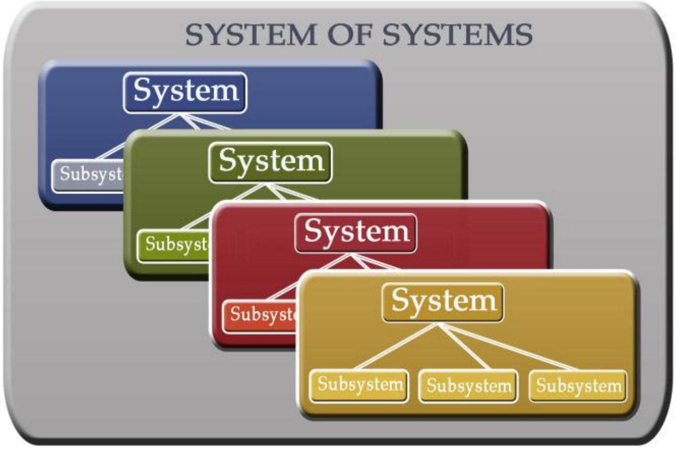
SMC
Systems Engineering
PRIMER & HANDBOOK

Concepts, Processes and Techniques



Space & Missile Systems Center
U.S. Air Force

Volume 1 4th Edition **11 March 2013**

SMC Systems Engineering

Concepts, Processes and Techniques

Approved for Public Release; Distribution is Unlimited

Table of Contents

List of Tables

List of Figures

Preface

This booklet was prepared for the United States Air Force Space and Missile Systems Center (SMC). It is intended as a primer to systems engineering (SE) that focuses on acquisition of large DoD systems, especially space. It is not all-inclusive and should be supplemented with Air Force and Department of Defense (DoD) directives, policies and procedures.

This SE handbook is written to provide SMC personnel with fundamental SE concepts and techniques as they apply to space, launch, and ground control systems and the SMC environment. The intended audience includes the project officer, junior systems engineer, an engineer in another discipline that must perform SE functions, or the experienced engineer who needs a suitable reference.

The authors recognize that SE subject matter is very broad and that approaches to performing SE vary greatly. This exposition is not intended to cover them all. It addresses general concepts and common processes, tools, and techniques that are mostly familiar to SMC. It also provides information on recommended SE practices and pitfalls to avoid. Many references are provided for the reader to consult for more in-depth knowledge.

This handbook describes SE as it could be applied to the development of major space, launch, and ground control systems. SE provides a disciplined approach that covers the entire lifecycle of a system to include development, design, manufacture, integration and test, and operation and sustainment. Consequently, the handbook's scope properly includes SE functions regardless of whether they are performed by the AFSPC operational user, SMC system program office (Program Office), or a systems contractor.

This book is also prepared to accommodate the SMC SE training program. It is written to accompany formal SMC SE training courses. The first chapter introduces the reader to system of systems and family of systems. Chapter 2 expands on SE concepts and terms and provides a more detailed explanation of the SE process. The end-to-end life cycle on a major space system is covered in Chapter 3. The first three chapters provide the basis for Chapter 4 – SE management. Chapter 5 introduces the reader to common SE tools and methods; Chapter 6 on specialty engineering integration, and Chapter 7 on validation and verification. The chapters are supplemented by appendices that include templates and examples to perform focused SE related tasks.

Many different sources were used to prepare this book including the latest DoD Instruction (DoDI), Air Force Instruction (AFI), previous SE handbooks developed for SMC, and a number of engineering publications that are cited throughout this book.

Finally, this text should be considered only a starting point. The SMC environment is undergoing rapid evolution.

As these initiatives bear fruit, this handbook is likely to be updated. Therefore, a Customer Review & Feedback Form is in Appendix E for your submission to Mr. Dave Davis at david.davis.3@us.af.mil, Nick Awwad at naim.awwad@us.af.mil, or Teresa Yeh at teresa.yeh@iseservices.com

Acknowledgments

This work was conducted under the overall direction of Mr. David Davis and Mr. Nick Awwad at SMC/EN. Our thanks and expressed appreciation is extended to the many individuals who contributed material, reviewed various drafts, or otherwise provided valuable input to this handbook and its predecessor versions. The list of major contributors is shown below.

Contributors (1st Edition)

Guy Hoshour
Barry Portner
Bill Crabtree
Brian Sullivan
Alan Adams
Rod Bennett
John Kern
Claude Kitchens
Dar Johnson
Bill Berrier
Jon Mathis
Kamran Shaik

Contributors (2nd Edition)

Barry Portner
Bill Crabtree
John Kern
Claude Kitchens
Kamran Shaik
Jonathan Gayek
Carl Billingsley
Reuven Dar
Theodore Krawczyk
Michael Zambrana
Albert Ramos

Contributors (3rd Edition)

Barbara Adams
Barry Portner
Carlos Echevarrio
Chuck O’Connor
Clint Roberson
Colonel Rakesh Dewan
Dan Martens
Dave Davis
Dave Lillie
Dr. Charles Tang
Dr. Edmund Conrow
Elizabeth Collins
Frank Knight
Harry Hamilton
Jaime Rico
Joseph Gennaro
Jon Mathis
Kamran Shaik
Norman Mendoza
Larry Pennell
Ray Johnson
Theodore Krawczyk
Tony Anthony
Trent Givens
Tracy Kaber
Wynn Rowell

Contributors (4th Edition)

Kamran Shaik
Teresa Yeh
Barry Portner
Bill Kaneshiro
Mario Moya
Ryan Mahelona
Jerry Verduft
Claude Kitchens
Aamil Shaik
Hana Kwon
Wynn Rowell

1 Systems Engineering Primer

1.1 System Definition

A system is a set of elements that interact with one another in an organized or interrelated fashion toward a common purpose that cannot be achieved by any of the elements alone or by all of the elements without the underlying organization. The personal computer (PC) shown in Figure 1-1 is symbolic of the manufacturing, test, maintenance, and other engineering functions that are also integral to the system. The system elements are organized or interrelated to achieve the purpose of the PC. The organization is facilitated by electrical cables and connectors and mechanical fasteners.

Figure 1-1 A personal computer system

The reader may have noted that each of the elements of the PC in turn satisfies the definition of a system. For example, the elements of the processor include the motherboard, the power supply, and the case; all organized to carry out the processing. The motherboard is further made up of parts and materials that have been assembled by way of processes such as soldering. Parts, materials, and processes are the building blocks of most manmade systems.

The purpose of military systems is to provide a needed new or improved operational capability to the warfighter. Some military systems are weapon systems applied in combat while others are operational support systems used for situational awareness, training, testing, or characterizing natural or threat environment in which the forces and equipment must operate. Highlight Box 1-1 shows a few authoritative definitions of a system.

1.2 System Elements

The elements of a system may be quite diverse, consisting of hardware, software, people, data, and facilities. The hardware or equipment and the installed software include operational elements to provide the needed capability, and manufacturing tools and test equipment to build and test the hardware. For military systems, the equipment usually also includes maintenance and support elements (i) to keep all elements working, (ii)

Highlight Box 1-1
What is a system?

ANSI/EIA-632-1999: "An aggregation of end products and enabling products to achieve a given purpose."

IEEE Standard 1220-1998: "A set or arrangement of elements and processes that are related and whose behavior satisfies customer/operational needs and provides for life cycle sustainment of the products."

ISO/IEC 15288:2008: "A combination of interacting elements organized to achieve one or more stated purposes."

NASA Systems Engineering Handbook, Rev. 1, 2007: "(1) The combination of elements that function together to produce the capability to meet a need. The elements include all hardware, software, equipment, facilities, personnel, processes, and procedures needed for this purpose. (2) The end product (which performs operational functions) and enabling products (which provide life-cycle support services to the operational end products) that make up a system."

INCOSE, Systems Engineering Handbook", v3.1, 2007: "homogeneous entity that exhibits predefined behavior in the real world and is composed of heterogeneous parts that do not individually exhibit that behavior and an integrated configuration of components and/or subsystems."

Joint Publication 1-02: “A functionally, physically, and/or behaviorally related group of regularly interacting or interdependent elements; that group of elements forming a unified whole.”

training elements to train people in the use of all the elements, and (iii) special tools and deployment elements to install, checkout and maintain elements in their operational location. For military systems, the people are usually specified in terms of manpower and skill levels. The data typically include the procedures to manufacture, verify, train, deploy, operate, and support/maintain the system and to responsibly dispose expendables or equipment no longer needed. Government facilities may include control centers, launch pads, test and training facilities, and connecting roadways and utilities such as power and water.

1.3 Space Systems

Space systems are characterized by the inclusion of earth-orbiting satellites or spacecraft as an integral part of the system, offering a unique capability or service. For example, Global Positioning System (GPS) provides highly accurate estimates of position, navigation, and time globally that would be hard to achieve otherwise. A space system typically consists of (i) a space segment, (ii) a launch segment, and (iii) ground and user segments.

1.3.1 Space Segment

The space segment consists of one or more Space Vehicles (SV) or satellites. Each satellite is made up of its elements, typically the payload (that provides the basic mission capability such as communications, surveillance, or navigation) and the spacecraft or bus that supports the payload by providing services like electrical power, thermal control, communications, and attitude control. The payload and bus are, of course, subdivided into lower tier elements such as processors, sensors, communications (radios), and clocks which are in turn made up of parts (such as integrated circuits, relays, or roller bearings) and materials (such as metallic or composite structures), all fabricated and assembled using various processes.

1.3.2 Launch Segment

The launch segment of a space system provides the lift capability to propel the SVs in earth-orbit or deep space. The launch system is typically made up of the launch vehicles like Atlas V and Delta IV that provide the initial boost toward orbit, upper or transfer orbit stages which place the satellite in or near its operational orbit, ground control and monitoring systems, and other facilities used for checking out, mating, and supporting the launch vehicles, upper stages, and satellites prior to launch. Each launch vehicle may be made up of multiple launch stages. Each launch stage and upper stage is typically made up of propulsion, guidance and control, and environmental protection elements.

The distinction between launch systems and satellite systems is not always clear such as the case of the Space Shuttle which was a launch system that could also perform or support operations on orbit, or the case of integral upper stages which are supplied as part of the satellite system to complete part or all of the transfer orbit function.

1.3.3 Ground and User Segments

The ground and user segments consists of space system assets on the ground and air used to control, service, and operate the SVs to help deliver situational awareness and other information to the warfighter.

Ground control segment typically includes Space Operations Centers (SOCs), antennas, tracking stations, information processing facilities, and communications networks. At times the user segment is separated from the ground segment, as in the case of the GPS where the assets are developed and deployed on ground and air to permit the warfighter to take advantage of the capabilities of the space system. Examples of ground and user segment include Air Force Satellite Control Network (AFSCN) and GPS User Equipment (UE), respectively.

1.4 Uniqueness of Space Segment

Three major differences that make space systems unique and set them apart from a non-space system are: the space environment, unattended operation, and the implications of the ultimate high ground. Each one of these factors is briefly discussed below.

1.4.1 The Space Environment

Space environment is extremely harsh. It places additional constraints on the design and operation of satellites, components, and parts of the space segment. The space environment is characterized by (i) near total vacuum, (ii) ambient thermal inputs varying from direct sun illumination in one direction to the near absolute zero of deep space in others, (iii) passage through belts of charged particles and magnetic fields, (iv) persistent bombardment of protons and electromagnetic radiation from the sun, (v) manmade or natural micro-particles and space debris, to name a few. These constraints must be factored into the design of space assets to assure their long term survival and operation. Special test facilities such as thermal vacuum chambers are required to verify that the hardware can operate in the space environment. In addition, high vibration, acoustic, shock, and other environments during launch and deployment into the operational orbit require careful characterization, design, and testing to prevent irreversible failures during launch and early on-orbit operations.

1.4.2 Unattended Operation

The space segment of all military space systems developed so far operates unattended. If a component fails on-orbit, only remote maintenance actions can be carried out. Such actions must usually be preplanned and take advantage of provisions designed into the SV such as redundant hardware or re-loadable software. Satellites are usually designed to eliminate or at least minimize single point failures. Increasingly, redundancy has been designed into the launch segment as well. Additionally, space parts go through a stringent qualification process for reliability to avoid premature failure and loss or degradation of intended capability. Care is taken to verify that the hardware has a positive margin with respect to the launch and space environments. When a software defect affects operation, the satellite must usually be capable of being placed in a safe mode until the defect can be identified and corrected. Therefore, software that could cause the irretrievable loss of a mission is validated through such steps as extensive simulations, sometimes with flight hardware in the loop. Experience shows that the cost of these steps together with the cost of space launch is perhaps ten times or more the cost of comparable hardware deployed in terrestrial applications. Balancing such factors as performance, cost, and reliability is a SE task for all systems, but the high cost of space equipment places an extraordinary premium on balancing the operational capability to be provided. To achieve balance, alternative approaches or concepts must be compared or traded off against each other with respect to effectiveness, affordability, and risk.

1.4.3 The ultimate High Ground

Military forces have strived for the high ground for millennia because of the advantages it provides including the increased ability to observe or survey the opposition and the operational environment, maintain line of sight communications with friendly forces, and orient oneself with respect to the enemy and the surrounding terrain. Space provides the ultimate high ground so it is not surprising that current military space systems provide for surveillance of both potential enemies and the meteorological conditions in the operational theatre as well as communications and navigation. New systems are being planned or under development to extend these capabilities. But the cost to build and launch satellites means that each must be exploited to the extent practical by all land, sea, and air forces. As a result, many of the space programs are joint programs to provide capability to be used in joint operations by elements of all the military forces and sometimes, in conjunction with allied forces. The user equipment for such systems can become deployed on a wide range of platforms and therefore rival or even exceed the cost of the satellites and launch vehicles so that the SE task of balancing effectiveness and cost can be still more demanding and important. The extreme example is the Global Positioning System (GPS) that provides navigation data via user equipment carried directly by military personnel and on most of the thousands of land, naval, and air platforms operated by the Department of Defense (and also used in a wide range of civil and private applications).

1.5 System of Systems

Traditional systems were individually developed, managed, and operated autonomously with little consideration for the overall enterprise environment and needs to share and reuse capability. Systems were typically developed for a specific service (AF, Navy, Army, Marines) that owned the CONOPS. There was no requirement to interoperate, especially at the enterprise level. Often, severe integration and interoperability problems are encountered when attempts are made to share information and capability among systems built around disparate design philosophies and sometimes proprietary interfaces.

Most modern systems must operate in the context of a broader system of interrelated systems. System of Systems (SoS) is a collection of autonomous systems that link together, pooling their resources and capabilities, offering more functionality, performance, and information superiority than its constituents. Changes in the DoD acquisition policy now emphasizes requirements identification and prioritization for user capability needs at the joint CONOPS or enterprise level. It requires sharing and reuse of information and assets. It mandates net-centric approach to information management and assurance, and calls for system interoperability for cost-effective implementation that leads to an SoS or enterprise view or joint CONOPS for integrated military operations. It is a critical concept for the development of the new and improvement of the old systems. For the DoD, simply providing a capability is no longer adequate. The capability must be provided and justified within the overall enterprise doctrine of warfighting.

The SoS approach does not require new tools or methods; it simply offers a holistic way to meet today's challenge to deliver affordable performance by sharing capability data and other resources.

1.5.1 Essential Characteristics of SoS

Typical behavior and properties of the SoS and its component systems are briefly described here.

- **Operational independence –** component systems of the SoS are typically operated independently.
- **Managerial independence –** component systems of the SoS are typically managed independently.
- **Emergent behavior –** Just like a system is more than the sum of its parts, the interaction of systems in the SoS typically delivers new properties or capabilities.
- **Service-like behavior –** implementation details and methods of operation of component systems, no matter how simple or complex, are hidden with clearly defined and open criteria and standards for acceptance from and delivery of data and materials to their counterpart component systems, users, and external systems and organizations.
- **Interoperable (open) interface behavior –** The SoS component systems typically provide interoperable open interfaces or end nodes for current and future counterpart systems and other entities (i) to receive and deliver capability data to share, enhance, or merge capabilities, and (ii) to provide standardized reusable physical and logical interconnects for integration at all levels of system or product development.

There are other concepts like "family of systems" or "federated systems" that are similar to the SoS construct. Suffice it to say, the differences are usually centered around operational or management control and the expected level of integration and interoperation among the component systems. For more details see DoD's "Systems Engineering Guide for Systems of Systems," 2008.

1.5.2 SMC Space Portfolio in SoS Environment

The SPOs at SMC help develop and build hardware and software for space and ground. For the user, their end product or service for the warfighter is data – space weather, battlefield imagery, navigation, time and communications. However, even today for various understandable reasons, the capability data from these programs is made available to the user typically by means of specialized equipment and terminals with unique data formats that are sometimes proprietary.

Under the Capabilities Based Assessment process and the weight of the insistent mandates to improve cost-effectiveness of space assets, there is ever-present pressure on the SMC SPOs (and elsewhere) to make available capability data in an information-assured net-centric environment, where it can be discovered, shared, reused, manipulated, and enhanced as needed by known and unanticipated but otherwise authorized users. The SMC SE also needs to reengineer acquisition methods to inculcate interface interoperability at each stage of the product development, from the SV constellation that must talk to the enterprise to the interoperable physical interfaces between components.

Highlight Box 1-2
What is Systems Engineering?

INCOSE Handbook, 2004: "An interdisciplinary approach and means to enable the realization of successful systems"

NASA Systems Engineering Handbook, Rev. 1, 2007: "Systems engineering is a methodical, disciplined approach for the design, realization, technical management, operations, and retirement of a system…[It] seeks a safe and balanced design in the face of opposing interests and multiple, sometimes conflicting constraints."

Systems Engineering Tools by Harold Chestnut, 1965: "The systems engineering method recognizes each system is an integrated whole even though composed of diverse, specialized structures and sub-functions. It further recognizes that any system has a number of objectives and that the balance between them may differ widely from system to system. The methods seek to optimize the overall system functions according to the weighted objectives and to achieve maximum compatibility of its parts."

Defense Acquisition Guidebook (DAG) Definition: "Systems engineering is an interdisciplinary approach and process encompassing the entire technical effort to evolve, verify and sustain an integrated and total life cycle balanced set of system, people, and process solutions that satisfy customer needs. Systems engineering is the integrating mechanism for the technical and technical management efforts related to the concept analysis, materiel solution analysis, engineering and manufacturing development, production and deployment, operations and support, disposal of, and user training for systems and their life cycle processes."

In short, the overriding need in the current restricted fiscal environment is for the SMC SPOs to quickly evolve toward an SoS – an enterprise that interoperates to share its assets and products for enhanced effectiveness.

We know backward compatibility is sometimes a roadblock, but it does not have to be. We know cost is always a big consideration, yet a considered evolution toward open interoperable equipment and standards can help curtail system and program lifecycle costs.

1.6 Systems Engineering

Systems Engineering (SE) is the business of integrating materials and methods in an organized and cost-effective manner to conceive, develop, design, implement, and operate a system that fulfills a specified need. It's a big-picture view that considers every aspect of a SE program, from costs and environmental impact, to time lines and life expectancy of equipment. Highlight Box 1-2 shows authoritative SE definitions.

SE is an interdisciplinary method or engineering practice to produce systems, especially when they are complex and are initially not well-defined or understood, like the GPS or the International Space Station. It involves (i) working with the customer early on to define, analyze, and document customer requirements, (ii) develop system model and architecture based on available information, risk assessment, and trade studies, and (iii) creation of sequential build and test plans.

SE views the system holistically in its entirety over its utility or the lifecycle to include the concept, design synthesis, system validation, operation, and eventual disposal at the end of its useful life. It is composed of two sister processes: (i) an SE technical process that helps produce the required capability and system performance, and (ii) an SE management process to help orchestrate technical effort in an organized and phased manner to manage complexity and to monitor progress, risk, and effectiveness.

