, safety, mission success, and supportability. However, considerations should also include technical criteria specific to the decision being made. Criteria should be objective and measurable. Criteria should also permit differentiating among options or alternatives. Some criteria may not be meaningful to a decision; however, they should be documented as having been considered. Criteria may be mandatory (i.e., "shall have") or enhancing. An option that does not meet mandatory criteria should be disregarded. For complex decisions, criteria can be grouped into categories or objectives.

6.8.1.2.2 Identify Alternative Solutions to Address Decision Issues

With the decision need well understood, alternatives can be identified that fit the mission and system context. There may be several alternatives that could potentially satisfy the decision criteria. Alternatives can be found from design options, operational options, cost options, and/or schedule options.

Almost every decision will have options to choose from. These options are often fairly clear within the mission and system context once the decision need is understood. In cases where the approach has uncertainty, there are several methods to help generate various options. Brainstorming decision options with those knowledgeable of the context and decision can provide a good list of candidate alternatives. A literature search of related systems and approaches to identify options may also provide some possible options. All possible options should be considered. This can get unwieldy if a large number of variations is possible. A "trade tree" (discussed later) is an excellent way to prune the set of variations before extensive analysis is undertaken, and to convey to other stakeholders the basis for that pruning.

A good understanding of decision need and criteria will include the definition of primary and secondary factors. Options should be focused on primary factors in the decision as defined by the

decision criteria. Non-primary factors (i.e., secondary, tertiary) can be included in evaluations but should not, in general, define separate alternatives. This will require some engineering judgment that is based on the mission and system context as well as the identified decision criteria. Some options may quickly drop out of consideration as the analysis is conducted. It is important to document the fact that these options were considered. A few decisions might only have one option. It is a best practice to document a decision matrix for a major decision even if only one alternative is determined to be viable. (Sometimes doing nothing or not making a decision is an option.)

6.8.1.2.3 Select Evaluation Methods and Tools

Based on the decision to be made, various approaches can be taken to evaluate identified alternatives. These can range from simple discussion meetings with contributing and affected stakeholders to more formal evaluation methods. In selecting the approach, the mission and system context should be kept in mind and the complexity of the decision analysis should fit the complexity of the mission, system, and corresponding decision.

Evaluation methods and tools/techniques to be used should be selected based on the purpose for analyzing a decision and on the availability of the information used to support the method and/or tool. Typical evaluation methods include: simulations; weighted tradeoff matrices; engineering, manufacturing, cost, and technical opportunity trade studies; surveys; human-in-the-loop testing; extrapolations based on field experience and prototypes; user review and comment; and testing. Section 6.8.2 provides several options.

6.8.1.2.4 Evaluate Alternative Solutions with the Established Criteria and Selected Methods

The performance of each alternative with respect to each chosen performance measure is evaluated. In all but the simplest cases, some consideration of uncertainty is warranted. Uncertainty matters in a particular analysis only if there is a non-zero probability that uncertainty reduction could alter the ranking of alternatives. If this condition is obtained, then it is necessary to consider the value of reducing that uncertainty, and act accordingly.

Regardless of the methods or tools used, results should include the following:

- Evaluation of assumptions related to evaluation criteria and of the evidence that supports the assumptions; and
- Evaluation of whether uncertainty in the values for alternative solutions affects the evaluation.

When decision criteria have different measurement bases (e.g., numbers, money, weight, dates), normalization can be used to establish a common base for mathematical operations. The process of "normalization" is making a scale so that all different kinds of criteria can be compared or added together. This can be done informally (e.g., low, medium, high), on a scale (e.g., 1-3-9), or more formally with a tool. No matter how normalization is done, the most important thing to remember is to have operational definitions of the scale. An operational definition is a repeatable, measurable number. For example, "high" could mean "a probability of 67 percent and above." "Low" could mean "a probability of 33 percent and below." For complex decisions,

decision tools usually provide an automated way to normalize. It is important to question and understand the operational definitions for the weights and scales of the tool.

> Note: Completing the decision matrix can be thought of as a default evaluation method. Completing the decision matrix is iterative. Each cell for each criterion and each option needs to be completed by the team. Use evaluation methods as needed to complete the entire decision matrix.

6.8.1.2.5 Select Recommended Solutions from the Alternatives Based on the Evaluation Criteria and Report to the Decision-Maker

Once the decision alternative evaluation is completed, recommendations should be brought back to the decision maker including an assessment of the robustness of the ranking (i.e., whether the uncertainties are such that reducing them could credibly change the ranking of the alternatives). Generally, a single alternative should be recommended. However, if the alternatives do not significantly differ, or if uncertainty reduction could credibly alter the ranking, the recommendation should include all closely ranked alternatives for a final selection by the decision-maker. In any case, the decision-maker is always free to select any alternative or ask for additional alternatives to be assessed (often with updated guidance on selection criteria). This step includes documenting the information, including assumptions and limitations of the evaluation methods used, and analysis of the uncertainty in the analysis of the alternatives' performance that justifies the recommendations made and gives the impacts of taking the recommended course of action, including whether further uncertainty reduction would be justifiable.

The highest score (e.g., percentage, total score) is typically the option that is recommended to management. If a different option is recommended, an explanation should be provided as to why the lower score is preferred. Usually, if an alternative having a lower score is recommended, the "risks" or "disadvantages" were too great for the highest ranking alternative indicating the scoring methods did not properly rank the alternatives. Sometimes the benefits and advantages of a lower or close score outweigh the highest score. If this occurs, the decision criteria should be reevaluated, not only the weights, but the basic definitions of what is being measured for each alternative. The criteria should be updated, with concurrence from the decision-maker, to more correctly reflect the suitability of each alternative.

6.8.1.2.6 Report Analysis Results

These results are reported to the appropriate stakeholders with recommendations, impacts, and corrective actions

6.8.1.2.7 Capture Work Products

These work products may include the decision analysis guidelines, strategy, and procedures that were used; analysis/evaluation approach; criteria, methods, and tools used; analysis/evaluation assumptions made in arriving at recommendations; uncertainties; sensitivities of the recommended actions or corrective actions; and lessons learned.

6.8.1.3 Outputs

6.8.1.3.1 Alternative Selection and Decision Support Recommendations and Impacts

Once the technical team recommends an alternative to a NASA decision-maker (e.g., a NASA board, forum, or panel), all decision analysis information should be documented. The team should produce a report to document all major recommendations to serve as a backup to any presentation materials used. A report in conjunction with a decision matrix provides clearly documented rationale for the presentation materials (especially for complex decisions). Decisions are typically captured in meeting minutes and should be captured in the report as well. Based on the mission and system context and the decision made, the report may be a simple white paper or a more formally formatted document. The important characteristic of the report is the content, which fully documents the decision needed, assessments done, recommendations, and decision finally made.