SE continues to evolve as the systems become more complex and correspondingly more demanding. In particular, SE for military space systems is evolving to keep pace with and support the current space systems acquisition policy.

1.6.1 Extended Role of SoS Engineering

Traditional SE seeks to optimize an individual system (i.e., the product), while SoS engineering pursues optimization of a collection of interacting legacy and new systems (i.e., the enterprise). The enterprise or the SoS brings together current and future systems to satisfy multiple objectives including cost and operational effectiveness. An SoS Engineer has the task to enable the decision-making by developing an understanding of the implications of various choices on technical performance and effectiveness of not only the system itself but also its interaction and interoperability with other current and future systems in the enterprise. This leads to requirements, architecture, and design constraints or mandates that optimize the operation of the enterprise rather than just the system.

SoS management, with the advice of SoS engineers, set policies regarding capability objectives and constraints and may also address the costs applicable to each individual system. In general, such capabilities and constraints either define or lead to technical requirements and constraints that each system must meet. Accordingly, managers for an SoS may have oversight authority over the design and operational decisions for each system.

1.6.2 Useful SE Concepts and Definitions

1.6.2.1 Requirements

A requirement is a documented formal statement that specifies a characteristic, attribute, capability, constraint, or quality of a system that meets stakeholder need to perform a particular function or service. Highlight Box 1-3 shows definition of a requirement, taken from SMC Standard, SMC-S-001. Typically, the operational capabilities to be provided are subsequently translated into verifiable and allocable system technical or engineering requirements by the System Program Office or the Contractor(s) selected to develop the system. The technical requirements must also be completed by deriving the additional requirements and constraints that affect the system and its cost and risk over its life cycle such as the threat, natural environment, and policy and legal constraints. The resulting technical requirements are usually formalized in a System Requirements

Highlight Box 1-3
What is a requirement?

SMC Standard SMC-S-001, 2010: "(1) A condition or capability needed by a user to solve a problem or achieve an objective; (2) a condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents; (3) a documented representation of a condition or capability as in (1) or (2)."

Document (SRD), a Technical Requirements Document (TRD), or a system specification and associated interface control documents or interface specifications. Characteristics of a good requirement include:

- **Unitary or atomic –** The requirement addresses one and only one parameter or thing
- **Unambiguous –** The requirement is objective, concise, and with minimal descriptive matter to assure one and only one interpretation
- **Complete –** The requirement is stated fully in one place
- **Consistent –** The requirement does not contradict any other documented system requirement
- **Allocable –** The top level requirement can be parceled, flowed-down, and assigned to one or more subsystems or components for implementation
- **Traceable –** All allocated and build-to requirements must be traceable to the authoritative and documented need of the stakeholders
- **Necessary and sufficient –** The requirement must be necessary and sufficient to meet the stakeholder need faithfully. (The system is not over- or under-specified.)
- **Feasible –** The requirement must be implementable with the cost, schedule, and technology constraints of the program
- **Contingency and margin –** The requirement allows for a reasonable contingency and margin to reduce risk
- **Verifiable –** The requirement is implementable, its parameters are measureable, and it can be verified by inspection, analysis, demonstration, or test.

There are different types of technical or engineering requirements. The common categories are:

- An **architectural requirement** describes how a system is assembled. It provides a necessary physical, logical, or functional view of a system with clearly defined interfaces.
- A **functional requirement** describes what a system must do. It is simply a task (sometimes called an action or activity) that must be accomplished to provide an operational capability (or satisfy an operational requirement). Some functional requirements that are associated with operations and support can be discerned from the needed operational capability. Others often result only from diligent SE. Experience in SE has identified eight generic functions that most systems must complete over their life cycle: development, manufacturing, verification, deployment, training, operations, support, and disposal. These are known as the eight primary system functions. Each must usually be considered to identify all the functional requirements for a system.
- A **non-functional requirement** describes necessary characteristics of a system to provide expected usability, sustainability, reliability, safety, look and feel, or other specialty engineering attributes.
- A **performance requirement** describes how well a system must perform. It is a statement of the extent to which a function must be executed, generally measured in terms such as quantity, accuracy, coverage, timeliness, or readiness. The performance requirements for the operational function and sometimes a few others often correlate well with the statement of the needed operational capability, like those developed through the JCIDS process. The statement of other performance requirements usually requires thorough SE.

- A **constraint**, as the word implies, is an imposed requirement such as an interface requirement (e.g., the interface between a launch system and a satellite system that constrains the design of both systems), policy, public law, or the natural or threat environment. An important new type of constraint is derived from the mandates to interoperate with the enterprise. This leads to the need to comply with and incorporate existing and typically open standards for sharing data and materiel.

System technical requirements including constraints result in both allocated and derived requirements. Allocated requirements flow directly from the system requirements down to the elements of the system. Derived requirements depend on the design solution (and so are sometimes called design requirements). They include internal interface constraints between the elements of the system.

1.6.2.2 Baselines

SMC-S-001, System Engineering Requirements and Products, defines a baseline as a set of "document(s) or decision database(s) that record the current set of requirements for the system and its design or system product solutions." A baseline is the primary product of the SE process. It is the documented and accepted current state of the system at any time. During the life cycle of a system a number of baselines are defined and maintained formally at appropriate transition points that include:

- **Requirements baseline –** establishes system requirements traceable to and validated against stakeholders' capability needs
- **Architectural/functional or allocated baseline –** establishes requirements allocated to functional or architectural blocks traceable to and validated against the requirements baseline
- **Design baseline –** establishes detailed build-to specifications for the system
- **Product baseline –** establishes implementable or as-built design and process requirements for the system

1.6.2.3 Configuration Control and Management

Configuration management is the process of managing change in the state of a system or product hardware, software, firmware, documentation, measurements, and other significant data. As change requires an initial state and next state, the identification of significant states within a series of several changes is important. This is established by defining and maintaining a baseline as discussed earlier. The identification of significant states within the revision history of a configuration entity is the central purpose of baseline identification.

1.6.2.4 Interfaces

SMC-S-001 defines interface as "the boundary between two or more systems, functions or other logical representations, or system products or between a system and a facility at which interface requirements or constraints are set. Interfaces can be physical or functional." When the interface for a new system is to an existing system, the interface is a constraint on the design of the new system. Even when systems or system elements are designed in parallel but by separate design organizations, a point is reached in the development process where the interface eventually becomes a constraint on each design. As a result, interfaces are usually viewed as constraints.

Complex systems have many interfaces, both internal and external. Common well-defined interfaces are desirable since they:

- reduce system complexity
- help with system architecture and design by offering clean boundaries between subsystems
- clear interface identification and definition reduces risk

- provide clean testable boundaries for verification and validation of subsystems before integration
- allow development of subsystems to proceed in parallel at physically different locations and by different government and contractor organizations
- allow for interoperability between current and future systems within the context of the enterprise

Similar to the relationship between two systems, the interfaces between subordinate elements of a system evolve into constraints on each element as the development process proceeds. The interfaces between systems are sometimes referred to as the external interfaces; those within a system are called internal interfaces. Formally establishing and controlling internal interfaces is particularly important when the elements are designed by separate design teams such as groups with different engineering specializations or subcontractors.

As examples of interfaces, the Personal Computer discussed above usually must interface with a number of other systems including the source of electrical power to which it connects, other peripherals like the printer, and adaptor cards such as those that provide for connection to the Internet or other networks. The Personal Computer also includes internal interfaces such as between the mother board and the power supply. All of these involve both physical and functional interfaces.

Interfaces can be physical or functional. Physical interfaces include definitions of the means of attachment (bolt patterns, connectors, fasteners, etc.) and keep-out volumes. Figure 1-2 shows some of the common interfaces. These interfaces can include:

- **Physical connection –** when two parts (i) directly touch each other like rollers, brake pad and disk, finger and touch screen, (ii) have a reversible connection like electrical or data connectors, latch mechanism, or nuts and bolts, or (iii) are permanently connected like rivets and spot welds
- **Mass flow –** when matter is exchanged between two subsystems or components to include gases, fluids, and solids to include fluids like air, coolants, paper. It typically implies an underlying physical connection.
- **Energy flow –** when there is a net exchange of work between two components like exchange of electrical, thermal, EM, or mechanical power. It typically requires a physical connection but not always. RF antennas send and receive EM energy or solar cells generate current without physical connection.
- **Information flow –** when systems (software, electro-mechanical, sensors, actuators, controllers) exchange information (telemetry, command) to perform their appointed function.

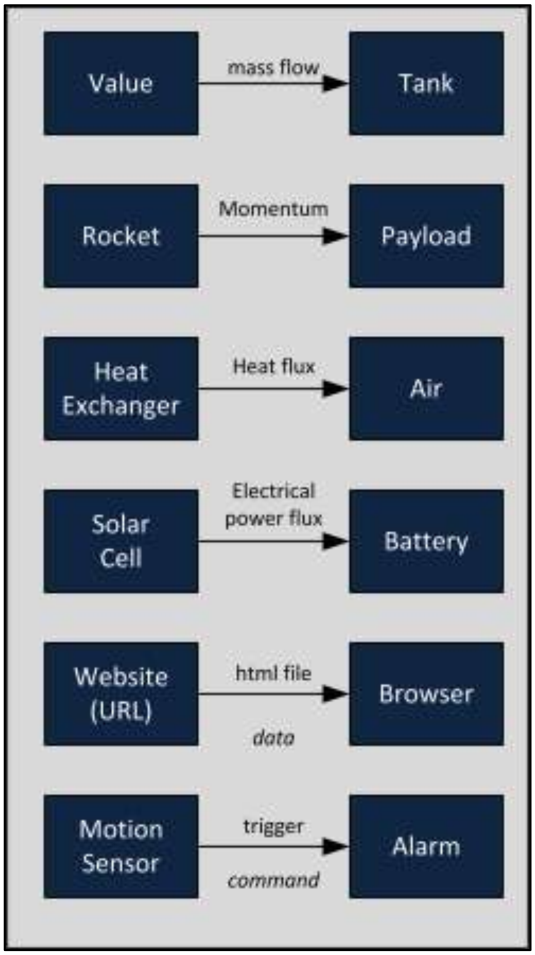


Figure 1-2 Examples of common interfaces

The interface between two systems managed by different organizations – such as a satellite system and a launch system – may be captured in an interface specification or in an Interface Control Drawing or Document (ICD). Figure 1-3 shows major elements, typically produced at different facilities, of an integrated space and launch vehicle. Well-thought interfaces for these elements are developed in advance to avoid problems at the integration facility. Interface between two elements of a single system developed by different design groups or subcontractors may be captured in an Internal ICD (IICD). Interfaces that are managed by a single organization may simply be captured in the design drawings.

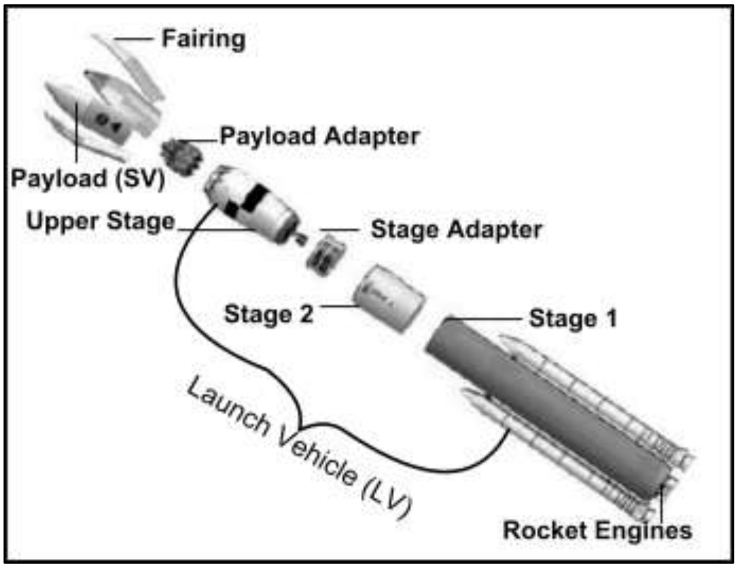


Figure 1-3 Major interfaces of an integrated space and launch vehicle

As another example of an interface, a space launch system may use a liquid fuel and oxidizer. To achieve the planned performance, the liquids must meet certain requirements for purity, density, stability, etc. Such “interface” constraints are usually defined in specifications or standards to help ensure the needed launch performance.

Some interfaces have become standards used throughout an industry or even throughout much of the world. For example, the IS-GPS-200 is a world-wide standard to receive and consume GPS Standard Positioning Service (SPS). Often global standards are developed and maintained by international organizations like the Institute for Electrical and Electronics Engineers (IEEE) or the International Telecommunications Union (ITU) with on-going support from contributing industries and nations. Because of the problems that can result from informally defined interfaces, the interfaces for most military systems are defined by specifications, ICDs, or standards published by independent standards organizations.

1.6.2.5 Interoperability

Interoperability is the property of an entity that is fully documented in an open physical or functional interface to allow for well-defined but unrestricted interaction with other existing or future entities. Entities can be System of System, System, system element, subsystem, organization, process, or mission – when they have an interface to share. The standardized interfaces are designed for the purpose of sharing, reusing, or merging data and materiel thus providing enhanced emergent capability of the combined interoperating entity. Typically, the use of a fully documented interface obviates the need of specialized and unique adaptation devices, equipment, software, or hardware. Interoperability enables form, fit, or function reuse and interchangeability as well as cost-effective enterprise operations over disparate entities. Highlight Box 1-4 provides DoD interoperability definitions.

The elements can be either functional or physical and, if physical, can be hardware, firmware, or both. Alternatively, architecture may refer to some high-level attribute of a system such as openness or interoperability.

DoD has mandated acquisition strategies to foster and develop open standards based program architectures. These strategies are adapted by SMC programs to help achieve greater portability, interoperability, compatibility, reusability, maintainability, scalability, vendor independence, ease of technology insertion, and user productivity at reduced lifecycle cost. However, we are remiss if we do not emphasize that the success of OSA depends strongly on advocacy of its concepts with the combatant commanders, the SPO directors at SMC, and the vendors alike.

For the systems engineer at the SPOs, the interoperability issues extend far beyond the capability data. Here the SE must contend with standardization of both the physical and the logical interfaces at every stage of the product development to find cost-effective solutions for interoperability and integration. For example, consider the power subsystem on a SV: it needs to have common voltage, current, harnesses, connectors, and monitoring devices to deliver well-conditioned power to all other subsystems. A standardized electrical power subsystem can offer savings in cost and development time for the SV, especially if it can be reused by other SV development programs.

Interoperability is further discussed in Section 4.9.

Highlight Box 1-4
What is Interoperability?

Open Systems Joint Task Force (OSJTF): " The ability to (1) interchange and use information, services and/or physical items among components within a system (platform, program or domain) and (2) support the common use of components across various product lines."
DoDD 5000.01: The ability of systems, units, or forces to provide data, information, materiel, and services to and accept the same from other systems, units, or forces, and to use the data, information, materiel, and services so exchanged to enable them to operate effectively together.
DoDD 4630.05: "The ability of systems, units, or forces to provide data, information, materiel, and services to and accept the same from other systems, units, or forces and to use the data, information, materiel, and services so exchanged to enable them to operate effectively together. IT and NSS interoperability includes both the technical exchange of information and the end-to-end operational effectiveness of that exchange of information as required for mission accomplishment. Interoperability is more than just information exchange. It includes systems, processes, procedures, organizations and missions over the life cycle and must be balanced with information assurance."

1.6.2.5.1 Net-centric Operations

DoD's emphasis on net-centric operations seeks to translate information advantage into a competitive advantage through robust networking of well-informed geographically-dispersed forces. The focus is on five key areas to realize a net-centric information sharing vision: (i) data and services, (ii) secured availability, (iii) computing infrastructure readiness, (iv) communications readiness, and (v) Network Operations (NetOps) agility. DoD envisions acquisition of services and systems that are secure, reliable, interoperable, and able to communicate across a universal information infrastructure based on Internet Protocol (IP) and related non-proprietary and vendor-neutral standards to meet the challenges of modern warfare. This internetworking, combined with changes in technology, organization, processes, and people allows new and robust forms of organizational behavior.

All SMC programs classified as National Security Systems (NSS) are required to comply with CJCSI 6212.01, "Net Ready Key Performance Parameter (NR-KPP)." Adherence to net-readiness in design and Net-centric operations forms the backbone of DoD's system of systems vision for enterprise-wide interoperability. SMC considers net-centric approach to be of great importance for enhancing space systems performance and, as such, is cooperating with the Space and Naval Warfare Systems Command (SPAWAR) and other organizations to develop tools, requirements, and implementable guidance that is available through the Net-Centric Enterprise Solutions for Interoperability (NESI) websites. This information is critical to develop some of the required program documents such as Information Support Plan (ISP), Net-centric Data Strategy (NCDS), and the Information Assurance Strategy (IAS).

1.6.2.6 System Architecture

Many of the ideas discussed so far are often brought together or are visualized as the system architecture where expected capabilities are matched with specific physical or functional constructs or blocks with well-defined and typically widely accepted and open interfaces. The first thing to know about the word architecture is that it can have many different meanings – the meaning in a particular instance must be discerned from the context or the user's definition. Webster offers five definitions starting with ones having to do with designing buildings and other structures.[1] During the 1960s, the term was extended to computer systems where it generally referred to the way the electronic hardware was organized to process software instructions that facilitated the important idea of evolving the architecture to provide upward compatibility as new models and generations were developed. More recently, it has been extended to apply to all systems; hence, the term system architecture.

A book on system architecting identifies eight different definitions of system architecture published by various technical organizations and authors.[2] Most of these definitions have to do with some representation of the elements of a system and the way they are structured, interconnected, or organized. Thus, a functional architecture usually refers to some representation of the tasks or functions that a system is to perform and the organization or relationship among the functions. Similarly, a physical architecture usually refers to some representation of the structure or organization of the physical elements of the system. The elements of a physical architecture can represent hardware, software, or both. As will be discussed more under SE below, a functional architecture is sometimes developed and mapped to the physical architecture to better define and understand the design requirements for the physical elements.

The official definition of architecture in the Joint Capabilities Integration and Development System (JCIDS) captures the notions of both "systems architecture" and "architectural standards" discussed above: "the structure of components, their relationships and the principles and guidelines governing their design and evolution over time."

To summarize, the term system architecture may refer to the elements of a system and the way they are organized. The elements can be either functional or physical and, if physical, can be hardware, software, or both. Alternatively, architecture may refer to some high-level attribute of a system such as openness or interoperability.

DoD has mandated acquisition strategies to foster and develop open standards based program architectures. These strategies are adapted by SMC programs to help achieve greater portability, interoperability, compatibility, reusability, maintainability, scalability, vendor independence, ease of technology insertion, and user productivity at reduced lifecycle cost. However, as noted earlier the success of OSA depends strongly on advocacy of its concepts with the services and the vendors as well as within the SMC and its various programs to acquire economical warfighter assets in space and elsewhere.

1.6.2.7 Modular Open System Approach

Open Systems Joint Task Force (OSJTF) defines Modular Open System Approach (MOSA) as "an integrated business and technical strategy that employs a modular design and, where appropriate, defines key interfaces using widely supported, consensus-based standards that are published and maintained by a recognized industry standards organization." And, it further defines an open system as "a system that employs modular design, uses widely supported and consensus based standards for its key interfaces, and has been subjected to successful validation and verification tests to ensure the openness of its key interfaces." MOSA supports achieving the following:

---

[1] Merriam Webster's Collegiate Dictionary, Tenth Edition p. 61, as quoted in Maier, Mark W. and Eberhardt Rechtin, The Art of Systems Architecting, 2nd edition, CRC Press, 2002, p. 284.

[2] Maier, Mark W. and Eberhardt Rechtin, The Art of Systems Architecting, 2nd edition, CRC Press, 2002, p. 285ff.

- reduced acquisition cycle time and overall life-cycle cost
- ability to insert cutting edge technology as it evolves
- commonality and reuse of components among systems
- commonality and reuse of widely accepted and industry supported open interfaces for current and to-be interoperability
- increased ability to leverage commercial investment

Responding to changes in national policy, modernization needs, user requirements, mission application constraints, and DoD mandates on open standards, SMC emphasizes the use of open standards to reduce system acquisition cost, foster vendor competition, and reduced program risk. There are six basic elements of an open architecture that are described below:

- **Open standards –** Parts, modules, objects, products, and systems are based on vendor-independent, non-proprietary, publicly available, and widely accepted standards. Standards allow for a transparent environment where users can inter-mix hardware, software, and networks of different vintages from different vendors to meet differing needs. Selection of open standards should be based on sound market research and due consideration must be given to the DoD mandated standards as listed in DoD Information Technology Standards Registry (DISR) when applicable.

- **Interoperable –** The ability of systems, units, or forces to provide and receive services from other systems, units, or forces and to use the services so interchanged to enable them to operate effectively together. Open standards by definition support and enforce interoperability.

- **Interchangeable –** The ability of two or more parts, modules, objects, or products to be transparent replacements for one another without other changes in hardware or software. This property provides opportunities for upgrades and technology insertion.

- **Portable –** The ability of two or more systems or components to exchange and use information or the ease in which a system or component can be transferred from one hardware or software environment to another.

- **Modular –** Physical or logical modularity to meet functional requirements.

- **Scalable –** The ability to grow (and interlink hardware and software) to accommodate increased loads.

Industry groups or academic/engineering organizations build many open standards, yet other “widely accepted” commercial standards start their life as “mandates.” The PCI bus was forced on the electronics industry by Intel. ActiveX and other Microsoft inventions have become de facto but largely proprietary standards with published interfaces. The Internet, currently supported by world-wide web consortium (W3C) standards and protocols, is a shining example of global interoperability that was supported by the DoD in its infancy. DoD is not new to the “standards” game. Many of the DoD-built open and non-proprietary standards now form the basis of ANSI, IEEE, ASME, and other civil and commercial open standards. For example, Global Positioning System's ICD-GPS-200 is a non-proprietary and open standard that provides the basis for a multi-billion dollar commercial GPS user equipment industry.

Standards building is not an easy task and is usually a high risk activity for any program. Only a few percent of the standard building efforts actually succeed. For example, the Inter-Services Digital Network (ISDN) standard took about 15 years to get to the market even when it had the government-like big phone companies fully behind it. While the ISDN was successfully built, it failed in the marketplace as the technology was already too old and too expensive. This is why the adoption of existing commercial open standards, if they meet the program capability and performance requirements, makes great sense.

DoD use of an open systems approach reduces the cost/risk of ownership of weapons systems, delay system obsolescence, and allow fielding of superior warfighting capability more quickly. An open systems approach reduces weapon system cost through the use of widely accepted standard products from multiple suppliers in DoD weapon systems. If program managers define weapon system architecture by specifications and standards used in the private sector, DoD can leverage the benefits of the commercial market place, and take advantage of the competitive pressures that motivate commercial companies to improve products and reduce prices. Program managers can then have access to alternative sources for key subsystems and components to construct DoD weapon systems. The open systems approach could reduce the DoD investment early in the weapon system life cycle because some of the required systems or components may be available or under development without direct DoD investment. Also, program managers can competitively select production sources from multiple competitors. Additionally, an open systems approach delays system obsolescence by allowing program managers to incrementally insert technological improvements into existing or developing systems rather than having to make large-scale system redesigns or to develop new systems. Further, an open systems approach enables program managers to deliver weapons systems to warfighters more quickly as a result of reduced developmental effort.

In summary, open systems with modular standard interfaces provide for the ability to

- add new or improve existing capabilities through planned and unplanned incremental improvements
- to integrate entities and enable commonality, portability, and interoperability
- replace items with high replacement frequency and cost

1.6.2.7.1 DoD Information Technology Standards Registry (DISR)

DoDD 4630.05 states, "The DISR provides the minimal set of rules governing the arrangement, interaction, and interdependence of system parts or elements, whose purpose is to ensure that a conformant system satisfies a specified set of requirements. It defines the service areas, interfaces, standards (DISR elements), and standards profiles applicable to all DoD systems. Use of the DISR is mandated for the development and acquisition of new or modified fielded IT and NSS systems throughout the Department of Defense."

The DISR contains a minimal set of primarily commercial information technology standards. Use of the DISR facilitates interoperability among systems and integration of new systems into the DoD enterprise. Additionally, the DISR provides the capability to build profiles of specific standards that programs can use to deliver net-centric capabilities. Not all are applicable to SMC space systems acquisition. However, a number of standards for networking, net-centric operations, and information assurance are mandated for space systems to assure enterprise-wide interoperability. The standards are available on the web along with other useful information including resources to develop information support plan (ISP).

1.6.2.8 DoD Architecture Framework (DoDAF)

DoD Architecture Framework (DoDAF) is a data-centric comprehensive conceptual model to facilitate the ability of DoD managers at all levels to make key decisions more effectively through organized information sharing within and across the Joint Capability Areas (JCAs), Mission, Component, and Program boundaries. For example, it supports the DoD Chief Information Officer (CIO) in his responsibilities for development and maintenance of architectures required under the Clinger-Cohen Act. It also provides extensive guidance on the development of architectures supporting the adoption and execution of net-centric services.

Development, design, and implementation of space systems and system of systems invoke complex iterative processes. The DoDAF views offer invaluable engineering information to support these processes. The operational view identifies what needs to be accomplished and who does it. The systems and services view relates systems, services, and characteristics to operational needs. The technical standards view prescribes standards and conventions. The three views and their interrelationships – driven by common architecture data elements – provide the basis for deriving measures such as interoperability or performance, and for measuring the impact of the values of these metrics on operational mission and task effectiveness.

DoDAF offers a common approach for architecture description development, presentation, and integration. It is intended to ensure that architecture descriptions can be compared and related across boundaries, without prescribing any particular methodology or process for creating the actual architecture model, but only the elements and relationships that any given methodology would use.

By offering a uniform set of architectural products or viewpoints, DoDAF helps promote open design and interoperable solutions for the enterprise. Process owners and managers, within their area of authority and responsibility, have the discretion to select and implement a set of DoDAF architectural viewpoints to meet their SE and management needs, requirements, and enterprise-level mandated constraints.

CJCSI 6212.01 mandates the development of a number of DoDAF viewpoints over the lifecycle of a program. DoDAF is also consistent with DoDI 4630.8 requirements for integrated architectures. These customized views, and the models that utilize the data, enable the architecture information to be communicated to, and understood by, stakeholders in diverse functional organizations.

1.6.2.9 Service-Oriented Architecture

Much like DoDAF, SOA proposes a device-independent, standards-based, and transparent approach to architecting that highlights the concept of a service. Though the concepts have evolved and are proven for software and Information Technology (IT) systems, they can be applied and put into practice at space systems level. However, it is challenging to see and package legacy SMC capabilities like weather, command and control of satellites, or navigation as services.

Some of the common characteristics are:

- Service performs certain function(s) or makes available certain capabilities for the user
- Service user can be known or unanticipated but otherwise authorized
- Service user can be human, or machine, or other services
- Service functionality is made available through open standard interfaces
- Service is provided and complies with a specified set of constraints and policies
- Service source, description, and its vocabulary is registered and advertised with the enterprise network registry for discovery like the DoD Metadata Registry
- Service communicates with other services using standard protocols for both humans and machines
- Services are self-contained, reusable, and easily distributed
- Services are loosely coupled from users to reduce integration costs
- Services offer capabilities independent of their implementation
- Services shield users from implementation details to include software, hardware, and processes
- Service may allow access to only a subset of capabilities of the system

1.6.2.10 Compatibility – Backward and Forward

Systems that last for many years often change a great deal over those years. New technologies emerge and specific needs change over time; thereby, creating the impetus to change deployed system components to use new technologies and satisfy new needs.

As the new components are developed and deployed, they often have to still work productively with the other parts of the system, sometimes deployed decades before. This is called backward compatibility. Backward compatibility typically requires that the newly developed components allow the other parts of the system to function and perform at least as well as they did before the new components were deployed. However, backward compatibility is at its best when the newly developed components improve the functionality or performance of the existing components, thereby improving the entire system. Achieving backward compatibility is often a great challenge.

On the other hand, with forward compatibility, the design of the system takes into account potential changes that can be predicted to occur over the life of the system. One should design in forward compatibility from the beginning of the program, to ensure relatively less cumbersome backward compatibility in the future. This could include changes in needs, improvements in technology and algorithms and desire to scale for more capacity. Once these potential improvements are identified, they can be designed into the system with modular and scalable designs, flexible hardware, or reprogrammable software and room for growth in computer systems. Another way to improve forward compatibility is to use open standards. Open-standards as mandated in DISR or described by MOSA propagate standards that tend to provide compatibility over decades while improving functionality.

1.6.3 SE Models

SE is usually described in terms of process flows. The system engineer's choice of the process model usually depends on the application or the type of the system. Yet, the model must be phased to control and monitor risk, the relationship between the phases must be clearly identified, and the process must incorporate feedback within and between the phases as an integral part of the program. Each formal SE process model provides staged means to manage cost, schedule, and technology. The process is transparent and documented, and allows for formal and informal communications and interactions among various stakeholders – customer, operator, user, engineer, and manager. System or program documents are created and updated as necessary like the Technical Requirements Document (TRD) to maintain clear and current definition of the system for all stakeholders.

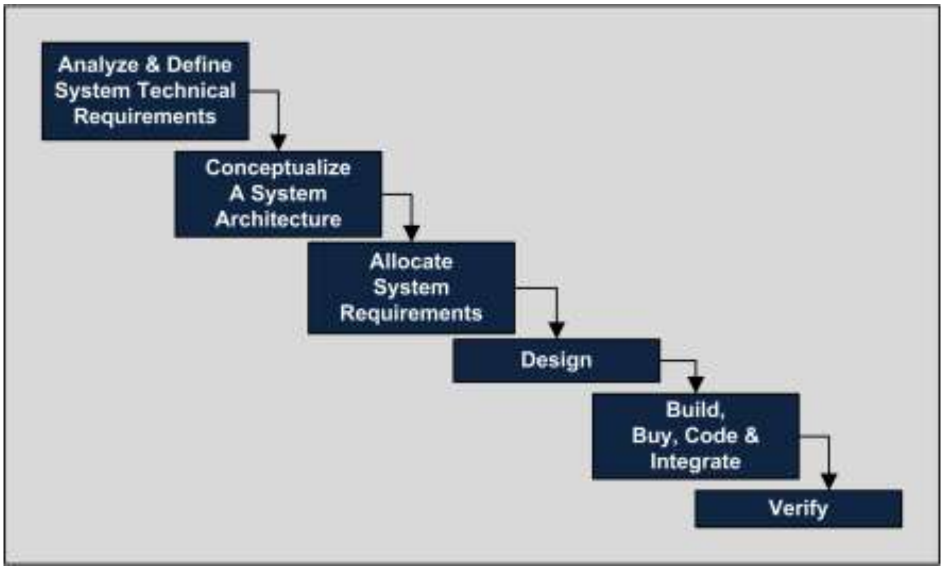


Figure 1-4 Waterfall engineering process to develop a system

Examples of such models include the Waterfall model, the V model, the object-oriented analysis/ development (OOA/D), the model-based SE (MBSE) model, and the Spiral development model. As systems become increasingly more software oriented and the requirements for the systems to interoperate at the enterprise level that include data sharing and reuse become more stringent, object-oriented modeling has gained more prominence in recent years. At times, depending on the project needs or maturity a hybrid process can be applied borrowing features from the standard models. As sponsors of systems of unprecedented complexity, the military has evolved mandatory SE process requirements to ensure greater probability of success for its programs.

1.6.3.1 The Waterfall Model

The waterfall method is a sequential development process as shown in Figure 1-4. It emphasizes documentation to maintain discipline and control change as the project moves from the concept development to the build and test steps.

This method is especially appropriate for large and complex systems that require a formal process to control and manage cost, schedule, and technology. A formal process for system design is based on transparent processes and documented and traceable communications or interaction among the customers, users, engineers, and other stakeholders. To formalize the relationship between the customers or users and the engineers, SE usually starts with the system technical requirements that drive the engineering design or response. The system technical requirements state the customers or users purpose for the system, i.e., what they need or desire the system to do. They also include the needs or desires of other stakeholders such as program decision makers and the constraints imposed by the environment which the system will operate – the natural environment and, in the case of military systems, the threat environment – and the interfaces with other systems. Through analysis, SE seeks to define system technical requirements that completely and accurately capture the need and all other requirements and constraints such that compliance of the resulting system can be objectively verified by test or other means.

To enforce and clearly delineate the relationship between multiple teams of engineers, SE focuses on allocating the system requirements to the system constituents (e.g., segments or subsystems) to be designed by each team.

But before the allocation can take place, the SE must conceptualize a system architecture, i.e., the definition and organization of the system elements that will act together to achieve the purpose of the system, i.e., to meet the system capability or technical requirements.[3] The system technical requirements are then allocated to each of the elements of the conceptual architecture to provide a framework for design.

It was noted that SE is a technical art form that does not offer a direct and unique way to arrive at the design of a complex space system. Similarly, there is no prescribed or fixed method for the systems engineer to define the system technical requirements, or the system concept and architecture, or to allocate the system requirements to the system elements. If a system element is of sufficient complexity, the art of SE is applied in turn to it. In this sense, SE process is applied repeatedly or recursively. Recursion usually continues to define lower tier system elements to the point that a single engineering team can do the design. The system technical requirements are then allocated to each of the elements to guide their design by each of the teams. The hardware specified by the design is then built (manufactured) or bought, the software is coded, the system is integrated, and the design is verified through test or other means to confirm that it satisfies or meets the system technical requirements.

But most systems, especially most military systems, are of such complexity that an initial pass through the steps is inadequate to arrive at a design that meets the intended purpose along with other objectives such as affordability and reliability. Instead, the practicing systems engineer usually finds it necessary to iterate, usually a substantial number of times. A given iteration may be confined to a single step, may involve several steps, or all of the steps. The need to iterate is a direct consequence of the fact that SE is an art, not a science.

---

[3] Some consider system architecting as a separate undertaking from systems engineering. This primer is based on the view that both are necessary and should be integrated into a single process. The process is called systems engineering here in keeping with long standing tradition in military programs.

Each iteration is guided by (i) SE analyses, (ii) SED requirements, (iii) trade-offs that compare alternative statements of the system technical requirements, (iv) alternative system concepts or architectures, (v) alternative requirements allocations, and (vi) alternative designs to achieve a balance between such factors as effectiveness, cost, schedule, and risk. Achieving and maintaining a focus on the balance is essential to the success of a military acquisition program. An unbalanced emphasis on cost, schedule, technology, a specific technical capability, or other factor can often result in a poor solution.

The results of the iterations and associated tradeoffs may also be helpful to the customers and users. For example, if it is found that the cost to meet the ultimate need is prohibitive or the risk is high because of limitations in the available technology, then the users may wish to identify an initial increment that is now affordable and feasible and defer fulfilling the ultimate need for later.

The iterations and associated tradeoffs are planned to provide increasing specificity and completeness so that key decisions are first made for the system technical requirements, then for the system concept and requirements allocation to each element in the concept, and finally, for the design. As more information becomes available about the feasibility of the design to meet the requirements in a way that balances cost, schedule, technology risk, and other factors, it may become apparent that a more optimal system can be built by changing a requirement, allocation, or a design choice. Hence, a concurrent formal process to manage or control change is essential to a SE process.

It is also important from the outset to guard against unintended consequences such as unsafe operation, high failure rates, or electromagnetic interference (EMI). As a result, SE must also provide for the integration of specialists in safety, reliability, IA, EMI, and other engineering areas to help define and allocate the requirements, complete the design, and verify that the design satisfies the requirements.

1.6.3.2 The V-Model

The V-model offers a complementary view of SE, as depicted in Figure 1-5. This model is ideal where project requirements are developed before technology choices are made and the system is implemented. It provides a top-down recursive approach to develop the system at increasingly detailed levels followed by a bottom-up recursion during assembly and integration.

On the left side of the V, the system definition progresses from a general user view of the system to a detailed specification of the system design. The system is further decomposed into subsystems and components through repeated application of a decomposition process. In the process, the more general user or system requirements become increasingly specific and detailed as are allocated to the system components. These requirements are documented in a series of system baselines that support the steps to follow.

On the right side of the V, the hardware and software is implemented to start the bottom-up process of integration and verification that leads to the final system. The final system is then validated against the stakeholder’s needs.

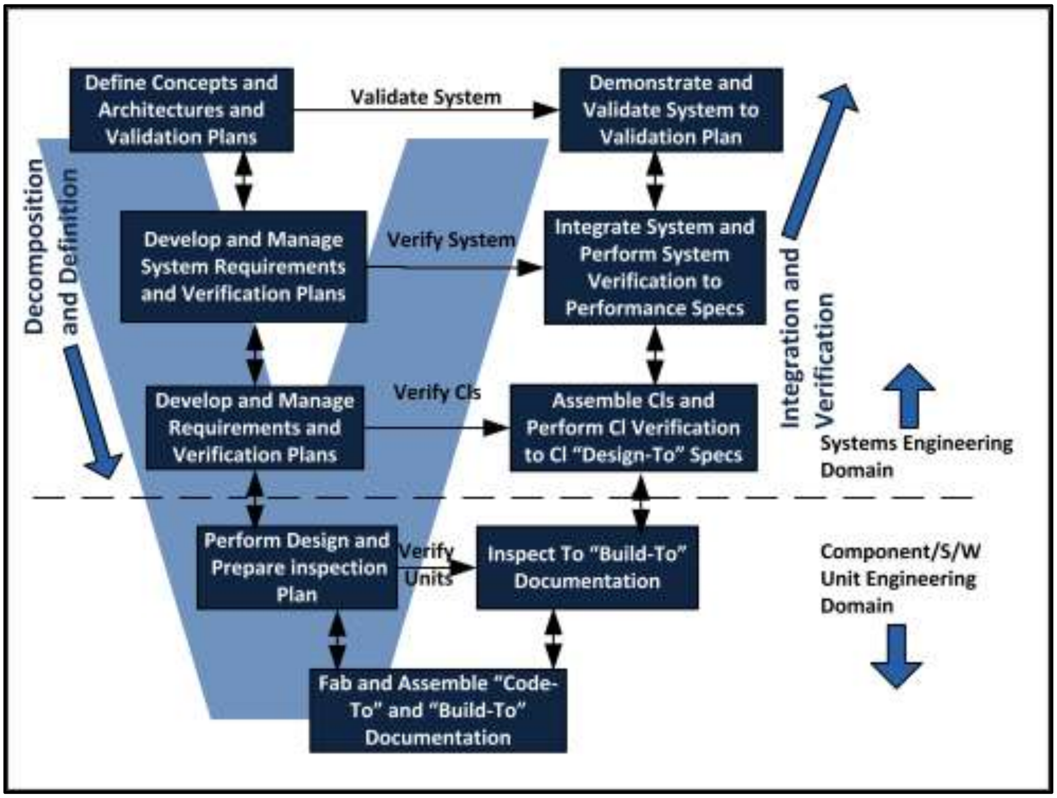


Figure 1-5 The V-Model

1.6.3.3 Object-oriented Analysis and Design

An Object-oriented Analysis and Design (OOA/D) model views, analyzes, and designs a system as a set of interacting and interrelated objects – almost anything including structures, functions, devices, events, roles, organizations, and external entities can be viewed as objects. OOA/D (i) helps manage complexity by focusing on the essential parts of the system – abstraction, (ii) separates and hides object implementation details from other objects – encapsulation, (iii) offers modularity, and (iv) a hierarchical ordering of abstraction in tree-like structure. Each of the objects in a system has a unique identity, a well-defined structure, and a state representing its condition. Furthermore, an object has a clearly defined behavior characterized by its functionality and its interface.