This report includes the following:

- Mission and system context for the decision
- Decision needed and intended outcomes
- Decision criteria
- Identified alternative solutions
- Decision evaluation methods and tools employed
- Assumptions, uncertainties, and sensitivities in the evaluations and recommendations
- Results of all alternative evaluations
- Alternative recommendations
- Final decision made with rationale
- Lessons learned

Typical information captured in a decision report is shown in Table 6.8-1.

Table 6.8-1 Typical Information to Capture in a Decision Report

#	Section	Section Description
1	Executive Summary	Provide a short half-page executive summary of the report: • Recommendation (short summary—1 sentence) • Problem/issue requiring a decision (short summary—1 sentence)
2	Problem/Issue Description	Describe the problem/issue that requires a decision. Provide background, history, the decision maker(s) (e.g., board, panel, forum, council), and decision recommendation team, etc.

#	Section	Section Description
3	Decision Matrix Setup Rationale	Provide the rationale for setting up the decision matrix: • Criteria selected • Options selected • Weights selected • Evaluation methods selected Provide a copy of the setup decision matrix.
4	Decision Matrix Scoring Rationale	Provide the rationale for the scoring of the decision matrix. Provide the results of populating the scores of the matrix using the evaluation methods selected.
5	Final Decision Matrix	Cut and paste the final spreadsheet into the document. Also include any important snapshots of the decision matrix.
6	Risk/Benefits	For the final options being considered, document the risks and benefits of each option.
7	Recommendation and/or Final Decision	Describe the recommendation that is being made to the decision maker(s) and the rationale for why the option was selected. Can also document the final decision in this section.
8	Dissent	If applicable, document any dissent with the recommendation. Document how dissent was addressed (e.g., decision matrix, risk).
9	References	Provide any references.
A	Appendices	Provide the results of the literature search, including lessons learned, previous related decisions, and previous related dissent. Also document any detailed data analysis and risk analysis used for the decision. Can also document any decision metrics.

6.8.2 Decision Analysis Guidance

Several different approaches to support decision analyses in collecting and assessing data for complex decisions are described in the following subsections. These processes can be considered in two categories: analysis methods supporting all systems engineering processes and phases, and specific methods supporting formal decision analysis.

Table 6.8-2 provides a list of the methods contained in the following subsections.

Table 6.8-2 Decision Analysis Methods

Type	Method
Analysis methods supporting all SE processes and phases	Systems Analysis, Simulation and Performance
	Trade Studies
	Cost-Benefit Analysis
Specific methods supporting formal decision analysis	Influence Diagrams
	Decision Trees
	Analytic Hierarchy Process
	Borda Counting
	Utility Analysis

6.8.2.1 Analysis Methods Supporting all Systems Engineering Processes and Phases

Systems analysis, trade studies, and cost-benefit analysis are examples of activities typically conducted during system design processes throughout the product life cycle. These may also be used as specific decision analysis techniques during any life-cycle phase of a program, project, or activity.

6.8.2.1.1 Systems Analysis, Simulation, and Performance

Systems analysis within the context of the life cycle is responsive to the needs of the stakeholder at every phase of the life cycle, from Pre-Phase A through Phase B to realizing the final product and into Operations and Sustainment, Phase E. Figure 6.8-3 is a notional view of the systems analysis process through the life cycle. It depicts hypothetical life-cycle events (i.e., concept collaboration, assessment and feedback to design) including corresponding lists of potential activities that may be performed under each event within the Decision Analysis Process.

Systems analysis of a product should support the transformation from a need into a realized, definitive product; be able to support compatibility with all physical and functional requirements; and support the operational scenarios in terms of reliability, maintainability, supportability, serviceability, and disposability, while maintaining performance and affordability.

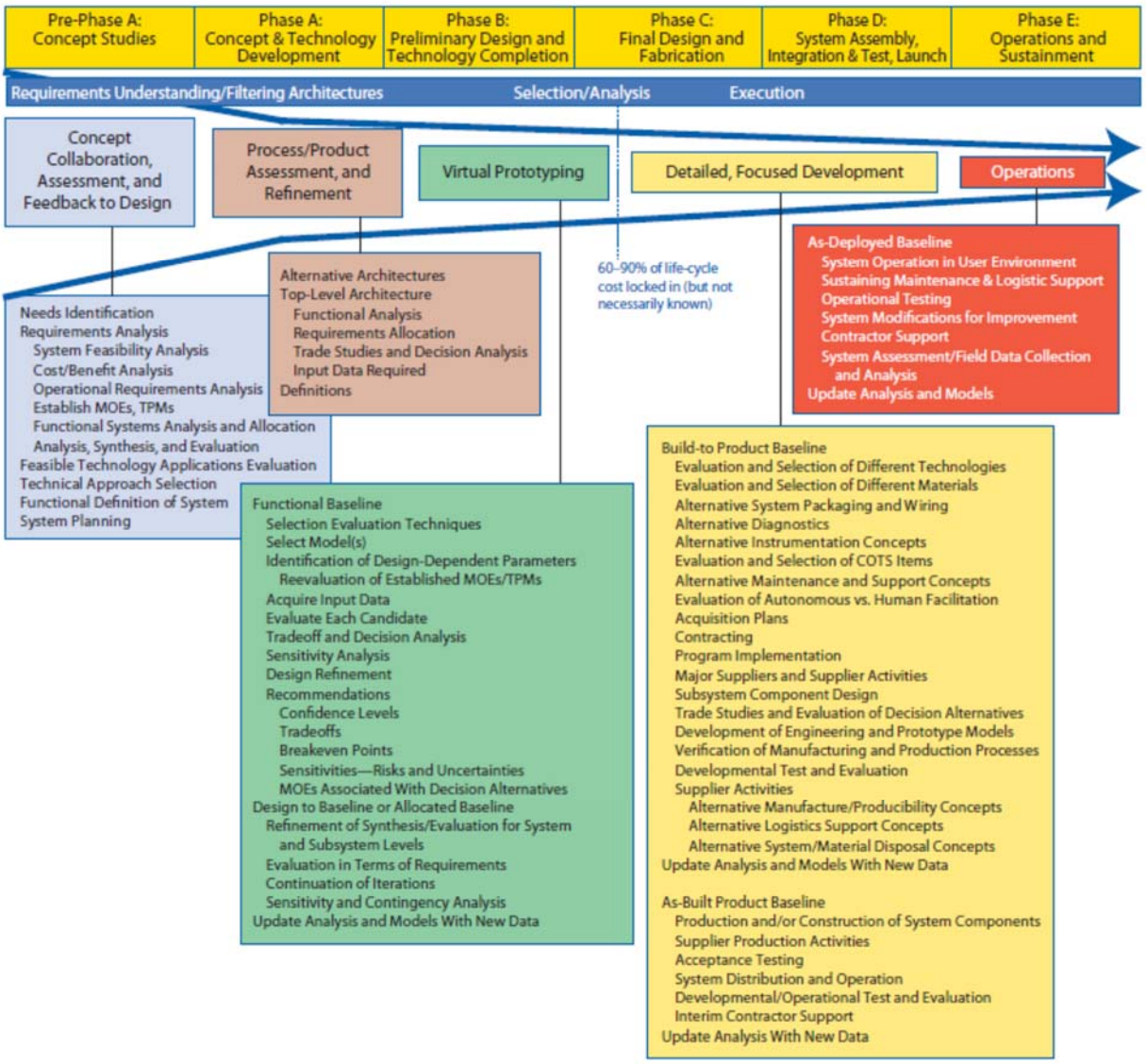


Figure 6.8-3 Example of Systems Analysis across the Life Cycle

Systems analysis support is provided from cradle to grave of the system. This covers the product design, verification, manufacturing, operations and support, and disposal. Viewed in this manner, life-cycle engineering is the basis for concurrent engineering.

Systems analysis should support concurrent engineering. Appropriate systems analysis can be conducted early in the life cycle to support planning and development. The intent here is to support seamless systems analysis optimally planned across the entire life cycle. For example, systems engineering early in the life cycle can support optimal performance of the deployment, operations, and disposal facets of the system.

Historically, this has not been the case. Systems analysis would focus only on the life-cycle phase that the project occupied at that time. The systems analyses for the later phases were treated serially, in chronological order. This resulted in major design modifications that were very costly in the later life-cycle phases. Resources can be used more efficiently if the

requirements across the life cycle are considered concurrently, providing results for decision-making about the system. See Section 2.6 and 7.9 on human systems integration as an example of the crosscutting approach that provides structure to this type of life-cycle analysis.

Figure 6.8-3 shows a life-cycle chart that indicates how the various general types of systems analyses fit across the phases of the life cycle. The requirements for analysis begin with a broader scope and more types of analysis required in the early phases of the life cycle and funnel or narrow in scope and analysis requirements as decisions are made and project requirements become clearer as the project proceeds through its life cycle. Figure 6.8-4 presents a specific spaceport example and shows how specific operational analysis inputs can provide analysis result outputs pertinent to the operations portion of the life cycle. Note that these simulations are conducted across the life cycle and updated periodically with the new data that is obtained as the project evolves.

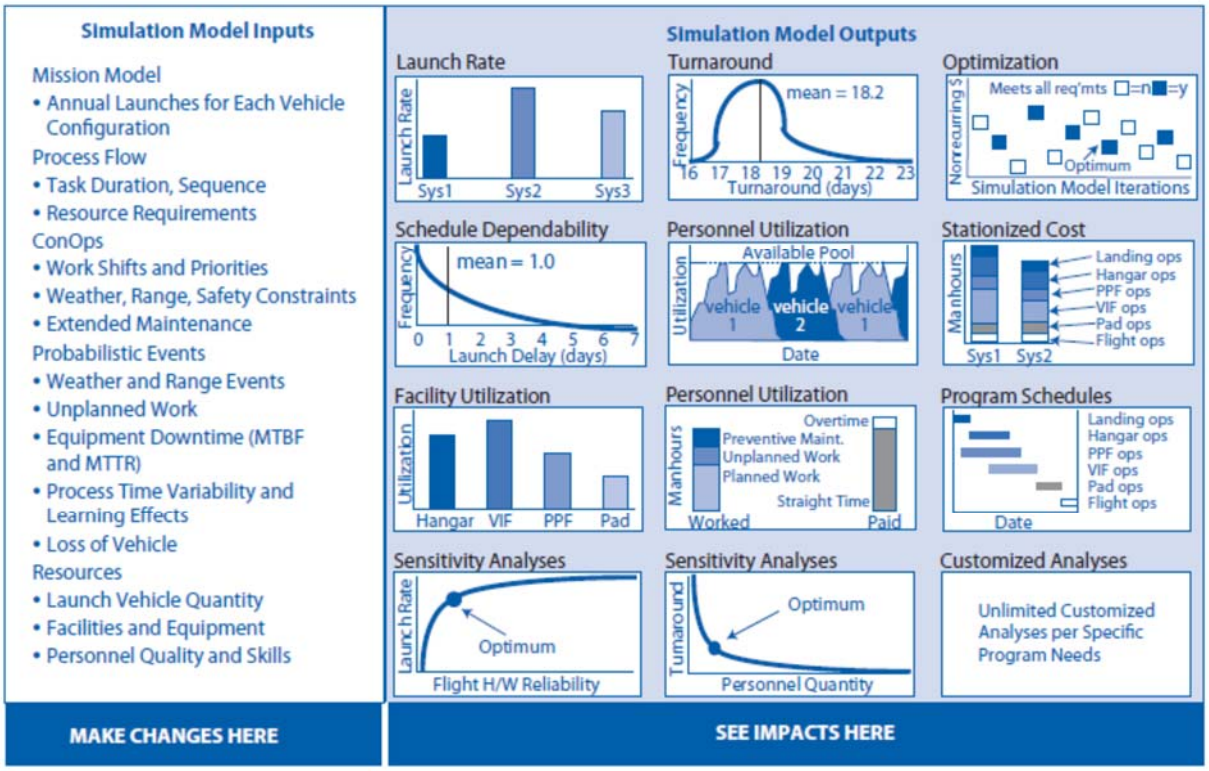


Figure 6.8-4 Simulation Model Analysis Techniques

From: Lockheed Martin presentation to KSC, November2003, Kevin Brughelli, Lockheed Martin Space Systems Company; Debbie Carstens, Florida Institute of Technology; and Tim Barth, KSC.

During the early life-cycle phases, inputs should include a plan for collecting the quantitative and qualitative data necessary to manage contracts and improve processes and products as the project evolves. This plan should indicate the type of data necessary to determine the cause of problems, nonconformances, and anomalies and propose corrective action to prevent recurrence. This closed-loop plan involving identification, resolution, and recurrence control systems is critical to producing actual reliability that approaches predicted reliability. It should indicate the information technology infrastructure and database capabilities to provide data sorting, data

mining, data analysis, and precursor management. Management of problems, instances of nonconformance, and anomalies should begin with data collection, should be a major part of technical assessment, and should provide critical information for decision analysis.

6.8.2.1.2 Trade Studies

The purpose of trade studies is to assist the decision-maker in selecting the most suitable option - from a set of several viable options - to achieve the goal and objectives within the constraints of the program, project, system, subsystem, or activity.

In performing trade study functional allocation, it is particularly important to ensure that the functions of all human roles and responsibilities in the system's behavior and success are included and allocated. See Section 2.6 and 7.9 on human systems integration.

Trade studies are essential in achieving product success through systems engineering. Trade studies help to define the emerging system at each level of resolution. Effective trade studies require the participation of people with many skills and a unity of effort to move toward an optimum system design. Figure 6.8-5 shows the trade study process in simplest terms.