The model was developed primarily to deal with and is currently the most preferred methodology for complex software development projects. As systems have become increasingly more software intensive, OOA/D model and its vocabulary has become an integral part of the SE repertoire, as codified in the Unified Modeling Language (UML)[4].

1.6.3.4 Model-based Systems Engineering

Model-based SE (MBSE) processes can be applied during the system requirements development and allocation phase of the program as well as during the design stage for component modeling activities. MBSE

---

[4] UML specification is maintained by the Object Management Group (OMG), see www.uml.org

methods can be used alongside the traditional models for greater and quicker understanding of the SE tasks at hand, especially when physical modeling and testing of components are cumbersome or too expensive.

One of the rigorous and comprehensive approaches to MBSE has been developed by the Object Management Group (OMG) and INCOSE jointly. Here, mature modeling concepts as embodied in the UML are extended to create a general-purpose MBSE language, the System Modeling Language (SysML™)[5]. SysML™ is a graphical modeling language for specifying, analyzing, designing, and verifying complex systems. SysML™ reduces UML's software centric restrictions and adds two new diagrams – requirements and parametric – to greatly extend the appeal of OOA/D methods for SE. See Chapter 5 for more information on modeling and simulation, SysML™, and UML.

1.6.3.5 The Spiral Model

The spiral development method is an iterative process for developing a defined set of capabilities within one increment. This process provides the opportunity for interaction between the user, tester, and developer. In this process, the requirements are refined through experimentation and risk management. There is continuous feedback, and the user is provided the best possible capability within the increment. Each increment may include a number of spirals. An increment or block is a useful and supportable operational capability that can be effectively developed, produced or acquired, deployed, and sustained. Each increment of capability has its own set of thresholds and objectives set by the user. The customer or user controls the decision to field, continue development, or the pace at which the capability requirements are met.

The use of spiral development allows more innovative use of SE principles to field capabilities to the user. It emphasizes mitigation of technical risk by maturing technology before its application to projects. It encourages the need to apply proper SE methods to achieve desired capability over time using flexible system architecture. The overall consequence of the spiral development process to the acquisition programs is that:

- The system requirements can be revisited as system is developed
- The technology insertion opportunities are expanded, but require forethought into architecture and when to insert
- There is emphasis on SE to ensure flexibility in implementation
- The funding and program forecast must coincide with plan or program will stall
- The testing strategy must be innovative to provide flexibility

In recent years Boehm[6] and others have proposed modifications to the original spiral model. One such improvement, as summarized in Figure 1-6, is the Win-Win Spiral Model.

The model includes the following strategy elements:

- **Success-critical stakeholders' win criteria –** all of the project's success-critical stakeholders participate in integrated product teams (IPTs) or their equivalent to understand each other's needs and to negotiate mutually satisfactory (win-win) solution approaches.
- **Risk management –** The relative risk exposure of candidate solutions and the need to resolve risks early drives the content of the spiral cycles. Early architecting spirals likely will be more analysis-intensive; later incremental or evolutionary development spirals will be more product-intensive.

---

[5] SysML™ specification is maintained by the OMG, see www.omgsysml.org

[6] Barry Boehm, Understanding the Spiral Model as a Tool for Evolutionary Acquisition, 25 January 2001

- **Spiral anchor-point milestones –** These focus review objectives and commitments to proceed on the mutual compatibility and feasibility of concurrently engineered artifacts (plans, requirements, design, and code) rather than on individual sequential artifacts.
- **Feasibility rationale –** In anchor-point milestone reviews, the developers provide a feasibility rationale detailing evidence obtained from prototypes, models, simulations, analysis, or production code that supports a system built to the specified architecture that (i) support the operational concept, (ii) satisfies the requirements, (iii) is faithful to the prototype(s), (iv) is capable of being built within the budgets and schedules in the plan, (v) has all major risks resolved or covered by a risk-management plan, and (vi) has its key stakeholders committed to support the full life cycle.

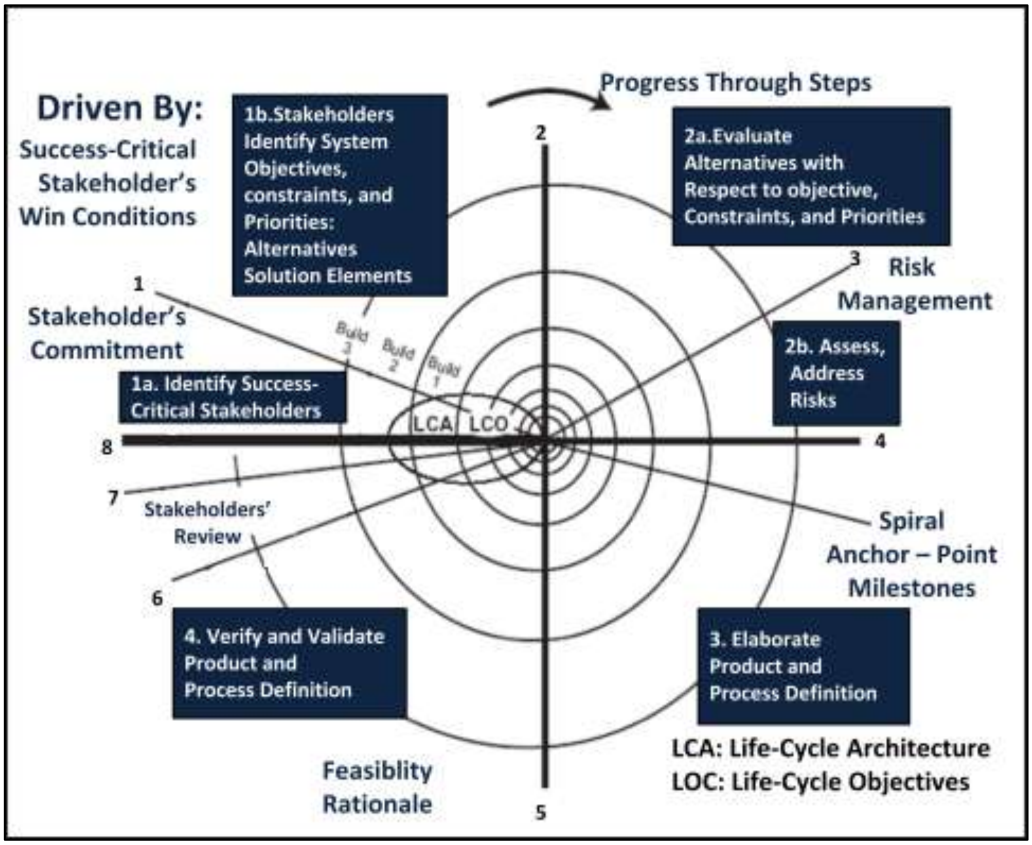


Figure 1-6 Win-Win Spiral Model

1.6.3.6 Evolutionary Acquisition Model

Evolutionary acquisition strategy defines, develops, produces or acquires, and fields an initial hardware or software increment (or block) of operational capability. It is based on technologies demonstrated in relevant environments, time-phased requirements, and demonstrated manufacturing or software deployment capabilities. These capabilities can be provided in a shorter period of time, followed by subsequent increments of capability over time that accommodate improved technology and allow for full and adaptable systems over time. Each increment meets a militarily useful capability specified by the user. However, the first increment may represent only a partial fulfillment of the desired final capability. There are two basic approaches to evolutionary acquisition. In one approach, the ultimate functionality can be defined at the beginning of the program, with the content of each deployable increment determined by the maturation of key technologies. In the second approach, the ultimate functionality cannot be defined at the beginning of the program, and each increment of capability is defined by the maturation of the technologies matched with the evolving needs of the user. The salient features of evolutionary acquisition include:

- Strategy to provide core capability quickly (60%-80% of full requirement)
- Followed by increments to deliver full capability
- Multiple increments can occur simultaneously
- Not all increments fully determined up front (spiral approach to evolutionary acquisition)
- Each increment based on technologies demonstrated in relevant environments
- Each increment proved to be militarily useful (use of Operational Architecture(s) (OAs) where appropriate)
- Time-phased requirements (Capability Development Document(s) (CDDs) and Capability Production Document(s) (CPDs) for each fielded increment)

1.7 DoD System Acquisition Program

Modern military systems result from extraordinarily complex processes involving a number of iterative steps, usually over many years. First, the capabilities to be provided or the requirements to be satisfied are defined. Then, alternative concepts, both materiel and non-materiel solutions are considered to provide the capability, including maintenance and training, are developed and evaluated to compare capability performance, effectiveness, affordability, schedule, risk, and potential for growth. The evaluations may lead to refinements in the capabilities to be provided, further concept development, and, ultimately, the selection of a preferred concept to provide the capability. If the cost and risks are viewed as acceptable, an acquisition program may be initiated to complete development of the selected concept. The products that must be developed to implement the concept include not only the operational elements to provide the capability but also the equipment to train the operational personnel and to maintain and support the operational equipment over its life cycle. Equipment design and software development is followed by verification that developmental items meet their technical requirements and constraints. If successful, limited production of the equipment is typically followed by operational testing to validate that the operational and maintenance elements and associated instructions provide the needed or desired capability in the intended operating environments. If the system proves acceptable, production continues and is followed by deployment of the equipment to operational military units along with support equipment.

In most cases, there is no known synthesis approach that can accomplish the steps leading to acceptable system elements based on first principles. Instead, the steps must usually be accomplished iteratively, often a substantial number of times for some of the steps, before the system is ready for operations. Further, incremental military capabilities often evolve through evolutionary or spiral acquisition processes. Current technology is applied to develop the initial increment while the needed end-state operational capability or requirement may require further technology maturation. In such cases, the capabilities to be provided are defined for time-phased increments or spirals. Future capabilities are typically added to operational space systems by the development and deployment of new blocks of satellites to replace earlier versions that have failed or served their full operational lives.

1.7.1 DoD System Acquisition Processes

The DoD has three overarching and interactive management systems to implement the acquisition process. These are the JCIDS, The Defense Acquisition System (DAS), and the Planning, Programming, Budgeting, and Execution (PPBE). The DoD acquisition steps formally define the (i) needed capabilities and non-materiel solutions (JCIDS process, VCJCS/JROC Oversight as documented in CJCSI 5123.01 and CJCSI 3170.01), (ii) materiel solutions (DAS process, USD (AT&L) Oversight as documented in DoDD 5000.01 and DoDI 5000.02), and (iii) allocation of required resources (PPBE process, DepSecDef Oversight as documented in DoDD 7045.14 and DoDI 7045.7). All three of these processes are supported by SE activities that provide assessments of cost, schedule, and risk based on the evolving design of the system to provide the desired capability.

Many other Government processes or management systems support the acquisition of a new capability. The intelligence services provide information on the threat that could potentially act as a constraint on the operation of a system – the potential threat usually depends on the design of the system so that threat assessment is carried out interactively with SE. The meteorological or weather community provides data on the natural environment which may also constrain the system's operation – since the operational environment can depend on the design; this is another step that should usually be conducted interactively with SE. Finally, the operational test community validates that a system provides the needed capability[7] – any deficiencies that are discovered must be resolved by SE.

After DoD approval in the PPBE process, the President's budget is submitted to the Congress for the annual Authorization and Appropriation of public funds. Public funds as appropriated by the Congress are managed by the DoD Financial Management System[8] and are disbursed to the approved programs as appropriate.

1.7.2 Joint Capabilities Integration and Development System (JCIDS) Process

The JCIDS process operates in an iterative manner with a focus on non-materiel solutions to bridge identified gaps in force capability. Initial capability requirements drive the early acquisition process, and the early acquisition process drives updates to capability requirements related to specific materiel and/or non-materiel capability solutions to be pursued.

JCIDS addresses capability shortfalls, or gaps as defined by the commanders. It provides a capabilities-based approach to requirements generation. Since the requirements are generated at the force level the chance of developing superfluously overlapping systems is reduced. It also fosters awareness of capabilities under development leading to interoperable solutions. JCIDS process aims to provide a balanced system solution that combines doctrine, organization, training, materiel, leadership and education, personnel and facilities (DOTMLPF) considerations. Since capability needs are defined in consultation with the Office of the Secretary of Defense (OSD), there is greater ability to see and close capability gaps earlier in the acquisition process.

1.7.2.1 Capabilities-based Analysis

The new capabilities that are to be provided by a new system or the upgrade of an existing system can arise from a wide range of DoD activities. These activities generally fall in two broad categories. The first consists of opportunities created by the science and technology (S&T) developed by OSD and the military services – the Air Force S&T program is carried out by the AF Research Laboratory (AFRL) – and by academic, industrial, commercial, and international sources. Such situations are sometimes called

---

[7] See DoDD 5141.2, Director, Operational Test and Evaluation, AFPD 99-1, Test and Evaluation Process, and AFI 99-102, Operational Test And Evaluation, current editions.

[8] DoD Financial Management Regulation (FMR).

technology push or opportunities push. The second type of activity giving rise to new capabilities consists of operational problems or challenges that may be identified during training, exercises, operational testing, or military operations. Such capabilities are sometimes referred to as operational pull, or operational challenges. Either category can result in the identification of desired or needed capabilities (or requirements) through a wide range of planning activities: strategic, operational, budget, or capability planning.

As noted above, the operational capabilities to be provided by a military system are normally formalized by the JCIDS for defense acquisition programs.

The analyses may point to a new or improvements to an existing system. The required capabilities are documented in the Initial Capabilities Document (ICD) that describes the specific gap in capability that is to be filled and makes the case to establish the need for a materiel approach to resolve the gap. Subsequently, an Analysis of Alternatives (AoA) would provide the basis for choosing a specific concept and for the JCIDS to refine the capabilities to be provided in the Capability Development Document (CDD) to support the initiation of a formal acquisition program. Still later, the JCIDS prepares the Capability Production Document (CPD) that refines the capabilities to be provided by a production increment in an acquisition program.

1.7.3 The Defense Acquisition System (DAS) Process

The DoD 5000 series documents strongly encourage and promote the use of evolutionary acquisition strategies relying on an incremental development process. Evolutionary acquisition and spiral development are methods that reduce cycle time, speed the delivery of advanced capability to the warfighter, and provide opportunities to insert new technology and capabilities over time.

The DAS process dovetails the JCIDS when early SE indicates the need for a materiel solution is necessary to field the required capability. It is a four step process:

- **Materiel Solutions Analysis (MSA) –** following the Materiel Development Decision (MDD) in conjunction with the JCIDS process, materiel solutions are developed, assessed, and documented for Milestone A reviews.
- **Technology Development (TD)** – specific technologies as identified in the MSA phase are developed as necessary for incorporation into the engineering design and manufacturing. A decision to proceed is made at Milestone B.
- **Engineering and Manufacturing Development(EMD) and Production and Deployment (PD)** – Engineering solutions for the required capability are architected, developed, and designed for eventual production after Milestone C reviews
- **Operations and Support (O&S) –** The built system is fielded, supported, and sustained for the duration of its useful life and them disposed appropriately.

These DAS phases are iterative and evolutionary. The TD and EMD phases can be repeated for on-going system increments for fielding improved capability to support stakeholder needs.

1.7.4 The Planning, Programming, Budgeting, and Execution (PPBE) Process

DoDD 7045.7 states “The purpose of the [Planning, Programming, Budgeting System] PPBS is to produce a plan, a program, and, finally, a budget for the Department of Defense. The budget is forwarded in summary to the President for his approval. The President's budget then is submitted to the Congress for authorization and appropriation.”

The focus of planning is to (i) define military strategy to maintain national security, (ii) plan an integrated and balanced approach to force structure to accomplish that strategy, and (iii) provide decision options to formulate national security policy and help with issuance of the defense guidance. Programming entails DoD components developing programs to implement defense guidance. DoD components then develop and submit detailed budgets for the proposed programs which are reviewed and the results are issued in the Program Budget Decisions (PBD). The execution review occurs simultaneously with the program and budget reviews where feedback is sought concerning the effectiveness of current and prior resource allocations.

Current planning, programming and budgeting procedures support a two-year cycle that results in two-year budgets as described in Management Initiative Decision (MID) 913, dated May 22, 2003.

1.7.5 Interaction Between JCIDS, DAS, and PPBE Processes

A validated ICD through the JCIDS process and approval from the Milestone Decision Authority (MDA) is required to conduct an Analysis of Alternatives (AoA) and enter the Materiel Solution Analysis (MSA) Phase in the DAS or 5000.02 process when a materiel solution is necessary to produce the desired capability. The interaction between JCIDS and DAS continues as a CDD is developed after entering the TD phase and is completed and approved by the MDA before Milestone B. A CPD is produced during the EMD phase and approved by the MDA before the Milestone C.

The Deputy Secretary of Defense manages PPBE as the primary process for enabling the funding of the various JCIDS and DAS activities which develop, field, and sustain effective capability solutions to the warfighters. JCIDS and other program documents along with The Office of Management and Budget (OMB) fiscal guidance are used to develop Program Objective Memoranda (POM). OSD submits the overall funding program to the OMB based on validated POM, department priorities established in JCIDS, and the annual Program and Budget Review (PBR) it conducts. OMB consolidates DoD funding requirements into the overall President’s Budget for submission to Congress. When funded, the SPOs execute their programs and continue to interact with the JCIDS and DAS processes with activities that include study, identification, and validation of new capability requirements and associated gaps and operations and sustainability of fielded capabilities.

For more details on the system lifecycle see Chapter 3.

1.8 Early Systems Engineering and Program Foundation

Most descriptions of SE are in terms of the process for carrying out the iterations and associated tradeoffs that result in a design for the system that can fulfill the capability needs of the users and other stakeholders. As shown in Section 1.6, there are several SE process models that can be adapted depending on the program complexity and need. If the system elements can be sufficiently compartmentalized, more than one model can be applied. The benefit is obvious when large software components can be separated and developed using the OOA/D model. Furthermore, for large and complex systems, the models can be applied recursively to its constituents.

The following description recounts typical early SE process for developing a complex military space system within the context of the 5000 series of DoD acquisition directives and instructions and instructions and manuals for the capability needs process issued by the Chairman of the Joint Chiefs of Staff (CJCS).