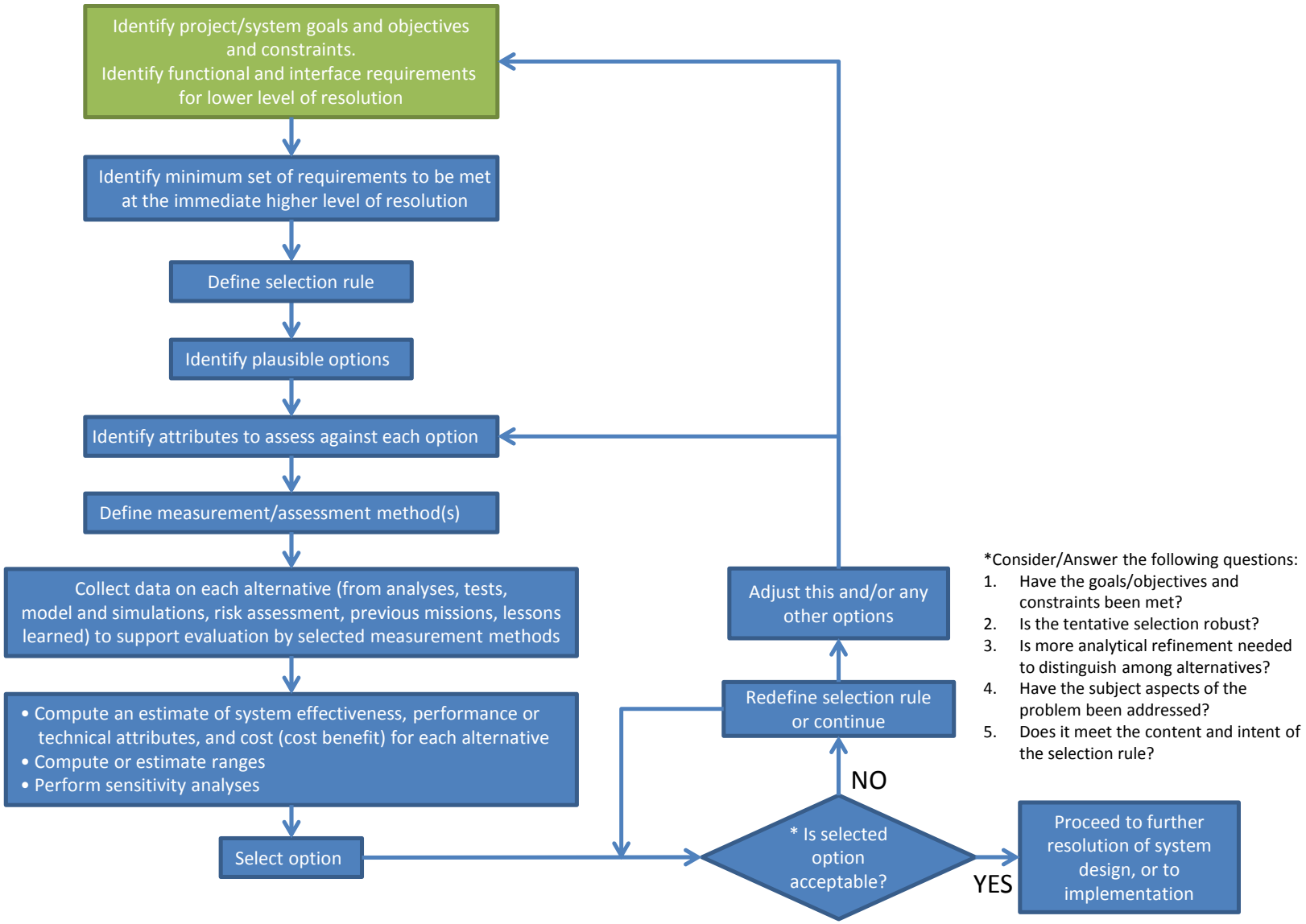


Figure 6.8-5 Trade Study Process

A program, project, system, or activity usually defines goals and objectives at the beginning of its life cycle including specific constraints to be met. These characteristics should be well understood by the systems engineer and should be taken into consideration at any step during the product life cycle. These characteristics may be affected by the results of the trade studies in such a way that they may be considered for modification. Therefore, the trade study report should include rationale justifying the proposed modification of goals, objectives, and constraints and the implications for cost, schedule, and technical performance as well.

In the same way, at each level of system resolution there exist goals, objectives, and constraints. The technical trade study team should acquire knowledge and understanding of these characteristics at the system level of resolution into which the results of the trade study will be directly implemented. This is the immediate higher system level of resolution at the starting point of the trade study. This step may be accomplished by performing a functional analysis. "Functional analysis" is the process of identifying, describing, and relating the functions a system should perform to fulfill its goals and objectives and is described in detail in Section 4.4.

Closely related to defining the goals and objectives and performing a functional analysis is the step of defining the measures and measurement methods for system effectiveness (when this is practical), system performance or technical attributes, and system cost. (These variables are collectively called outcome variables in keeping with the discussion in Section 2.3. Some systems engineering books refer to these variables as decision criteria, but this term should not be confused with "selection rule," described below. Sections 2.5 and 6.1 discuss the concepts of system cost and effectiveness in greater detail.) Defining measures and measurement methods begins the analytical portion of the trade study process since it suggests the involvement of those familiar with quantitative methods.

For each measure, it is important to address how that quantitative measure will be computed; that is, which measurement method is to be used. This step explicitly identifies those variables that are important in meeting the system's goals and objectives.

Evaluating the likely outcomes of various alternatives in terms of system effectiveness, the underlying performance or technical attributes, and cost before actual fabrication and/or programming usually requires the use of a mathematical model or series of models of the system. So a second reason for specifying the measurement methods is to identify necessary models.

Sometimes these models are already available from previous projects of a similar nature; other times, they need to be developed. In the latter case, defining the measurement methods should trigger the necessary system modeling activities. Since the development of new models can take a considerable amount of time and effort, early identification is needed to ensure they will be ready for formal use in trade studies.

For each option, testing (software, simulation, hardware-in-the-loop, subscale, prototype, etc.) in the specific defined configuration is advisable. Testing should be used to verify system / subsystem attributes' analytical performance to show system/subsystem flexibility and robustness or to complement analytical data.

Defining the selection rule is the step of explicitly determining how the outcome variables will be used to make a (tentative) selection of the preferred alternative. As an example, a selection rule may be to choose the alternative with the highest estimated system effectiveness that costs less than x dollars (with some given probability), meets safety requirements, and possibly meets other political or schedule constraints. Defining the selection rule is essentially deciding how the selection is to be made. This step is independent from the actual measurement of system effectiveness, system performance or technical attributes, and system cost.

Many different selection rules are possible. The selection rule in a particular trade study may depend on the context in which the trade study is being conducted—in particular, what level of system design resolution is being addressed. At each level of the system design, the selection rule generally should be chosen only after some guidance from the next higher level. The selection rule for trade studies at lower levels of the system design should be in consonance with the higher level selection rule.

Defining plausible alternatives is the step of creating some alternatives that can potentially achieve the goals and objectives of the system. This step depends on understanding (to an appropriately detailed level) the system's functional requirements and operational concept. Running an alternative through an operational timeline or reference mission is a useful way of determining whether it can plausibly fulfill these requirements. (Sometimes it is necessary to create separate behavioral models to determine how the system reacts when a certain stimulus or control is applied, or a certain environment is encountered. This provides insights into whether it can plausibly fulfill time-critical and safety requirements.) Defining plausible alternatives also requires an understanding of the technologies available, or potentially available, at the time the system is needed. Each plausible alternative should be documented qualitatively in a description sheet.