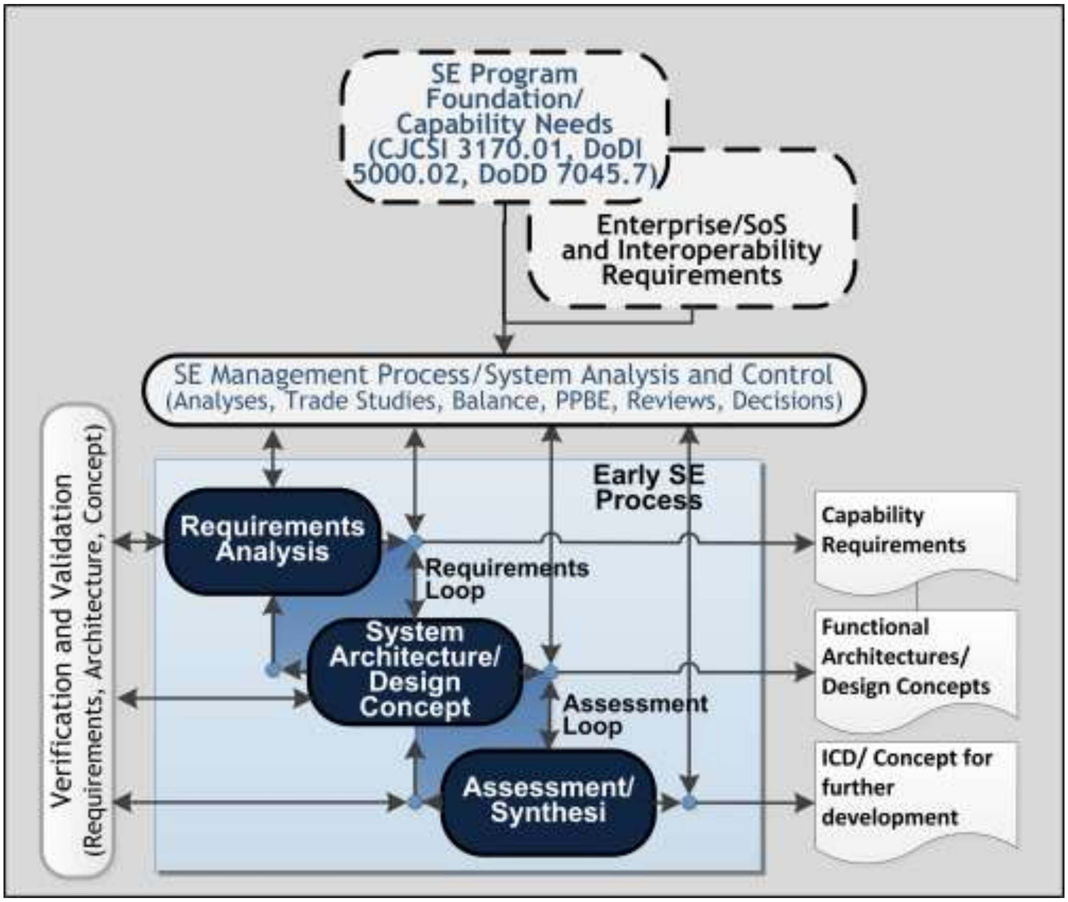


Figure 1-7 Early SE process diagram

In DoD programs, the early SE process starts with the iterative definition of the driving requirements and the architecture or design concept that responds to those requirements as the basis for further development.

A simplified early SE process, shown in Figure 1-7, begins with the identification of the needed capability and other related stakeholder issues that establishes the foundation for SE on a program. For a major program (one with high potential cost or high level of interest), the Joint Capabilities Integration and Development System (JCIDS) leads the development of the capability needs. In all programs, the Operators and Users establish the needed capability and have a significant role in the selection of the concept for further development. For example, the Operator/Users will have a major role in selecting between space and terrestrial concepts for providing a given capability. Also, as part of the foundation, the Operator/Users[9] may establish objectives or goals that indicate that increased capability beyond the minimum or threshold need would be militarily useful if it can be affordably and feasibly provided without significantly delaying the introduction of the needed capability. The range between the thresholds and objectives creates a SE trade space where effectiveness, cost, schedule, risk, and growth potential of alternative design concepts is assessed and compared to select architectures for further development that are balanced with respect to those factors.

---

[9] In many systems, the operator and user are the same military operational command. In the case of many space systems, however, the operator may be one of the service space commands such as Air Force Space Command while the users may be one to all of the other service or unified commands. Both the operator and the users may have needs or desires that help establish the program foundation.

The identification of the need may flow from operational experience or from the opportunities created by new technologies – either way; the technology base is a fundamental limiting factor in developing the architecture or design concept in response to the need. As a result, a thorough understanding of the applicable technologies and their level of maturity is essential in defining and evaluating the architecture or design concept.

The program foundation includes considerably more than the needs identified by the Operator/Users. For example, constraints such as (i) external interfaces required for interoperability with the DoD enterprise, (ii) CJCS mandated joint CONOPS for sharing and reuse in as SoS environment, (iii) the storage, transportation, and operating environments (terrestrial or space) such as temperature, electromagnetic, and the threat imposed by known or potential enemy capabilities also limit the range of practical design concepts. Note that the precise character of these constraints may depend on the proposed solution. As an example, one set of capabilities might lead to a design concept that might in turn result in a satellite weight within the capability of one launch system (such as the Delta II) and its interface constraints while a more demanding capabilities might lead to a satellite requiring a more capable launch system (Atlas V or Delta IV) having a different physical interface and launch environment. The range of potential threats is also likely to depend on the design solution.

Also, policy and public law (legal constraints) involving factors such as (i) environmental impact, (ii) safety hazards, and (iii) information assurance are important to understanding which concepts will be useful and practical. When a system is being acquired to replace an existing system, the plan for transitioning from the current system may place additional constraints on the concept and program (such as the schedule for the current system to be retired).

1.8.1 Requirements Analysis

The purpose of requirements analysis is to convert the program foundation into system technical or engineering requirements that can be related to the characteristics or attributes of the architecture or design concept and of the technologies needed to implement it. As the process unfolds, the system technical requirements must also be allocable to the elements that make up the system concept and also be verifiable so that compliance of the system design can be confirmed. Completeness and accuracy are also necessary to prevent costly and time-consuming changes late in the development process. One way to help achieve completeness and accuracy is to analyze the needed capability and the constraints in the context of the concept of operations and the characteristics of the operational environment. Based on the results of the analysis, one systematic way to state the capability is by defining the tasks or functions that the system must perform, i.e., by defining the functional requirements. This is followed by specification of performance requirements that state how well each function must be performed. Completeness with respect to the constraints can be achieved only by integrating specialists from each conceivably affected area into the requirements analysis and concept definition. The requirements are then validated to demonstrate that they completely and accurately reflect the needed capabilities and constraints. In part, this is done through a review by those who defined the needed capabilities as well as those who specialize in each of the factors that constrain the system and program.

As the concept (and subsequently, the design) is defined and assessed as discussed in the following paragraphs, further requirements and constraints may be derived that are dependent on the characteristics of the conceptual solution. The requirements analyses leading to the derived requirements and constraints are critical to ensure that the system achieves, for example, (i) electromagnetic compatibility in the operational and test environments, (ii) meets spectrum allocations and constraints, (iii) integrates human factors, (iv) effects safe use and controls any associated hazards, (v) eliminates or controls the vulnerabilities to security threats, (vi) is in accordance with DoD and service regulations and public law, and (vii) is reliable, maintainable, survivable, producible, transportable, and verifiable over the life cycle of the system. Such concept or design dependent requirements and constraints are one reason that iteration is an important part of any SE process.

1.8.2 Architecture and Design Concept

There is no analytic approach for defining a system architecture or design concept that is responsive to the technical requirements and constraints. In fact, a number of architecture or design concepts are possible to provide the needed capability. The SE process at this juncture considers all plausible design concepts based on (i) available technology and its maturity, (ii) existing enterprise and other constraints, and (iii) parameters such as effectiveness, cost, schedule, risk, and evolutionary potential. The concepts can range from an upgrade or evolutionary growth for an existing system to a new system; from those based on terrestrial platforms to those based on space platforms; and, for space-based concepts, to approaches ranging from a small number of large satellites to a large number of small satellites or from low altitude satellites to high altitude satellites.

The design concept for a space system can be arranged into space, terrestrial control, and user elements. Each of those can then be described in terms of signal flows (as in a communications system or the sensor elements of a surveillance system) or information processing (as in the information classification such as threat vs. non-threat, storage, and retrieval elements of a surveillance system). The signal or information flow can be organized into elements that correspond to the characteristics of applicable key technologies that might be used to implement the concept and the available engineering design teams.

1.8.3 Assessment and Synthesis

When one or more architecture or design concepts have been proposed, they must be assessed. The assessment starts with an evaluation of effectiveness, cost, schedule, risk, and potential for evolutionary growth. The effectiveness of a system is a quantitative measure of the degree to which the system's purpose is achieved, i.e., the degree to which the technical requirements and constraints are met or, if met, the margin relative to the threshold requirements and essential constraints. For example, the system effectiveness for a launch vehicle includes the mass that can be injected into a specified orbit and launch availability. For a given concept, injection mass and availability may tend to move in opposite directions so that as the injection mass for a given orbit is assessed to increase, the predicted availability may decrease giving rise to the need to assess the balance between the two parameters (and perhaps many others).

Effectiveness may initially be assessed via analysis such as calculation of a link budget for a communications subsystem based on key characteristics of the power amplifier and antenna concepts or determining the capacity of a given communications protocol standard in relation to a particular interoperability requirement. As the concept is refined, the assessment may be based on a simulation of the concept and its operating environment. If breadboards or prototypes are available, then the simulation may grow in fidelity to include hardware in the loop. The assessment of effectiveness (or performance) will eventually be based on verification data for the integrated system. However assessed, the expected effectiveness must be compared with the technical requirements and constraints to assess the feasibility of the concept to satisfy the need.

The predicted or estimated costs should be compared with affordability goals or constraints. The cost of a system is the value of the resources needed for development, production, and operations and support over its life cycle (which total to the life cycle cost). Since resources come in many forms such as contractor personnel, materials, energy, the use of facilities and equipment such as wind tunnels, factories, tooling, offices, computers, and military personnel, it is usually convenient to express the values in monetary units (dollars). Resources are scarce, i.e., dollars applied to one system will not be available to provide some other capability by another system – that's the reason decision makers sometimes impose constraints on part or all of the cost of a system.

Cost cannot be estimated or assessed based on first principles. Only extrapolating historical experience can assess cost. For example, the development of a system can be broken down into a set of tasks. The cost for each task can then be assessed or estimated based on the cost for similar tasks in the past. Alternatively, the cost can also be assessed based on key attributes of the system concept or design. As an example, the cost to develop software might be estimated based on an estimate of the historical cost per line of code for a similar type of software. A key point is that cost cannot be assessed based on the capability to be provided or on the technical requirements and constraints. Rather, the cost must be estimated based on the historical costs associated either with the tasks to develop, produce, and operate a particular system design over its life cycle or with key characteristics of the concept or design for which historical cost data is available.

In addition, the predicted schedule to develop, produce, and deploy the system should be compared with the need date (which can be particularly critical when a new or upgraded space system is to replace an existing space system before the end of the useful life of the satellites for the existing system). Like cost, schedule can be predicted or estimated based only on historical experience for to carry out similar tasks or develop similar designs.

The development and production of new capability is accompanied by risks and uncertainties that the work will cost more, that the schedule will take longer, or that the required effectiveness will not be achieved. As an example, the effectiveness may depend on an evolving technology that has not been previously applied to a military system. The resulting potential outcomes (such as the parts applying the new technology may not be available on schedule) and the consequences of each outcome (such as the cost and delays of continued development) must be assessed and, if judged to be unacceptable, then mitigation steps must be put in place (such as the selection of a different technology or the parallel development of a backup part using a different technology) which may have both an immediate impact on the cost and schedule (such as the cost and time to develop the backup part) as well as the potential for still further impacts (such as the costs and delays to integrate the backup part and the associated reduction in effectiveness).

Finally, to assess the provisions for evolutionary growth, such characteristics as the degree of openness of the architecture as well as provisions or potential for growth in weight, volume, and power must be assessed and their adequacy judged in relation to the objectives of the program.

1.8.4 Verification and Validation

Verification and validation is integral to a well-executed SE technical process. Within early SE process, requirements are verified for technical feasibility and are traced to and validated against stakeholders' capability needs. The process is repeated for architecture and design concepts, and also during the assessment and synthesis activities. V&V assures that the concept(s) for further development are realizable, balanced, and meet stakeholders' needs.

1.8.5 SE Management Process: System Analysis and Control

SE management process runs parallel to the SE technical process. Its function is to review and assess technical status of the program, balance effectiveness or capability needs against fiscal constraints, and provide direction for further development to minimize mission risk.

DoD 5000 series of acquisition directives and instructions provide the program decision makers substantial flexibility depending on the maturity of the technology and other risk factors and the urgency of the military need. As bound by the DAS, JCIDS, and the PPBE processes, the task of SE management process is to contain and drive the technical process as shown in Figure 1-7 toward a balanced concept that can be further developed or conclude and recommend that there is no plausible solution to meet the requested capability needs at this time.

Management process incorporates various reviews at the appropriate level to assess needed capabilities, technology maturation, and concept selection. Other management decisions that can be made based on SE products include those for budget levels, risk management, readiness for operational test, readiness for launch, readiness for production, system support and sustainment strategies. Thus, the program decision makers are among the customers of the SE assessments and other products. It can be useful to view SE as a staff function to the Government and Contractor program managers for such tasks as (i) requirements analysis, definition, and allocation, (ii) system status assessment, and (iii) risk management.

1.8.5.1 Balance

The assessments of balance are important to decisions at several points in the process. First, the balance between effectiveness, cost, and the other factors can usefully inform the work of the Operator/Users leading to a statement of capability needs that can be affordably and feasibly satisfied – this is indicated by the feedback to the capability needs through the SE Management Process as shown in Figure 1-7. Subsequently, balance is important in the selection of the concept or design parameters in the trade space between the technical requirements corresponding to the threshold needed capability and the objectives. Balance is also important to the selection of design margins to ensure that the needed capability is achieved in the final delivered system. Such design margins apply to the difference between the technical requirements and the predictions of effectiveness for a given design concept or design approach. Other margins that are important and must be balanced apply to the difference between the predictions of worst case environments and the technical constraints imposed on and subsequently met by the design. The penalty for inadequate margins can be severe, e.g., the loss of a billion dollar satellite if the margin between, say, the launch vibration environment and the satellite design's ability to survive that environment is inadequate.

If the requirements are demanding, it is unlikely that an initial proposed concept will meet all the technical requirements and constraints without excessive cost, schedule, or risks or inadequate potential for growth. Based on what is learned from an initial assessment, additional iterations can be formed to trade off alternative statements of the requirements (in the range between the thresholds and objectives), alternative concepts, or alternative design parameters for a given concept. Iteration can be confined to a single step or, as the feedback arrows in Figure 1-7 suggest, it can involve two steps or all of the steps.

Once a number of such iterations have been completed, the assessments can be reviewed to identify the concept(s) that provide the highest effectiveness and potential for evolutionary growth while avoiding excessive cost, risk, or schedule implications. The cost and effectiveness data formed by a series of iterations through the early SE process can be summarized as shown Figure 1-8 for easy reference.

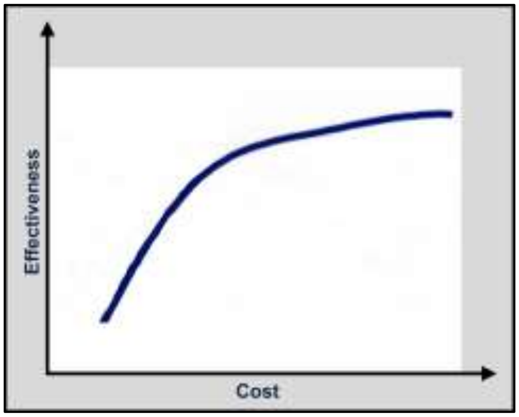


Figure 1-8 Cost-effectiveness curve

The design concept and the parameters of a specific design concept vary along the curve to produce the assessments of effectiveness and cost, i.e., the curve is an envelope of the best effectiveness that is achievable for a given cost with the available technology. At the lower left, the predicted effectiveness increases linearly with cost, but at the upper right the cost increases dramatically with little increase in effectiveness. Up to a point, the design concept with the best effectiveness is likely to be preferred as long as the cost is affordable. Since cost cannot be predicted with certainty, the top part of the curve represents high cost risk with little potential for increased effectiveness. One strategy that program decision makers select in light of such a curve would be to establish a cost "requirement" or "design-to-cost" constraint near where the curve bends over, i.e., at the "knee of the curve."

The assessments may show that some aspect of the needed capability is not achievable at low risk or that the cost may be unaffordable or the schedule (to, say, mature a needed technology) unresponsive to the need. For example, if the effectiveness in the area where the curve in Figure 1-8 bends over is below the desired capability tentatively established by the Operator/Users, then they may wish to establish a lower initial increment of desired capability level or reconsider whether a material solution is the best approach at the current time given the available technology. As noted above, such feedback to the Operator/Users is provided through the SE Management Process as shown in Figure 1-7. One outcome could be the decision for further technology development to achieve risk reduction before formally starting a new program.

1.8.6 Concept for Further Development

If the architecture or design concept is judged to be responsive to the needed capability and balanced with respect to cost, schedule, risk, and evolutionary growth potential, then the program decision makers may select it as a basis for further development to fulfill the need.

1.9 SE Process: System Development and Design

When the concept is approved for continued development, the simplified process in Figure 1-7 is usually expanded along the lines shown in Figure 1-9. A summary of the SE process with reference to the various baselines as shown in the Figure 1-9 is provided here. Refer to Chapter 2 for a detailed and technical discussion of the overarching SE process.

As in Figure 1-7, iteration can involve a single step, two or more steps, or all of the steps. In comparison to Figure 1-7, however, several steps have been re-titled somewhat to show their more general application over the next iterations of the process. In addition, steps have been added that experience has shown to be valuable in many situations. The first focuses on functional analysis and allocation, and the second focuses on the analysis and allocation of the technical requirements for the products that make up the design. The step of defining the architecture or design concept is incorporated into the latter step as it provides the framework for allocation. Finally, the overarching step that was labeled "balance" in Figure 1-7 has been expanded into SE management that includes not only balancing but also the challenging task of adapting the process to achieve and maintain balance over the life cycle of a system which often spans many decades and contractual phases to include modifications and upgrades after the start of operational use.

The objective of the process in Figure 1-9 is a series of baselines that define the requirements for the system and the design in increasing levels of detail. These baselines are primary products of the SE process. It can be helpful to maintain a clear distinction between the products of SE and products that are defined by the design process and constitute the elements of the system.