One way to represent the trade study alternatives under consideration is by a trade tree. A detailed and well-developed WBS, a system/subsystem functional tree, or even a fault tree may be used as bases for the trade tree. At each level of the tree and for each applicable function or component or mechanism proposed, trades are identified. When numerous trades are proposed, it is advisable to prioritize these trades based on the program/project/system/activity risk, relevance, and constraints.

During Phase A trade studies, the trade tree should contain a number of alternative high-level system architectures to avoid a premature focus on a single one. As the systems engineering process proceeds, branches of the trade tree containing unattractive alternatives will be "pruned," and greater detail in terms of system design will be added to those branches that merit further attention.

Given a set of plausible alternatives, the next step is to collect data on each to support the evaluation of the measures by the selected measurement methods. If models are to be used to calculate some of these measures, then obtaining the model inputs provides some impetus and direction to the data collection activity. By providing data, engineers in such disciplines as reliability, maintainability, producibility, integrated logistics, software, testing, operations, and costing have an important supporting role in trade studies. The data collection activity, however,

should be orchestrated by the systems engineer. The results of this step should be a quantitative description of each alternative. Test results on each alternative can be especially useful.

Early in the systems engineering process, performance and technical attributes are generally uncertain and should be estimated or conventionally defined. Data from breadboard and brassboard testbeds can provide additional confidence that the range of values used as model inputs is correct. Such confidence is also enhanced by drawing on data collected on related, previously developed systems. As the life cycle advances forward, testing becomes not only encouraged but required in some instances. Testing conditions (parameters; environment; duration; etc.) should be equally representative for each option and data should be collected with the same resources (equipment; personnel; etc.) within the same conditions for all options as well. When deviations occur, rationale supported with analysis or any other strong technical justification should be included in the assessment, and the endorsement by the trade study technical team should be documented and included in the trade study report.

Technical risk management for each option should be performed as discussed in Section 6.4, starting with an identification of potential system/subsystem risks and associated consequences and the likelihood of occurrence. A proposed risk mitigation should be developed (proposed) by the technical team and discussed in the report identifying potential effects in cost and schedule not only for the product level of resolution in question but also for the whole program / project / system / activity, if applicable.

The next step in the trade study process is to quantify the outcome variables by computing estimates of system effectiveness, its underlying system performance or technical attributes, and system cost. These variables should be calculated for each option independently having similar environmental operational and extreme conditions for each option. Rationale for any deviation should be endorsed by the technical team and documented for inclusion in the trade study report. If the needed data have been collected and the measurement methods (for example, models) are in place, then this step is, in theory, mechanical. In practice, considerable skill is often needed to get meaningful results.

In an ideal world, all input values would be precisely known and models would perfectly predict outcome variables. This not being the case, the systems engineer should supplement point estimates of the outcome variables for each alternative with computed or estimated uncertainty ranges. For each uncertain key input, a range of values should be estimated. Using this range of input values, the sensitivity of the outcome variables can be gauged and their uncertainty ranges calculated. The systems engineer may be able to obtain meaningful probability distributions for the outcome variables using Monte Carlo simulation, but when this is not feasible, the systems engineer should be content with only ranges and sensitivities.

Combining the selection rule with the results of the analytical activity should enable the systems engineer to array the alternatives from most preferred to least, in essence making a tentative selection. This tentative selection should not be accepted blindly. In most trade studies, there is a need to subject the results to a "reality check" by considering a number of questions. Have the goals, objectives, and constraints truly been met? Is the tentative selection heavily dependent on a particular set of input values to the measurement methods, or does it hold up under a range of reasonable input values? (In the latter case, the tentative selection is said to be "robust.") Are

there sufficient data to back up the tentative selection? Are the measurement methods sufficiently discriminating to be sure that the tentative selection is really better than other alternatives? Have the subjective aspects of the problem been fully addressed?

If the answers support the tentative selection, then the systems engineer can have greater confidence in a recommendation to proceed to a further resolution of the system design, or to the implementation of that design. The estimates of system effectiveness, its underlying performance or technical attributes, and system cost generated during the trade study process serve as inputs to that further resolution. The analytical portion of the trade study process often provides the means to quantify the performance or technical (and cost) attributes that the system's lower levels should meet. These can be formalized as performance requirements.

If the reality check is not met, the systems engineer or the decision-maker may consider modifying the definition rule based on technical observations during the data collection and processing for each option, and then reconsider how the preselected option meets the redefined rule. When the redefined rule is not properly met by the preselected option, the trade study may continue by considering adjusting options or bringing additional options to the study. Then, the trade study process returns to one or more earlier steps. This iteration may result in a change in the goals, objectives, and constraints; a new alternative; or a change in the selection rule based on the new information generated during the trade study. The reality check may lead instead to a decision to first improve the measures and measurement methods (e.g., models) used in evaluating the alternatives, and then to repeat the analytical portion of the trade study process.

Controlling the Trade Study Process

A technical trade study team is formed and is composed of representatives from those disciplines in engineering and science that need to be involved in the design, function, and operation of the system/subsystem that may be affected by the trade study. The decision-maker should select an experienced and technically qualified team leader who is capable of guiding and managing the study within a flexible and participative teamwork style of inclusion but under rigorous engineering and science principles, facts, and findings. The team should concentrate on the assigned trade study from the conception until its completion to assure continuity and traceability of concepts, data, and decisions made as the study progresses. A general and outline plan relative to the team activities should be developed and embraced by the whole technical team.

Individual tasks may be assigned to individuals or subteams as long as they report back to the entire technical team in open discussion. Each plausible option should be researched and studied under the same program/project/system/activity goals and objectives and constraints, and the analyses, testing, modeling, and simulation should be conducted under similar if not identical parameters and conditions for all options. The team leader should provide periodic briefings to the decision-maker regarding the technical team progress. Data should be collected in an orderly manner and should be available to each team member at any time throughout the trade study. The final report could be written by a group of selected members of the team as long as it has endorsement from all team members. Dissenting opinions that could not be sorted out within the technical team body should be brought forward to the attention of the decision-maker for consideration and disposition.

Trade Study Reports

Trade study reports should be prepared for each trade study. An executive summary of the report should be prepared containing the purpose of the study, the options considered and the results including option selected and recommendations. The members of the technical team and its leader should sign the report. At a minimum, each trade study report should identify the following:

- The system under analysis.
- System goals and constraints.
- Goals, objectives and constraints of the immediate higher system level of resolution.
- The measures and measurement methods (models) used.
- All data sources used.
- The alternatives chosen for analysis.
- The computational results, including uncertainty ranges and sensitivity analyses performed.
- Highest level of testing performed for each option (configuration and conditions) – results and assessment.
- Risk assessment (risk analysis; proposed risk mitigation; risk/benefit) for each option – results and recommendations.
- The selection rule used – reasoning when more than one was needed.
- The recommended alternative – rationale.
- Unresolved dissenting positions – documentation.
- Appendices.