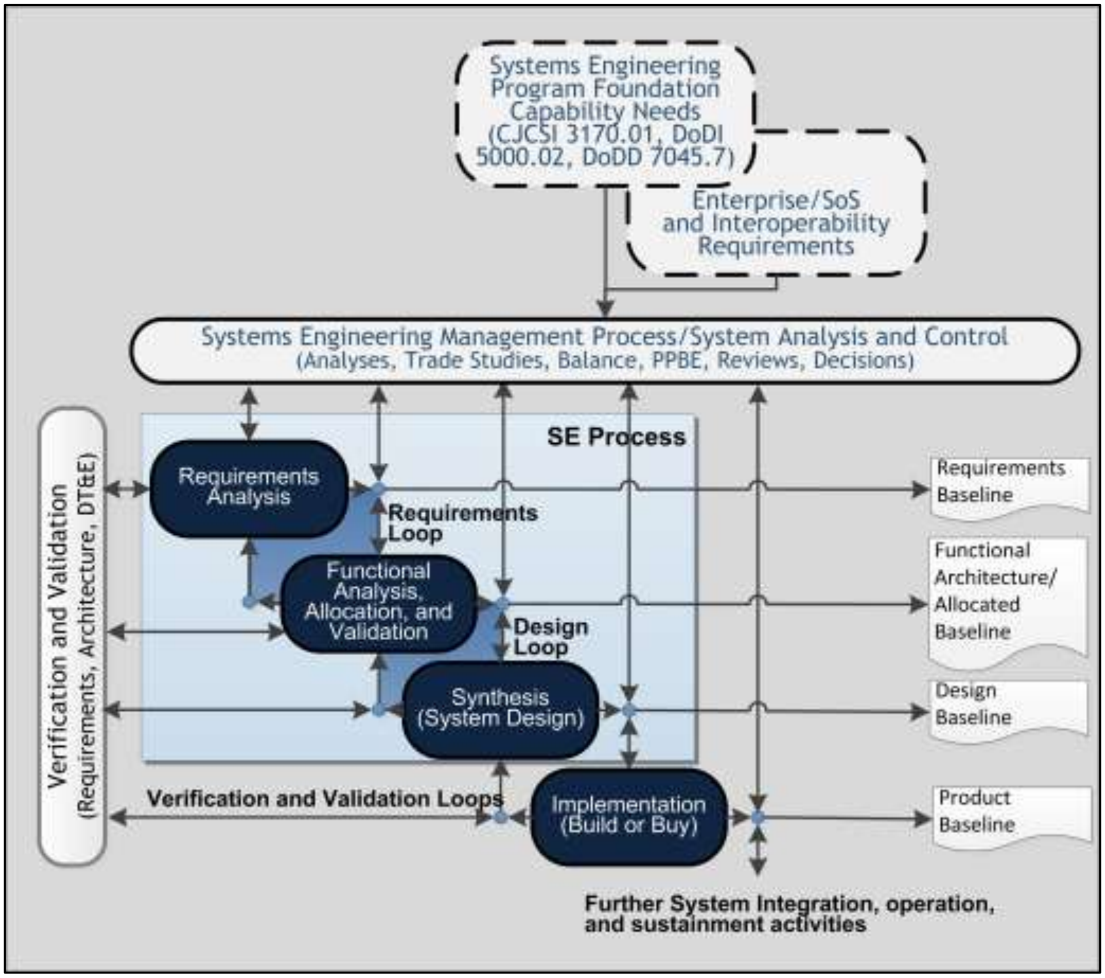


Figure 1-9 SE process diagram

1.9.1 Requirements Baseline

The first of the baselines is called the requirements baseline. It is simply the system technical functional and performance requirements and constraints described above under requirements analysis after they have matured as the result of several iterations of the process and been validated to capture the needed capability and the system and program constraints. In some definitions of the SE process, the requirements baseline also includes the allocation of the system level requirements to the major elements of the system (sometimes called the system segments). For a space system, one segment might be formed by the elements in space, another by the ground control elements, and a third by the user equipment. The term functional baseline is sometimes used in lieu of requirements baseline.

1.9.2 Functional Architecture and Allocated Baseline

This step iteratively decomposes and allocates requirements into sub functions to the point that they can be unambiguously related to subsystems and lower system elements or products that make up the design. The result is often called the functional architecture. A common starting point to defining the functional requirements and hence the functional architecture is the eight primary lifecycle functions that all systems must satisfy: development, verification, production (and construction), training, deployment (or fielding), operations, support, and disposal.[10]

A number of functional decomposition and allocation techniques like the FFBD, IDEF0, and NxN diagrams can be applied as discussed in detail in Chapter 5 on tools. An FFBD is used here to illustrate the functional architecture development process. Figure 8 shows the eight primary lifecycle functions organized into a simple, generic top-tier FFBD. It shows the order in which the functions must be carried out to provide the capability.

To develop the functional architecture, the top tier FFBD in Figure 1-10 is refined to apply specifically to the needed capability and constraints that are driving the development. This step may also lead to refinements in the requirements baseline.

Next, each of the primary functions is further decomposed until the associated technical requirements can be directly associated with and allocated to physical products that make up the system. This process is called functional analysis and allocation. The Functional Analysis section of Chapter 2 provides an overview of functional analysis approaches.

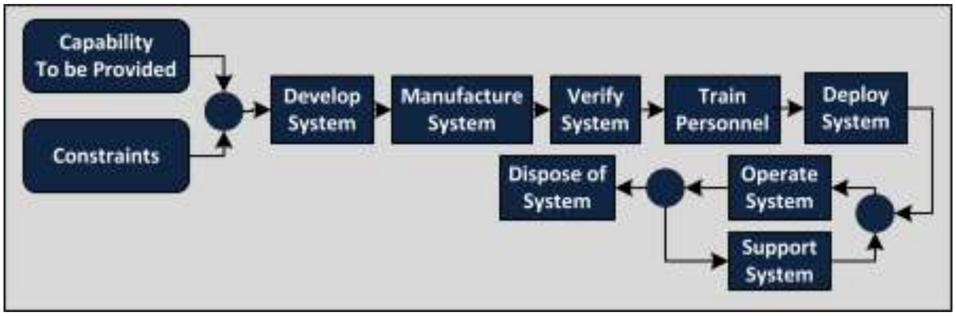


Figure 1-10 A top level FFBD showing a simple, generic relationship among the eight primary life cycle functions

For simplicity, this primer focuses on functional analysis to logically relate the system technical requirements to the technical requirements for each of the elements or products that make up the system. Some textbooks and standards recognize other approaches such as (i) OOA/D, (ii) SysML™, (iii) physical hierarchy, (iv) structured analysis, and (v) information engineering analysis for developing a logical system solution representation.[11] No matter what technique is applied, the objective is to ensure that all the system level requirements are identified and that each is allocated to one or more of the products that make up the system in a way that unambiguously communicates the tasks to be completed to the design engineers. Any methodology that can be shown to accomplish that objective is acceptable.

1.9.2.1 Extent of the Functional Analysis and Decomposition

Any form of logical analysis is tedious and demanding in resources. It is likely to pay off when applied to those requirements that lead to complex or unprecedented solutions. In situations where the road between requirements and solution is well traveled, most of the benefits may be achieved by curtailing the decomposition at a higher level with subsequent allocations down the physical hierarchy or physical architecture discussed below.

---

[10] Not all systems engineers agree that it is useful to include development as a system function. Others argue that it is helpful to extend the formalism of functional analysis and allocation to planning the development program. This is just one example of how systems engineering processes vary in practice.

[11] See ANSI/EIA-632-1998, Processes for Engineering a System, see Requirement 17 and the following discussion on page 23.

1.9.2.2 Product Requirements Analysis and Allocation

An important objective of the SE process in Figure 1-9 is to identify the requirements for each element or product in the system which is to be designed by a separate design team, separately manufactured and verified, procured from a subcontractor, or separately specified for any other reason. To allocate the system technical requirements and constraints to these physical products, a representation or framework is needed that identifies them. The starting point for developing such a representation is, of course, the system architecture or design concept discussed above. One such representation is the physical hierarchy.

1.9.2.3 The Physical Hierarchy/Physical Architecture

One physical representation that has proven useful for refining the system architecture or design concept is the hierarchical relationship among the elements that make up the system. This representation is often called the physical hierarchy, physical architecture, or product tree. A simple example for a space system is shown in Figure 1-11.

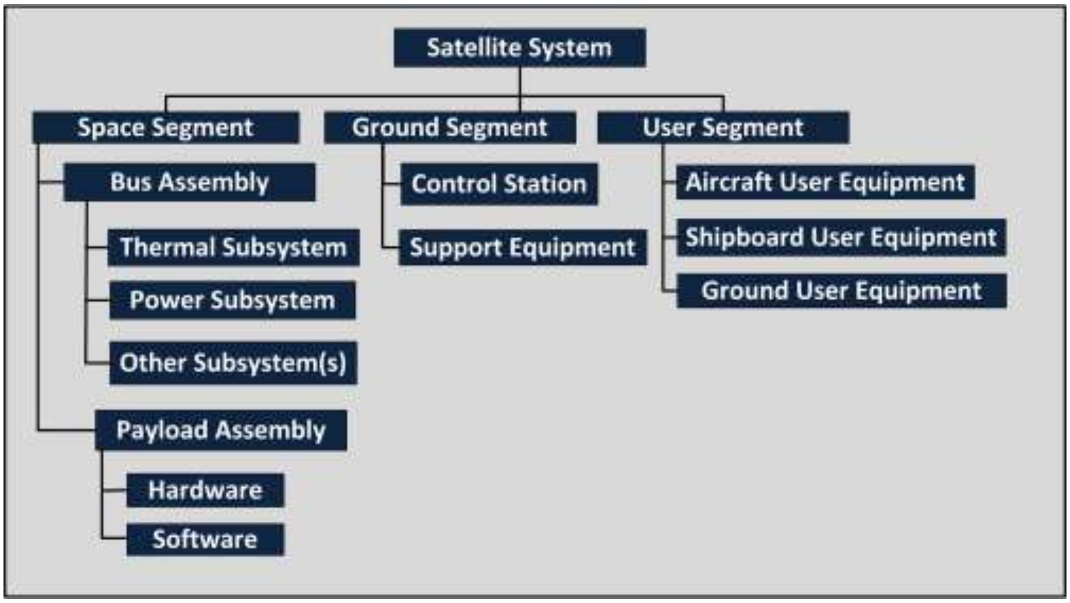


Figure 1-11 A simple satellite system physical hierarchy (product tree)

The physical hierarchy can be a powerful tool for organizing many of the tradeoffs that form the iterations depicted in Figure 1-7. For example, the projected life cycle cost of each element in the tree can be used to focus on the elements that most affect or drive the cost – one often-used heuristic rule is to focus on the elements that account for 80% of the cost. Or risk analyses linked to the tree can help focus risk mitigation and reduction steps on those elements judged to drive the risk. As a result of the tradeoffs, risk mitigation steps, and other development steps, the system product tree evolves in detail and is refined as the system design is iteratively developed. To complete the allocated baseline, the physical hierarchy is extended to the point that each system element is identified that is either to be designed by a different organizational element or that will be manufactured, procured, coded, inventoried, or supported as a separate element over the life cycle.

Many of the terms in Figure 1-9 are often used in describing or referring to various levels in a system. For example, the level below the system is made up of segments. (As noted above, the requirements baseline may define the technical requirements for the system and segment levels.) In the simple example in Figure 1-11, there is less nomenclature commonality at the next level as assembly, station, and equipment are all used. The space segment assembly level is made up of subsystems while components are among the descriptions that might be used at the next level. Instead of component, the term unit is sometimes used to describe the lowest level in the physical hierarchy.[12]

Note that software is included in the example product tree wherever it is installed in a next higher level product. However, some programs have used trees that are a mix of products and functional specialties. In such trees, software may be collected and shown at a higher level. In programs that use the product trees to assign Responsibility, Authority, and Accountability (RAA) for the performance of a complete element (such as the processor subsystem shown in the example), software should be placed as shown in the example so that the element can be fully verified to meet its allocated requirements. Also, placing software where it logically falls in the system facilitates helps with logical allocation of requirements down the product tree to each element.

The physical hierarchy can be easily extended to provide other helpful tools. For example, by adding identifiers for the corresponding requirements documents or specifications for each system element, the product tree becomes a specification tree. The documents defining interface constraints, whether contained in interface specifications or interface control documents, can be added to the specification tree to link it to the interface constraints. The physical hierarchy also provides a roadmap for integration of the system elements to form the system. As we will see below, many programs use the product tree as the starting point for a product-oriented structure to break down and plan the work necessary to develop and produce the system and then to monitor the progress.

1.9.3 The Allocated Baseline

The functional analysis and allocation discussed above provides the basis for allocating the system technical requirements and constraints to the elements of the physical hierarchy. The resulting set of technical requirements for each element starts to form the allocated baseline. In addition to those allocated directly from the system level, the allocated baseline should also include derived requirements such as interface constraints between the elements of the physical hierarchy or those to provide for expected dependability. In the design process discussed next, these requirements and constraints will lead to the selection and control of parts, materials, and processes and their organization for a balanced design. When complete, the allocated baseline defines all design-to requirements for each design team or subcontractor that is responsible for the design and development of each component or other element at the bottom of the physical hierarchy. Above the bottom level, it also defines the requirements for integrating the components to form higher level assemblies in the physical hierarchy. Arguably, the allocated baseline is one of the most important contributions of a structured SE process because it helps (a) ensure that the resulting system is balanced even though the design may be carried out by teams working at different locations or with different engineering orientations (such as sensing, communications, computation, propulsion, or structures) and (b) minimize the problems encountered as the components are subsequently integrated up the physical hierarchy.

1.9.4 The Design Baseline

Design baseline represents build-to system design and process specifications. Stated simply, design is the process of selecting and organizing the parts, materials, and processes and determining the associated personnel manpower and skill levels necessary to comply with the requirements in the requirements and allocated baselines. For hardware, the results of the design process include drawings, parts lists, and assembly and process instructions. For software, the design includes descriptions such as flow diagrams that define inputs, actions, outputs, and response time constraints. It can be useful to think of the design as an

---

[12] Alternatively, the term unit may refer to a specific copy of a system element such as in the unit under test. As with other terms and phrases used in systems engineering, the meaning varies in practice and must be determined by context.

extension of the physical hierarchy or product tree described above. In the case of hardware, that extension, in part, defines the selected parts, materials, and processes and their organization to provide both the capability need established by the Operator/Users as well as the reliability and other requirements and constraints defined in the iterative SE process to balance the system design. The documented design forms the product baseline.

1.9.5 The Product Baseline

Product baseline represents as-built system design and process specifications. Product configuration documentation is the detailed design documentation including those verifications necessary for accepting product deliveries that includes first article and acceptance inspections. For complex space systems, full design and process disclosure is expected for the finished product.

When the design is complete, several steps still remain before the capability is realized in operational use. These steps are shown in Figure 1-9 Design baseline provides the basis for manufacturing, buying, coding, and subsequent integration of the products that make up the system. For control or user equipment that is to be integrated into other platforms such as aircraft, it also includes the design of the hardware and software necessary for integration and the steps to complete the integration. Each completed or integrated product is then verified to comply with its requirements in the allocated and design baselines, and the system is subsequently verified to comply with the requirements baseline. The design baseline should also include the personnel manpower and skill levels required to operate, maintain, and sustain each of the products and the integrated system. Several steps still remain before the needed capability is available to the operational forces.

For one, the acquisition program must usually transition from development to production. For some elements such as large satellites and ground control elements, that change may primarily involve the details of the budgeting and financial management processes. For elements of the system to be built in quantity (such as the User Equipment for some systems); however, the production may involve new or additional tooling and other steps to achieve an efficient manufacturing process.

Furthermore, the Operator/Users must validate that the system provides the needed capability in an operational-like environment and that the projections for manpower and skill levels are adequate and necessary – the Operator/Users perform initial operational test and evaluation (IOT&E). IOT&E may be carried out on the initial satellite and deployed control hardware. For hardware that is planned to go into rate production, IOT&E is usually accomplished after development has been completed and the initial production hardware is available – such hardware is sometimes called Low-Rate Initial Production (LRIP). The validation step addresses not only the primary operational equipment but also the means to support and sustain that equipment. It includes such factors as field and depot maintenance equipment, documented workflows and procedures, and the availability of spares for replaceable elements that fail in the satellites prior to launch and in the terrestrial control and user equipment.

Eventually, the system must be deployed, first in sufficient quantities for IOT&E and later to complete the planned deployment. For satellites and launch systems, this includes transportation to the launch site, the physical and functional integration of the satellite and launch system, and launch. For the satellites, it includes on-orbit checkout to verify the required operation. For control and user elements, deployment includes transportation, assembly and installation at the operational sites or in the operational platform, if needed, and checkout to verify that the elements are operating properly. Once checkout is complete, the verified satellite or other equipment is turned over to the Operator/Users for IOT&E or operational use.

When production hardware and final software code are available and have been verified and validated to meet all requirements, the actual products may be compared with the design baseline documentation to arrive at the product configuration baseline shown in Figure 1-9. This configuration is maintained until modifications or upgrades are required for any reason. A strict iterative SE process is applied to document changes in the product baseline necessary for upgrades and other modifications.

As an element of the system or the system as a whole reaches the end of its useful life, means must be provided to responsibly dispose of it. For terrestrial elements, this can include such steps as rendering elements safe that could otherwise harm people or the environment or salvaging any remaining value. For satellites or launch systems, this can include either reentry such that humans or property are not endangered or moving the hardware to an orbit that will not interfere with future space operations (usually by raising the object to an orbit that will not soon decay into an occupied orbit).

In summary, the steps above mean that the requirements, allocated, and design baselines should be comprised of the following:

- Build-to, buy-to, or code-to requirements (instructions) for each component,
- Integrate-to requirements to assemble the components into intermediate assemblies and then the system,
- Deploy-to requirements for each separately deployable assembly,
- Verify-to requirements for each component, factory-integrated assembly, deployed assembly, and the system,
- Operate-to requirements (such as technical orders or TOs) to operate the system,
- Train-to requirements to train personnel to operate, maintain, and sustain the system,
- Support/sustain-to requirements to maintain operational status for the system,
- Dispose-to requirements to dispose of a system element or the system as a whole at the end of its operational life, and
- Personnel manpower and skill levels to operate and sustain the system.

When each component and assembly and the system as a whole has been verified to meet all requirements in the baselines and the production hardware has been compared with its requirements in the baselines, the product configuration baseline may be approved to guide continued production and serve as the point of departure for any future changes or evolutionary upgrades.

As the baseline shown in Figure 1-9 evolves, experience has shown that specialized management activity traditionally called configuration management or configuration control is beneficial to ensure that the baseline is fully documented and maintained in the face of inevitable system changes to overcome deficiencies or accommodate modifications or upgrades. To achieve this, the baseline is placed under configuration control. As subsequent changes are proposed, the baseline is maintained so that it forms the basis both for future manufacturing, procurement, and coding to initially field the system and to subsequently support and sustain it during its life cycle to include modifications and upgrades that prove necessary or desirable. For more discussion of configuration management, see system analysis and control in Chapter 2 and configuration management in Chapter 4.

1.9.5.1 Decision Database

To guide each iteration and tradeoff aimed at achieving the initial baselines and then to determine the potential impacts and benefits of changes that are subsequently proposed, experience has shown that it is helpful to maintain a record of the basis for each decision that is made in developing and maintaining each baseline. Such a record is called a decision data base. Usually, the decision data base is implemented via a computer application by which each decision is electronically linked to both its bases and the resulting element(s) in one or more of the baselines. A decision data base typically contains:

- The system engineering program foundation
- Each of the system baselines and the functional architecture (or other logical representation).
- Iteration/tradeoff results including assessments of cost, schedule, risk, and evolutionary growth potential and analytic techniques applied
- The chronology of decisions and implementing actions
- History of changes including approval authority and rationale

The decision data base should provide for efficient traceability through the baselines and functional architecture (a) from any element up to the Government sources for the requirements baseline or down to the lowest elements of each baseline; (b) from any requirement to its corresponding bases (in higher level requirements and/or tradeoff or other analyses), validation, verification method, and verification plans, procedures, and results; and (c) from any element to its change history.

1.9.5.2 Technical Reviews and Audits

To provide the opportunity for all the program stakeholders to develop in-depth insight into the direction and progress being taken by the contractor(s) toward providing the needed capability, technical reviews and audits have traditionally been held along the lines summarized in Table 1-1 taken form SMC Standard, SMC-S-021, “Technical Reviews and Audits for Systems, Equipment and Computer Software,” volume 1. (SMC-S-021 may be consulted for more details on each of the reviews including entrance and exit criteria.)

For programs, some of the reviews may not be necessary or the purposes of two of the reviews may be merged into one review. Furthermore, at times alternative names may be used for one or more of the reviews or the content of a given review may vary. Usually, the final objective in the above table is supported by a range of intermediate objectives or topics that are addressed in the review so that the final objective is achieved as part of the close out of the review.

The typical or nominal objective of each of the baselines required at each of the reviews and audits is summarized in the table. The actual requirements may vary from one program to another or even from one evolutionary phase to another

Finally, in some programs, the structure may differ from the nominal DoDI 5000.02 phases shown in the table, and as such the reviews and audits may be held in phases different from those shown in the table.