Additional guidelines for report content are included in Table 6.8-1.

Trade study reports should be maintained as part of the system archives to ensure traceability of decisions made through the systems engineering process. Using a generally consistent format for these reports also makes it easier to review and assimilate them into the formal change control process.

6.8.2.1.3 Cost-Benefit Analysis

A cost-benefit analysis is performed to determine the advantage of one alternative over another in terms of equivalent cost or benefits. The analysis relies on the addition of positive factors and the subtraction of negative factors to determine a net result. Cost-benefit analysis maximizes net benefits (benefits minus costs). A cost-benefit analysis finds, quantifies, and adds all the positive factors. These are the benefits. Then it identifies, quantifies, and subtracts all the negatives, the costs. The difference between the two indicates whether the planned action is a preferred alternative. Doing a cost-benefit analysis well means making sure to include all the costs and all the benefits and properly quantify them. Note that too often the costs, benefits, and limitations of the human elements in the total system's operation and performance are not adequately assessed.

(See Section 2.6 and 7.9 on human systems integration.) A similar approach, which is used when a cost cap is imposed externally, is to maximize effectiveness for a given level of cost. Cost-effectiveness is a systematic quantitative method for comparing the costs of alternative means of achieving the same equivalent benefit for a specific objective. A project is cost-effective if, on the basis of the life-cycle cost analysis of competing alternatives, it is determined to have the lowest costs expressed in present value terms for a given level of benefits.

Least-cost analysis aims at identifying the least-cost project option for meeting the technical requirements. Least-cost analysis involves comparing the costs of the various technically feasible options and selecting the one with the lowest costs. Project options should be alternative ways of achieving the mission objectives. If differences in results or quality exist, a normalization procedure should be applied that takes the benefits of one option relative to another as a cost to the option. This provides a cost penalty for an option that does not meet all of the mission objectives to ensure an equitable comparison. Procedures for the calculation and interpretation of the discounting factors should be made explicit, with the least-cost project being identified by comparing the total life-cycle costs of the project alternatives and calculating the equalizing factors for the difference in costs. The project with the highest equalizing factors for all comparisons is the least-cost alternative.

Cost-effectiveness analysis also deals with alternative means of achieving mission requirements. However, the results may be estimated only indirectly. For example, different types of systems may be under consideration to obtain science data. The effectiveness of each alternative may be measured through obtaining science data through different methods. An example of a cost-effectiveness analysis requires the increase in science data to be divided by the costs for each alternative. The most cost-effective method is the one that raises science data by a given amount for the least cost. If this method is chosen and applied to all similar alternatives, the same increase in science data can be obtained for the lowest cost. Note, however, that the most cost-effective method is not necessarily the most effective method of meeting mission objectives. Another method may be the most effective, but also cost a lot more, so it is not the most cost-effective. The cost-effectiveness ratios—the cost per unit increase in science data for each method—can be compared to see how much more it would cost to implement the most effective method. Which method is chosen for implementation then depends jointly on the desired mission objectives and the extra cost involved in implementing the most effective method.

There will be circumstances where project alternatives have more than one outcome. To assess the cost-effectiveness of the different alternatives, it is necessary to devise a testing system where the results for the different factors can be added together. It also is necessary to decide on weights for adding the different elements together, reflecting their importance in relation to the objectives of the project. Such a cost-effectiveness analysis is called weighted cost-effectiveness analysis. It introduces a subjective element, the weights, into the comparison of project alternatives, both to find the most cost-effective alternative and to identify the extra cost of implementing the most effective alternative.

6.8.2.2 Specific Methods Supporting Formal Decision Analysis

There are several methods that formal decision analysis can use to help define solutions for complex, diverse, multivariate, and diverse participant problems. Influence diagrams, decision

trees, Analytic Hierarchy Process (AHP), borda counting, utility analysis, and risk-informed decision-making are briefly described herein.

6.8.2.2.1 Influence Diagrams

An influence diagram (also called a decision network) is a compact graphical and mathematical representation of a decision state. (See Figure 6.8-6.) Influence diagrams were first developed in the mid-1970s within the decision analysis community as an intuitive approach that was easy to understand. They are now adopted widely and are becoming an alternative to decision trees, which typically suffer from exponential growth in number of branches with each variable modeled. An influence diagram is directly applicable in team decision analysis since it allows incomplete sharing of information among team members to be modeled and solved explicitly. Its elements are:

- Decision nodes, indicating the decision inputs, and the items directly influenced by the decision outcome;
- Chance nodes, indicating factors that impact the chance outcome, and items influenced by the chance outcome;
- Value nodes, indicating factors that affect the value, and items influenced by the value; and
- Arrows, indicating the relationships among the elements.

An influence diagram does not depict a strictly sequential process. Rather, it illustrates the decision process at a particular point, showing all of the elements important to the decision. The influence diagram for a particular model is not unique. The strength of influence diagrams is their ability to display the structure of a decision problem in a clear, compact form, useful both for communication and to help the analyst think clearly during problem formulation. An influence diagram can be transformed into a decision tree for quantification.

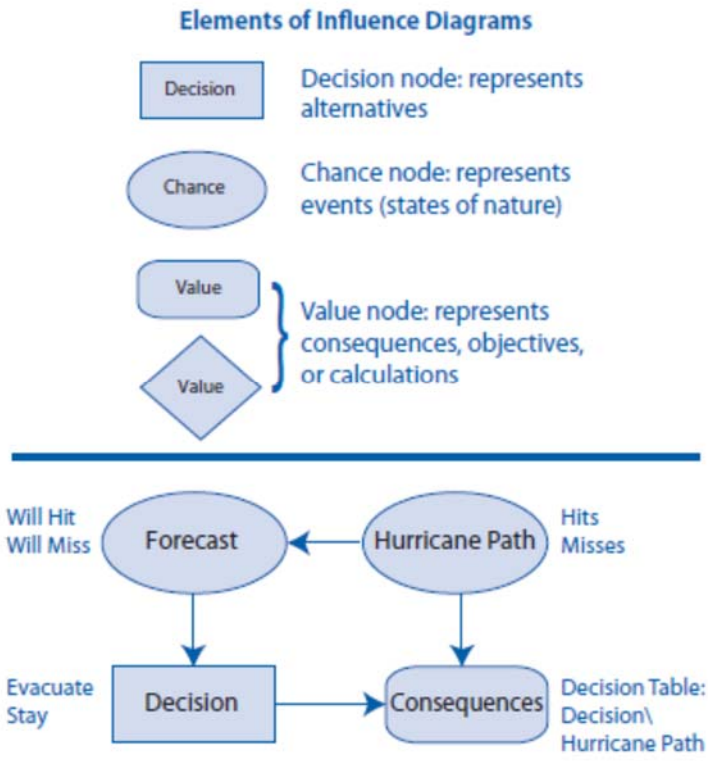


Figure 6.8-6 Influence Diagrams

6.8.2.2.2 Decision Trees

Like the influence diagram, a decision tree portrays a decision model, but a decision tree is drawn from a point of view different from that of the influence diagram. The decision tree exhaustively works out the expected consequences of all decision alternatives by discretizing all "chance" nodes, and, based on this discretization, calculating and appropriately weighting all possible consequences of all alternatives. The preferred alternative is then identified by summing the appropriate outcome variables (MOEs or expected utility) from the path end states.