As used in the table below, a preliminary baseline is one that documents the results of initial iterations through a process similar to the one summarized in Figure 1-9. A draft baseline may be reviewed for accuracy and completeness to identify the work necessary to approve it at the next review or audit. Once a baseline is approved, the contract may require that subsequent changes be reviewed in accordance with formal configuration control procedures. After the requirements baseline is approved, Government approval is usually required for changes. In some programs, the Government may also retain control of the baselines for other selected products or even all delivered products. In some programs, the allocated baseline may become part of the design baseline and not be separately maintained once the latter is approved – in other programs, it may be separately maintained to guide the application of the iterative engineering process for modifications, evolutionary upgrades or improvements.

Table 1-1 Technical review and audit objective

Technical Review	Objective	DoDI 5000.02Phase
Analysis of Alternatives (AoA)	Concept Selection, System CONOPS	Material Solution Analysis
System Requirements Review (SRR)	Review SE Program Foundation and Approval of the Initial Requirements Baseline	Technology Development
System Functional Review (SFR)	Review and Approval of the System Architecture and Functional Requirements Baseline	Technology Development
System Requirements and Architecture Review (SAR)	Review and Approval of the Software Architecture and Functional Requirements Baseline	Technology Development
Preliminary Design Review (PDR)	Approval of the Allocated Baseline	Engineering and Manufacturing Development
Critical Design Review (CDR)	Approval of the Design Baseline	Engineering and Manufacturing Development
Test Readiness Review (TRR)	Verification of the Contractor's Readiness to Begin a Formal Verification Testing	Engineering and Manufacturing Development
Functional Configuration Audit (FCA)	Qualification of the Design	Engineering and Manufacturing Development, Production and Deployment
Physical Configuration Audit (PCA)	Approval of the Product Configuration Baseline	Engineering and Manufacturing Development, Production and Deployment
Manufacturing Readiness Review (MRR)	Readiness for Production, Training, Deployment, Ops, Support, and Disposal	Engineering and Manufacturing Development, Production, Deployment, Operations and Support
Production Readiness Review (PRR)	Authorize Follow-On Procurement of Additional System EIs Complete Initial small Quantity Large Quantity Production-Centric Procurement	Engineering and Manufacturing Development, Production, Deployment, Operations and Support

1.9.6 Verification and Validation

Verification and Validation are important to the designer, the systems engineer, the program manager, and the customer. Verification confirms that the system element meets the design-to or build-to specifications as documented in appropriate system baselines, i.e., it tells if the right system is built. Validation tells if the system performs as expected.

The V&V efforts provide direct evidence of progress towards ultimately meeting the customer's requirements. V&V results obtained over the development, design, production, and integration processes provide incremental assurance that the product will pass the customer's criteria. Eventually these results provide proof that the product performs as specified, and provide an indication of how well the product satisfies operational needs. A description of V&V methods, processes, and procedures is provided in Chapter 7.

1.9.7 SE Management Process

SE management executes the traditional management tasks of planning, organizing, staffing, directing, monitoring, and controlling to systematically achieve a design that meets the system technical requirements and constraints and, at the same time, balances effectiveness, cost, schedule, risk, and evolutionary growth potential. For example, the iterations and tradeoffs discussed earlier are planned and directed. Based on the results, control actions are taken to plan and direct the continued development, often including additional iterations and tradeoffs. The primary tool SMC SPOs use for SE Management is the Systems Engineering Plan (SEP). For detailed discussion on the SE management process, see Chapter 4.

1.10 Software Systems Engineering

Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software[13]. The software lifecycle and the development models are generally akin to the overall SE lifecycle and models as outlined earlier in this primer. The difference is largely qualitative: software resides on hardware and is mainly concerned with providing a logical functionality to complement system capability.

"The quality of the processes involved in developing and acquiring software and systems has a significant effect on the quality of the resulting products. Public and private organizations have reported significant returns on investment through improvements to these processes. For example, the Software Engineering Institute (SEI) reported in 2006 that a major defense contractor implemented a process improvement program [with significant savings cost, schedule, and system defects][14]." SMC has developed an extensive body of knowledge to guide acquisition of software which has increasingly become a major factor in the success of its programs. This guidance includes:

- **SMCI 63-103, Software Acquisition Process Improvement Instruction –** this instruction outlines the process to comply with the requirements of the Air Force Software Acquisition Process Improvement Strategy (SWAPI). It serves as a guide to standardize the Air Force Space Command (AFSPC) and Air Force Materiel Command (AFMC) roles and responsibilities.
- **SMCI 63-104, Software Acquisition Instruction –** this instruction establishes the process, roles, and responsibilities regarding software acquisition that provide a new, improved, or continuing system or service capability in response to an approved need at Air Force Space Command's Space and Missile Systems Center (SMC). It serves as a method to standardize all software acquisitions at SMC.
- **SMCI 63-108, Software Acquisition Management Plan (SWAMP) –** this instruction provides the requirements for the preparation and approval of a Software Acquisition Management Plan (SWAMP) in association with the Software Acquisition Instruction 63-104, along with this document and Section 804 of the Bob Stump National Defense Authorization Act of 2003 for improving software acquisition processes at SMC. The SMCI 63-104 requires that each program within each Directorate prepare a SWAMP to document the software acquisition activities.
- **SMC-S-012, Software Development for Space Systems –** the purpose of this standard is to establish uniform requirements for software development activities. This standard applies to the development of systems that contain software (such as hardware-software systems), software-only systems, and stand-alone software products.
- **SMC-S-21, Technical Reviews and Audits for Systems, Equipment and Computer Software, Appendix C –** software Requirement and Architecture Review (SAR) is a formal, multidisciplinary

---

[13] Software Engineering Book of Knowledge (SWEBOK), Editors, Pierre Bourque, Robert Dupuis, IEEE Computer Society, 2004.

[14] GAO Report to Congress, GAO-09-888, Information Technology: "DoD Needs to Strengthen Management of Its Statutorily Mandated Software and System Process Improvement Efforts, September 2009.

review of the software requirements, architecture, and test planning technical products, software development processes, and current state of the software development.

1.11 System Engineer's Tools

We use tools to aid us to perform essential tasks or generate products. In this section we briefly discuss those tools that are peculiar to our space SE development environment. Typically we select and employ tools based on our assessment of the activities and tasks to be accomplished and products required for delivery. For the SMC Program Office environment, we might consider activities by program phase then associate the tools that would be needed or beneficial to use. For example, during the concept definition phase, we are very interested in modeling the missions that our system is going to support. We develop mission models and simulations to be able to run case scenarios in both threat environments and non-hostile operating environments. During the concept phase we also commence with concept and operational architecture definition. We use architecture tools to define the architectures and an assortment of modeling and analyses tools to assess and down-select the best conceptual choices.

Obviously, maintaining and upgrading a smaller suite of tools is preferable to a larger suite. Hence much thought and consideration must go into selecting the right set of tools. If tools are selected that have similar functions and databases, there may be the need to transfer data between the tools. Extensive and sometimes costly training requirements might be associated with some tools. Many specialized tools demand expert users that must also have an in-depth knowledge of their discipline to adequately use the tool. There are also many other things to consider such as system requirements (processing speeds, memory, operating systems, etc.), licensing, maintenance, peripheral software and hardware requirements, etc. Any tool selection assessment should also consider 'lessons learned' on other SMC projects.

Table 1-2 provides a summary of basic SE tools. These and other specific tools are described in detail in Chapter 5.

1.12 Specialty Engineering Disciplines

A number of specialty engineering disciplines, supporting a system throughout its lifecycle, are indispensable and integral to SE process. Chapter 7 briefly introduces the specialty engineering disciplines (SED) that are crucial to the success of SE for complex systems like those in the SMC portfolio of space related programs. Based on system and program requirements, the SPOs implement approved and documented processes to perform specialty engineering and support SMC's goals for OSS&E and Mission Assurance. SMC has developed a companion volume to this book that provides a standard approach and framework to specialty engineering disciplines (SED).

Table 1-2 Summary of SE tools and activities

Activity / Tools	Concept Development	Functional Analysis	Performance Analysis	Requirements Analysis	Architecture Development	Functional Allocation	Design Analysis	Implementation	Integration	Operations and Sustainment	Software Development	Verification and Validation	Program Management
AoA and Case Scenarios	X		X										
Audits and Reviews	X	X	X	X	X	X	X	X	X	X	X		X
Change Management	X	X											X
CMMI®											X	X	
Cost Analysis Tools													X
DoDAF Viewpoints	X	X			X	X							
Earned Value Assessment													
FFBD	X	X			X	X							
IDEF0	X	X			X	X							
Modeling and Simulation (M&S)	X	X	X	X	X	X	X				X	X	
NxN Diagrams	X	X	X	X	X	X	X						
Prototyping	X	X	X				X				X	X	
Requirements Analysis Sheet (RAS)		X	X	X			X					X	
Risk Analysis										X			X
SysML™	X	X	X	X	X	X	X						
Test and Evaluation Tools												X	
Trade Studies	X	X	X	X			X						
UML											X		
Value Stream Analysis	X	X	X	X	X	X	X	X	X	X	X		X
WBS and Physical Architecture		X			X	X	X	X	X				X

2 Systems Engineering Process

2.1 Introduction

Chapter 1 described basics of SE and related concepts with special reference to space and military systems. In this chapter, we expand on key constituents of the process such as requirements analysis, functional analysis and allocation, synthesis, and system analysis and control to further illustrate how it works.

The SE process is a series of repetitive operations whereby a set of possible solutions are narrowed to a single system design that optimally satisfies the perceived need. It is a continual excursion between the general and the specific, always striving toward a possible cost-effective implementation. Even the most talented SE team cannot initially identify the optimum solution with certainty. "What worked before" is the obvious starting point, but if existing systems met all the requirements, there would be no need for a new system. In fact, with the present emphasis on evolutionary design under DoDD 5000.1, one of the most important questions the systems engineer should ask is, "Can these requirements be satisfied using existing or slightly modified systems?" If the answer is yes, the customer's needs may be met much sooner and at lower cost.

2.2 Early Systems Engineering Process: JCIDS Capabilities-Based Approach

JCIDS implements a CBA approach that better leverages the expertise and assets of all government agencies, industry, and academia to identify capability shortfalls or the need for new capabilities that can be bridged by improvements to existing systems or by the development of new ones.[15] The JCIDS process plays a key role to identify, develop, and validate operational requirements that deliver improved or new capability to the warfighter.

The JCIDS process recognizes the necessity to view and identify capability shortfalls and new opportunities at the highest strategic level as modern systems have stakeholders that typically include organizations or services external to or unanticipated by the system sponsor(s). This is especially true for the space systems: GPS users, for example, extend beyond the military to the civil agencies, allied nations, and commercial space. The need for sharing, reuse, and interoperable systems in the restricted budgetary environment is well-served by the JCIDS CBA process. This early SE activity requires a collaborative process that utilizes joint concepts and integrated architectures to identify prioritized capability gaps and integrated Doctrine, Organization, Training, Materiel, Leadership and Education, Personnel and Facilities (DOTMLPF) solutions (materiel and non-materiel) to resolve those gaps.[16]

As described in Chapter 1, the JCIDS process is integrated with the DAS and PPBE processes. This ensures consistent decision making at the DoD enterprise level that delivers timely and cost effective capability solutions to the users, operators, and other stakeholders.

One of the major aspects of the CBA approach that is important to the early SE is the JCIDS emphasis on joint capability areas. JCIDS is organized and empowered at the DoD enterprise level to identify commonality of needs across the entire force where cost-effective solutions can be shared or reused. JCIDS identifies and provides a common vocabulary for the joint capability areas. These include battle-space awareness, command and control, logistics, and net-centric operations.[17] Joint capability areas and their

---

[15] CJCSI 3170.01, Joint Capabilities Integration and Development System (JCIDS), and JCIDS Manual (current)

[16] For more details see USAF's Early systems Engineering Guidebook, 2008.

[17] A detailed list of joint capability area attributes is provided in the JCIDS Manual

well-defined attributes naturally lead to (i) a joint understanding of enterprise level capability gaps and shortfalls, and (ii) a joint concept of operations. A validated and approved joint concept of operations is the right place to start constructing an SoS by recruiting and updating existing systems or building new ones. It is also the right place to identify joint capability requirements and develop optimal solutions to mitigate shortfalls in force interoperability.

JCIDS Manual states, "A CBA begins by identifying the mission or military problem to be assessed, the concepts to be examined, the timeframe in which the problem is being assessed, and the scope of the assessment. A CBA determines the relevant concepts, CONOPS, and objectives, and lists the related effects to be achieved. A CBA may also lead to policy development or support and validation of existing policies."[18] Joint capability requirements are based on the CBA, which ultimately serves the high-level strategy and guidance in the National Security Strategy, National Defense Strategy, National Military Strategy (NMS), Quadrennial Defense Review, Guidance for the Employment of the Force, and the Defense Planning Guidance (DPG). The CBA contains (i) analysis of warfighter needs across all functional areas to accomplish the mission, (ii) analysis of existing or planned systems to identifying gaps and redundancies, and (iii) recommendations on how the shortfalls can be optimally mitigated. Factors contributing to the CBA that leads to new or improved capability requirements include:[19]

- **Strategic direction –** as part of the national security, defense, or military strategy a CBA can be initiated and a sponsor identified to validate and field an operational capability for the warfighter
- **Combatant Commander Needs –** capability shortfalls can be identified in the field that may result in a CBA
- **Joint operations concepts –** changes or improvements in the Joint operations concepts can point to capability gaps or emergent needs. It can also help identify gaps in force interoperability and point to new SoS constructs to meet that challenge.
- **Concept of operations –** an approved Concept of Operations (CONOPS) identifies the problem being addressed, the mission, the commander's intent, the operational viewpoint, the objectives, and the roles and responsibilities of the organizations. Changes or improvements or alternate views in the concept of operations can point to capability gaps or emergent needs.
- **Development and capability plans –** The more traditional and systems level methods to identify capability gaps or emergent needs are based on development and capability planning. This type of activity includes Concept Explorations and Refinement (CER), Preferred System Concept (PSC), Technology Development (TD), lessons learned, and other studies. This aspect of early SE and related documents such as the Concept Characterization and Technical Description (CCTD) that support the CBA process are discussed in detail in the USAF's Early Systems Engineering Guide.

This CBA process to develop operational capability requirements is summarized inFigure 2-1. A typical CBA provides an in depth analysis of perceived capability gaps based on risks and actionable recommendations. JCIDS Manual tabulates risk factors to assess impact of capability gaps as reproduced in Figure 2-1 to include ability to achieve the strategic objectives, operational timelines; resources, unanticipated requirements, force provider resourcing and component functions, force management, and institutional capacity.

Since the space systems are typically high cost and cannot be refurbished or repaired in space, they require additional rigor in CBAs and in the validation of resulting capability requirements. Also note that the JCIDS provides a fast track for urgent or emergent operational needs where the scope of assessment can be reduced appropriately.

---

[18] JCIDS Manual, Appendix B, Enclosure A

[19] Based on AFI 10-601, Operational Capability Requirements Development

Figure 2-1 CBA process to develop operational capability requirements

Table 2-1 Risk factors to assess impact of capability

Risk Criteria	Low	Moderate	Significant	High
Strategic Objectives	Near certain achievement	Very likely achievement	Likely achievement	Significant risk of failure
Operational Timelines	As planned	Minor extension	Significant delay	Delays with significant risk of failure
Resources	As planned	Requires resources from other plans or operations	Requires resources that create significant shortfalls	Requires resources that preclude other plans or operations
Unanticipated Requirements	Easily managed, minimal impact	Managed via minor adjustments to other plans	Managed via significant adjustments to other plans	Cannot manage
Force Provider Resourcing	Fully capacity to source requirements	Sourcing requires limited duration capability gaps	Sourcing requires extended duration capability gaps	Requires full mobilization to cover capability gaps
Institutional Capacity	Fully capacity to source requirements	Requires shifts within DOD components to meet requirements	Requires shifts among DOD components to meet requirements	Requirements exceed capacity of the Joint force

The results of the CBA are reported in one of the following documents:

- Joint DOTMLPF Change Recommendation (DCR)
- Joint Urgent Operational Need (JUON) or Joint Emergent Operational Need (JEON)
- Initial Capabilities Document (ICD) – this leads to AoA after the decision is made to proceed with the MSA phase for a materiel solution in the DAS process, and later to the Capability Development Document (CDD) at Milestone B, and Capability Production Document (CPD) at Milestone C.

2.2.1 Joint DOTMLPF Change Recommendation (DCR)

A joint DOTMLPF Change Recommendation (DCR) is produced when a non-materiel solution is recommended to mitigate identified capability gap with no further action through the JCIDS process.

2.2.2 Joint Urgent Operational Need (JUON) or Joint Emergent Operational Need (JEON)

For urgent or emergent operational needs an expedited assessment, review, and approval process is followed to minimize delay and rapid fielding of the capability. It documents capability requirements driven by ongoing or anticipated contingency operations, which if left unfulfilled, would result in capability gaps leading to unacceptable loss of life or critical mission failure. As appropriate, a JUON, JEON, or a DoD component UON is produced. In an iterative process, the documents are updated over the urgent or emergent capability need solution lifecycle. Unless withdrawn by the sponsor, JEON or JUON documents are reviewed and re-validated after a period of two years to make sure that the emergent or urgent capability requirements are still operative.

2.2.3 Initial Capabilities Document (ICD)

The ICD and Information Systems (IS) ICD makes the case to establish the need for a materiel approach to resolve a specific capability gap derived from the CBA process. It documents the capability gap, gap analyses and associated risks, and the derived operational capability requirement(s). Typically requirements are based on identified measures of effectiveness (MOE). It recommends how to partially or wholly address the gap by a materiel solution.

The ICD supports the analysis of alternatives (AoA) and the Materiel Development Decision (MDD) as shown in Figure 2-2. It also supports the MSA phase activities, the eventual Milestone A acquisition decision, and subsequent Technology Development phase activities. Once approved, it normally is not updated.

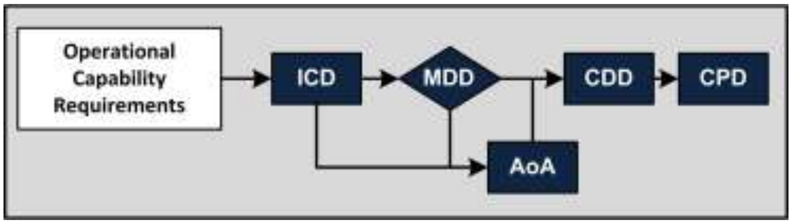


Figure 2-2 JCIDS process flow

2.2.4 Capability Development Document (CDD)

Guided by the ICD, the AoA (for ACAT I/IA programs), and technology development activities, the CDD captures the information necessary to develop proposed system(s), normally using an evolutionary acquisition strategy. The CDD outlines an affordable increment of capability. An increment is a militarily useful and supportable operational capability that can be effectively developed, produced or acquired, deployed, and sustained. Each increment of capability has its own set of attributes and associated performance values with thresholds and objectives established by the sponsor with input from the users. The CDD supports the Milestone B acquisition decision.

The CDD provides authoritative, measurable, and testable operational performance attributes, including producibility and supportability, necessary for the acquisition community to design the proposed system(s), including key performance parameters (KPP) and Key System Attributes (KSA) that guide the development, demonstration, and testing of the current increment. Because the operational performance attributes provided in a CDD apply only to a single increment of a program's development, the KPPs apply only to the current increment (or to the entire program when only a single increment is required to achieve full capability). The AoA should be reviewed for its relevance for each program increment requiring a Milestone B decision and, if necessary, the AoA should be updated or a new one initiated.

In addition to describing the current increment, the CDD will outline the overall strategy to develop the full or complete capability. For evolutionary acquisition programs, the CDD will outline the increments delivered to date (if any), the current increment and future increments (if any) of the acquisition program to deliver the full operational capability.