A decision tree grows horizontally from left to right, with the trunk at the left. Typically, the possible alternatives initially available to the decision-maker stem from the trunk at the left. Moving across the tree, the decision-maker encounters branch points corresponding to probabilistic outcomes and perhaps additional decision nodes. Thus, the tree branches as it is read from left to right. At the far right side of the decision tree, a vector of TPM scores is listed for each terminal branch, representing each combination of decision outcome and chance outcome. From the TPM scores, and the chosen selection rule, a preferred alternative is determined. In even moderately complicated problems, decision trees can quickly become difficult to understand. Figure 6.8-7 shows a sample of a decision tree. This figure only shows a simplified illustration. A complete decision tree with additional branches would be expanded to the appropriate level of detail as required by the analysis. A commonly employed strategy is to start with an equivalent influence diagram. This often aids in helping to understand the principal issues involved. Some software packages make it easy to develop an influence diagram and then, based on the influence diagram, automatically furnish a decision tree. The decision tree can be edited if this is desired. Calculations are typically based on the decision tree itself.

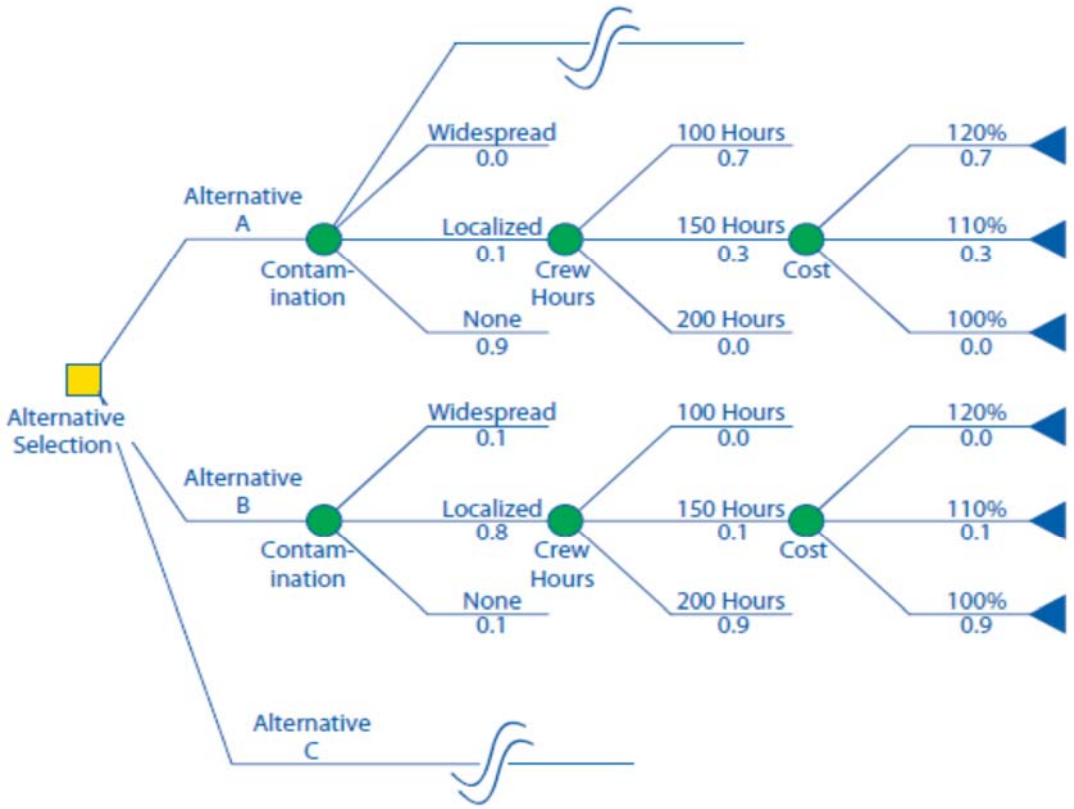


Figure 6.8‑7 Decision Tree

6.8.2.2.3 The Analytic Hierarchy Process

Multi-Criteria Decision Analysis (MCDA) is a method aimed at supporting decision-makers who are faced with making numerous and conflicting evaluations. These techniques attempt to highlight the conflicts in alternatives and derive a way to come to a compromise in a transparent process. For example, NASA may apply MCDA to help assess whether selection of one set of software tools for every NASA application is cost-effective. MCDA involves a certain element of subjectivity; the bias and position of the team implementing MCDA play a significant part in the accuracy and fairness of decisions. One of the MCDA methods is the Analytic Hierarchy Process (AHP).

AHP was first developed and applied by Thomas Saaty. AHP is a multi-attribute methodology that provides a proven, effective means to deal with complex decision-making and can assist with identifying and weighting selection criteria, analyzing the data collected for the criteria, and expediting the decision-making process. Many different problems can be investigated with the mathematical techniques of this approach. AHP helps capture both subjective and objective evaluation measures, providing a useful mechanism for checking the consistency of the evaluation measures and alternatives suggested by the team, and thus reducing bias in decision-making. AHP is supported by pair-wise comparison techniques, and it can support the entire decision process.

AHP does have some limitations and care should be used in its application. Use of AHP to determine criteria weighting is a good application of the methodology. AHP can bias results, however, in selecting actual alternatives. The pair-wise comparison approach assumes all other factors are independent in the comparison. This is often not true, and the collective influence of all alternatives in the decision criteria should be considered simultaneously in these cases.

The AHP normally has six steps:

1. Describe in summary form the alternatives under consideration.
2. Develop a set of high-level objectives.
3. Decompose the high-level objective from general to specific to produce an objectives hierarchy.
4. Determine the relative importance of the evaluation objectives and attributes by assigning weights arrived at by engaging experts through a structured process such as interviews or questionnaires.
5. Have each expert make pair-wise comparisons of the performance of each decision alternative with respect to a TPM. Repeat this for each TPM. Combine the results of these subjective evaluations mathematically using a process or, commonly, an available software tool that ranks the alternatives.
6. Iterate the interviews/questionnaires and AHP evaluation process until a consensus ranking of the alternatives is achieved.

If AHP is used only to produce the TPM weights to be used in a Performance Index (PI) or MOE calculation, then only the first four steps listed above are applicable.

With AHP, convergence may be achieved quickly or several feedback rounds may be required. The feedback consists of reporting the computed ranking, for each evaluator and for the group, for each option, along with the reasons for differences in rankings and identified areas of divergence. Experts may choose to change their judgments on TPM weights. At this point, divergent preferences can be targeted for more detailed study. AHP assumes the existence of an underlying preference vector with magnitudes and directions that are revealed through the pair-wise comparisons. This is a powerful assumption, which may at best hold only for the participating experts. The ranking of the alternatives is the result of the experts' judgments and is not necessarily a reproducible result. For further information on AHP, refer to Saaty, *The Analytic Hierarchy Process*.

6.8.2.2.4 Borda Counting

AHP will not always provide a clear comparison between multiple options because it is focused on pair-wise comparisons rather than the set as a whole. There are often alternatives that do well in all criteria but are not clearly better in any single criteria. Borda counting allows for an assessment looking for the alternative that best fits all decision criteria. In so doing, it allows for the intuition of subject matter experts (including the systems engineer) to be captured in the assessment.

Instead of comparing all possible options in pairs, borda counting allows for voting all options with a weighted ranking. For example, if trying to evaluate five decision options (A, B, C, D, E), the borda count would ask each individual assessor to score each option with a weighted score. (See Table 6.8-3.) The weighting reflects the preference such that a score of 4 would be first, 3 second, 2 third, 1 fourth, and 0 fifth. Summing the points of all votes provides a ranking of the options. It will often be found that an option that ranks second in most categories will outweigh other options that rank first in only a few categories.

Table 6.8-3 Borda Count

Category	A	B	C	D	E
Energy Efficiency	3	4	0	2	1
Mass	2	3	4	0	1
Operability	3	1	2	4	0
Safety	3	1	2	0	4
Cost	3	0	2	4	1
Schedule	2	3	0	1	4
Total	16	12	10	11	11

This allows for an integrated assessment of all criteria for all options without having a single criterion that biases the results. (If a single criterion does, then this indicates that the other criteria should be secondary factors as discussed in Section 6.8.1 on decision criteria identification.) The borda count provides for the application of the systems engineers' intuition when the evaluations are close and the subject matter experts differ on how specific criteria should be weighted.

6.8.2.2.5 Utility Analysis

"Utility" is a measure of the relative value gained from an alternative. Given this measure, the team looks at increasing or decreasing utility, and thereby explains alternative decisions in terms of attempts to increase their utility. The theoretical unit of measurement for utility is the util.

The utility function maps the range of the TPM into the range of associated utilities, capturing the decision-maker's preferences and risk attitude. It is possible to imagine simply mapping the indicated range of values linearly onto the interval [0,1] on the utility axis, but in general, this would not capture the decision-maker's preferences. The decision-maker's attitude toward risk causes the curve to be convex (risk prone), concave (risk averse), or even some of each. Therefore, a utility function, which can be a linear combination of nonlinear preference functions, is constructed looking at the decision-maker's preferences. Note that Multi-Attribute Utility Theory (MAUT) discussed below allows for preference interactions and nonlinear combinations of preference functions.

The utility function directly reflects the decision-maker's attitude toward risk. When ranking alternatives on the basis of utility, a risk-averse decision-maker will probably rank an alternative with greater benefit and highly uncertain performance below an alternative having lesser benefit but less uncertainty. The opposite outcome would result for a risk-prone decision-maker. When the individual TPM utility functions have been assessed, it is important to check the result for

consistency with the decision-maker's actual preferences; e.g., is it true that intermediate values of TPM_1 and TPM_2 are preferred to a high value of TPM_1 and a low value of TPM_2?

An example of a utility function for the TPM "volume" is shown in Figure 6.8-8. This measure was developed in the context of the design of sensors for a space mission. Volume was a precious commodity. The implication of the graph is that low volume is good, large volume is bad. Design alternatives with high uncertainty in their volume estimates offer potentially more compact designs but with a higher risk that additional volume will be needed to address unexpected design issues; e.g., the optical system cannot be folded as anticipated, thermal expansion requires more volume than expected, so radiation shielding may need to be thicker. The risk adverse decision-maker will choose options with less uncertainty even though they may have higher volume requirements, as long as the option meets the basic system limits. These can be shown by curve 1 in Figure 6.8-8. The risk taking decision-maker, on the other hand, will choose higher risk options, willing to risk the overall system gains in a more compact design even though the potential exists that the volume will expand. This is shown by curve 2 in Figure 6.8-8. The convexity (risk taking) or concavity (risk averse) of the curve is determined by comparing it to the slope of the reference line in Figure 6.8-8.

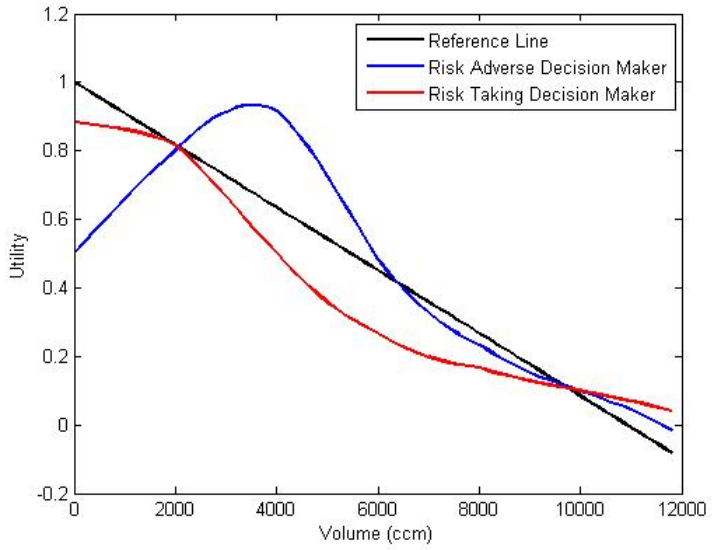


Figure 6.8-8 Utility Function for a "Volume" Performance Measure

Value Function

Value functions can take the place of utility functions when a formal treatment of risk attitude is unnecessary. They appear very similar to utility functions but have one important difference. Value functions do not consider the risk attitude of the decision-maker. They do not reflect how the decision-maker compares certain outcomes to uncertain outcomes.

The assessment of a TPM's value function is relatively straightforward. The "best" end of the TPM's range is assigned a value of 1. The "worst" is assigned a value of 0. The decision-maker makes direct assessments of the value of intermediate points to establish the preference structure

in the space of possible TPM values. The utility function can be treated as a value function, but the value function is not necessarily a utility function.

Multi-Attribute Utility Theory

One way to rank alternatives is to use a Multi-Attribute Utility Theory (MAUT) approach. With this approach, the "expected utility" of each alternative is quantified and alternatives are ranked based on their expected utilities.

Sometimes the expected utility is referred to as a Performance index (PI). An important benefit of applying this method is that it is the best way to deal with significant uncertainties when the decision-maker is not risk neutral. Probabilistic methods are used to treat uncertainties. A downside of applying this method is the need to quantify the decision-maker's risk attitudes. Top-level system architecture decisions are natural examples of appropriate applications of MAUT.