2.2.5 Capability Production Document (CPD)

A CPD provides authoritative, testable capability requirements and performance attributes to include KPPs and KSAs to enter the P&D phase of an acquisition program. The CPD addresses the production attributes and quantities specific to a single increment of an acquisition program. The sponsor finalizes a CPD after critical design review when projected capabilities of the increment in development have been specified. The CPD must be validated and approved before the Milestone C decision review.

Performance and supportability attributes in the CPD are specific to the increment. The threshold and objective performance values of the CDD are superseded by the specific production values detailed in the CPD for the increment. Reduction in threshold KPP performance at this stage requires an assessment of the military utility of the reduced capability and, possibly, a reexamination of the program to determine if an alternative materiel or non-materiel solution should be adopted.

2.3 Systems Engineering Process

Several SE process models were introduced in Chapter 1. Table 2-2 describes strengths and weaknesses of various models. Choice of a specific model or models and how they are tailored depend on program needs and complexity. However, it is important that the chosen models are comprehensive, fully documented, and applied rigorously over the entire lifecycle of the system, even more so when the systems are complex and must interoperate with other existing, planned, or future systems to meet the needs of joint forces.

All SE models, by necessity, are iterative in nature. For less defined system concepts, the more iterations it takes to translate amorphous capability needs into verifiable requirements, interoperable architectures and interfaces, and feasible design that can be produced, operated dependably, and sustained within cost and schedule constraints.

Few large systems are built from scratch. Programs like the GPS or MILSATCOM employ an evolutionary model, providing basic capability initially and incrementally improving upon it as new technologies and methods become feasible. Such systems rely on historical data and lessons learned. The task is complex as these systems intra-operate with several increments or blocks at different stages of their lifecycle, and at the same time interoperate with other enterprise systems by presenting standard and open interface like the IS-GPS-200 or those required by the net-centric KPPs.

An SE process based on Military Standard 499B is shown schematically in Figure 2-3. This representation is the most commonly used on DoD programs. The reader may refer to the SE process, Figure 1-9 of Chapter 1, to correlate the evolution of technical baselines with the constituents of this model.

Essential elements of SE process of concept development, requirements generation and allocation, architecture and interoperable interface development, verifiable design, integration, design verification and validation, and deployment are all represented in this model, used here to illustrate the SE process. These techniques, methods, and activities can be adapted for V or other SE models discussed earlier with minor modifications and change in emphasis that may be more suitable, given the nature of the system. For example, the V-model emphasizes top-down decomposition early in the process followed by rigorous verification and validation at every step-up on integration. The OOA/D model, more suitable for SW development, emphasizes rapid iterative and evolutionary development based on logical UML diagrams that help encapsulate functionality in interoperating objects.

Table 2-2 Comparison of SE process models

Model	Pros	Cons
Waterfall	This model is focused on the project. The traditional sequential waterfall development allows for departmentalization and managerial control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process like an assembly line, and theoretically, be delivered on time.	It allows for little reflection or revision. While the need for iteration is recognized, its application is usually ad hoc. Errors and omissions are harder to fix, especially those emanating from the concept stage. However, this model can be modified and applied to a highly iterative and recursive SE approach.
V-model	The model is focused on the project. Verification and validation is integral to this sequential development process. Every stage is tested.	The requirements and design loops are weakly implemented. The organization and execution of operation, sustainment, and disposal of the system are not covered by the V-Model. There is little room to modify concepts and requirements once developed. The first testing is done after the design is set, which makes integration difficulties harder and more expensive to rectify.
Spiral	Estimates are more realistic as work progresses, since important issues are handled earlier. Engineering resources can be deployed quickly for prototyping.	Highly customized, limiting re-usability. Applied differently for each application. Risk of not meeting budget or schedule.
OOA&D	Mainly used for software intensive projects, OOA/D offers agile (faster) development, code reusability, modular architecture, and a better mapping to the problem domain.	OOA/D does not replace traditional design and engineering processes, especially for large and complex (space) programs.
Modeling and Simulation	M&S techniques when good data is available offer (i) experimentation is faster and cheaper, (ii) reduced analytical requirements, and (iii) easily demonstrated models	M&S limitations include (i) accuracy of results strongly depends on data fidelity, (ii) may not provide adequate answers to complex questions, and (iii) cannot solve problems by itself, i.e., models may not be realizable with current technology.
Evolutionary	Evolutionary SE process acknowledges that not all requirements are understood or can be produced with current proven technology. System is built to satisfy some part of the need, leaving the implementation of full capability to later increments. User gets deploy part capability quickly.	Requires a strong understanding of end-state objectives to minimize interoperability and configuration management problems that can arise between legacy and modern parts of the evolved system.

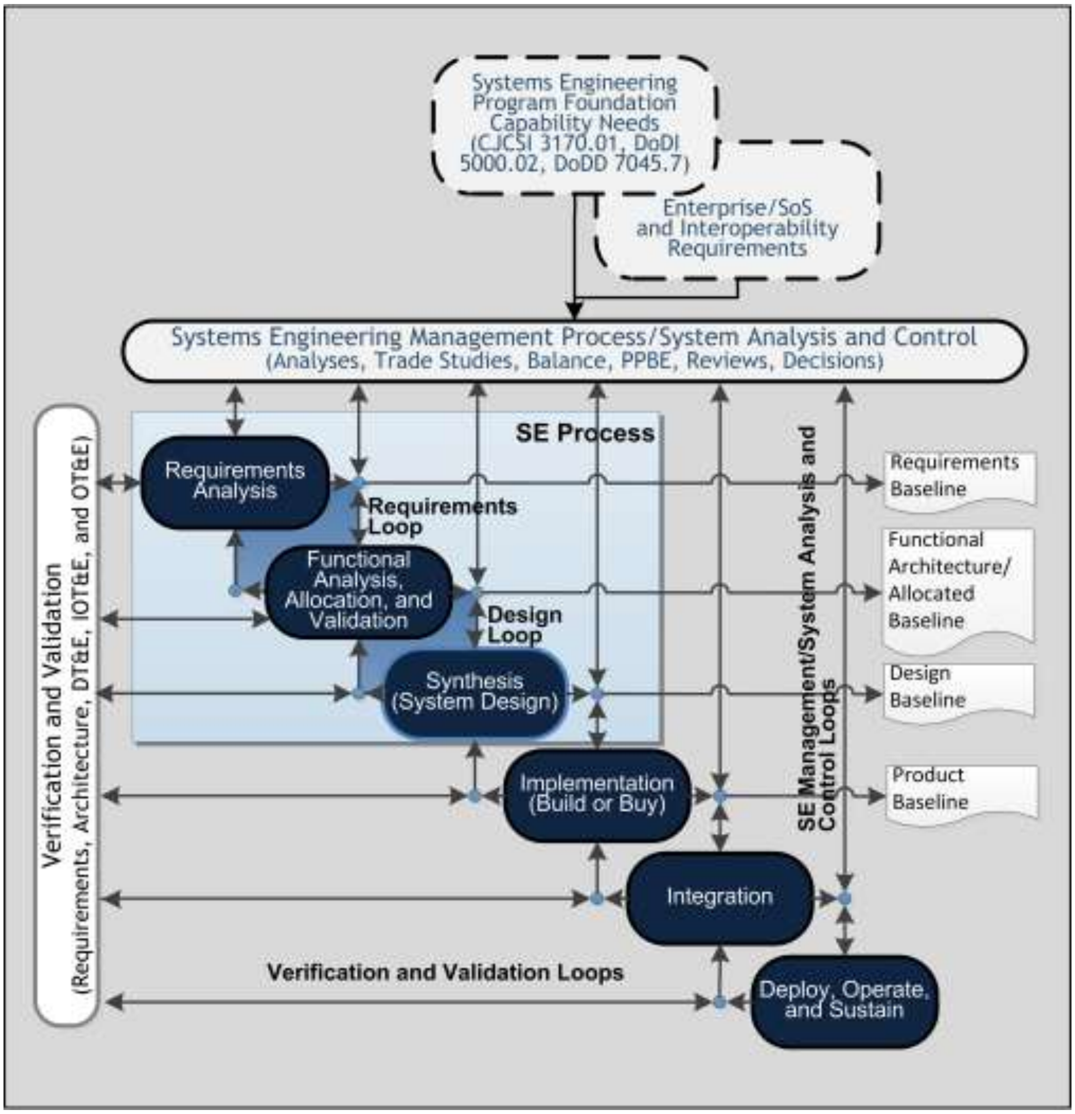


Figure 2-3 SE process based on traditional DoD model

The stakeholder's needs, objectives, and requirements in terms of capabilities, MOEs, MOPs, environments, KPPs, KSAs, and system constraints initiate the process. Each increment of capability is provided with its own set of attributes and associated performance values. MOEs quantify the results to be obtained and may be expressed as probabilities that the system will perform as required, e.g., the chance that a certain event will be recognized with a certain probability and that the probability of false alarm is below a certain percent. Environments refer to natural operating and threat environments, space, airborne, and ground segments. Internal environments, e.g., whether a particular system solution requires air conditioning or cryogenic cooling, are for the Systems Engineer to specify; it is of no consequence to the customer if the solution falls within the overall constraints and requirements. Enterprise- and Customer-imposed constraints may take the form of design for cyber security, net-ready implementation, interoperability with existing or other planned systems, operations and maintenance personnel skill level requirements, and costs and schedules.

Note that the SE process is event-driven, that is, it is concerned only with how activities flow from one to another, in what order activities are accomplished, what predecessor tasks are required as prerequisites, and what subsequent activities are affected. DoD Instruction (DODI) 5000.02 provides acquisition models used in developing DoD systems. Later, we relate these models to the SE functions of documentation, baselining, and review/audit, and to the requirements documents driving these functions. In addition, program decision points are intended to impose interim checks on the practicality and progress of the program. These decision points may occur with formal multiple milestone reviews, readiness reviews, or contractually required technical reviews and audits.

The CBA, technology base, and prior development efforts are natural inputs to the process. The basis should be built upon good, solid SE. Furthermore, SE is built upon what has been done before. However, in analyzing existing technology for use on the current program, the systems engineer is expected to identify critical areas and provide measurable evidence that use of new technology in a given application may be appropriate or beneficial. This may also indicate the need for additional research.

The major constituents of the traditional SE Process are requirements analysis, functional analysis and allocation, synthesis, and system analysis and control. There is continual interaction and feedback among these activities and refinement of their outputs as the program progresses.

The initial interaction is through the requirements loop. The results of the mission and environments analysis and the identification of functional requirements are the input to the decomposition to lower level functions and the allocation of the requirements to the lower functions. As these analyses and allocations are accomplished, the results are used in the requirements analysis to verify their compliance. This feedback is used to determine whether modification of the requirements is compatible with achieving the mission.

The design loop operates in parallel with the requirements loop. Functional interfaces are established and functional architectures defined so that physical system configurations can be developed. As concepts are transformed to hardware and software, the design characteristics are analyzed against the allocated requirements. Functional architectures and allocations are re-examined and modified if necessary. Some results of the design loop may even reflect into the requirements analysis necessitating further re-evaluation.

The final feedback “loop” is the verification of the emerging detailed design against the originating requirements. This may be accomplished by analysis, simulation, demonstration, proof testing of critical components, or a combination of these. Note that verification can be interpreted as a loop or a process, and different authors have treated it different ways. For this handbook, verification is considered to be a process, but there are certainly iterative aspects to the process that have the characteristics of a loop. What matters is that verification is accomplished thoroughly and correctly.

The SE management process (or system analysis and control activity) functions as the planner, manager, judge, traffic cop and secretary of the process. This activity identifies the work to be performed and develops schedules and costs estimates for the effort. It coordinates the other activities and assures that all are operating from the same set of agreements and design iteration. It evaluates the outputs of the other activities and conducts independent studies to determine which of the alternate approaches is best suited to the application. It determines when results of one activity require the action of another activity and directs the action to be performed. It documents the results of analyses and studies, maintains control of the evolving configuration, and measures and reports progress.

The output of the SE Process is a documented decision database and a balanced system solution that includes:

- The design,
- All the decisions made to arrive at the design,
- Defining specifications,
- Verification requirements, and
- Traceability of design features to imposed requirements, constraints, specifications and standards.

The balanced system solution is the best fit to all the final requirements and criteria imposed.

In the remainder of this chapter, a more detailed discussion is provided on various activities of the SE process with specific reference to the Figure 2-3. Sub-activities are identified to aid the discussion and to highlight specific efforts, attributes, and outputs.

2.3.1 SE Process Inputs

JCIDS directed CBA and the operational need requirements derived from the analyses form a major part of the process inputs. The CBA encompasses analysis of warfighter needs across all functional areas to accomplish the mission and the capability gaps that exist in meeting those objectives. The capabilities requirements for the proposed solution are documented in the ICD during early SE activities. Joint capability areas impose constraints (requirements) for commonality to achieve interoperable force structure such as the net-centric KPPs for information sharing, reuse, and assurance. Joint CONOPS from a higher vantage point view how existing, planned, and future systems may interoperate as an SoS entity resulting in superior execution of the mission, and thereby levy interoperable interface requirements.

Development planning is another substantive source of process inputs. It offers on-going concept exploration, technology development, and lessons learned databases relevant to the perceived capability shortfall or emergent need. Such studies are documented in CER, PSC, or TD reports. At times, they may be put together in a CCTD that could, in conjunction with the ICD, form the basis for a TRD.

Complex incremental and evolutionary systems like the GPS or MILSATCOM, which form the bulk of space systems acquisition activity, have historical data that is required for interoperability or can be applied to the current effort for significant savings in cost and schedule. This important set of process inputs from existing sources including prior increments includes:

- Legal and statutory constraints like survivability KPPs or safety requirements
- DoD Enterprise mandates like DISR, FISMA, and net-ready KPP
- Architectural and design constraints imposed by Joint concept of operations
- Interoperability requirements like the net-centric data strategy, MOEs, and MOPs
- CDD, CPD, and concept of operations
- AoA
- Acquisition decision memoranda (ADM)
- Prior increment program database to include system technical requirements, specifications, and standards
- Lists of government furnished equipment, property, and software
- Long term program objectives and philosophy
- OSS&E, space flight worthiness criteria, and Test-as-you-fly requirements data (See SMC Supplement to AFI 63-1201 and SMC Guides 120 through 1204.)
- Verification and validation data (IOT&E, FOT&E) from prior systems

2.3.2 Requirements Analysis and Validation

The CBA, the development planning documents, the joint and system concept of operations, and other foundational body of knowledge described in the previous paragraph forms the basis of requirements analysis. It is this process that transforms amorphous, sometimes conflicting, stakeholder wants into a consistent set of statements, called the requirements. The analysis documents complete, traceable, feasible, testable, necessary and sufficient, well-formed requirements that are defined to a level of detail needed to design the system. See Section 1.6.2 for characteristics of a good requirement and the types of requirements that are produced to define a buildable system that provides dependable stakeholder capability in its intended environment.

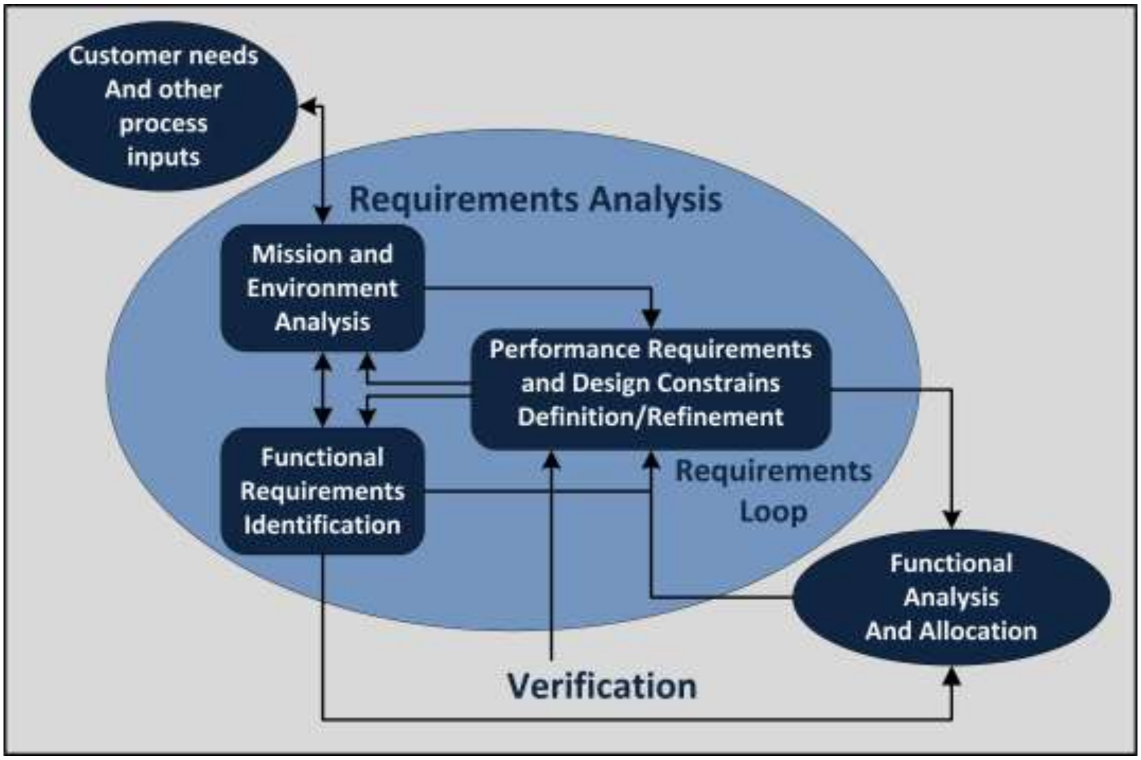


Figure 2-4 Requirement analysis–converting customer needs into system requirements

The Requirements Analysis is one of the first activities of the System Engineering Process and functions somewhat as an interface between the internal activities and the external sources providing inputs to the process. It examines, evaluates, and translates the external inputs into a set of functional and performance requirements that are the basis for the Functional Analysis and Allocation. It links with the Functional Analysis and Allocation to form the Requirements Loop of the System Engineering Process.

The activities of the Requirements Analysis are shown in Figure 2-4. The Missions and Environments Analysis firms the customer's needs and states them in terms that can be used to establish system functions, performance requirements and design constraints. The output of this activity initiates Functional Requirements Identification and the Performance/Design Requirements Definition and Refinement. As these activities progress, the original assumptions and conclusions are checked against evolving details. Usually this results in some modification of the original thinking, and may even reflect back to the customer's needs where certain ones may be impractical or excessively costly. The output of the Requirements Analysis is a set of top-level functional definitions and accompanying performance and design requirements that become the starting point of the Functional Analysis and Allocation. The Requirements Loop serves to refine the requirements and initiate re-evaluation to determine how firm the requirements are for items that prove to be major cost, schedule, performance or risk drivers. Later in the overall process, detailed system characteristics are compared against the established requirements to verify that they are being met. At this point there is usually little change to the requirements due to the verification feedback, but occasionally some minor changes are considered when the payoff is significant.

The Requirements Analysis and Validation activity culminates in a requirements baseline, the first of the several event-driven baselines that follow. As a snapshot in time, the requirements baseline provides:

- A two-way traceable summary of requirements as derived from documented capability needs
- A documented and configuration controlled system specification which is validated through stakeholder reviews
- Analyses of requirements to ensure that it is valid, necessary, current, and satisfies the higher-level capabilities, requirements, or constraints from which they resulted
- Analyses of lower level requirements as appropriate
- Documented decision trade studies that balance system effectiveness, life cycle cost, schedule, risk, and the potential for evolutionary growth.
- Documented method of verification for all requirements
- A traceable and iterative basis for functional analysis, allocation, and validation process to follow

A detailed description of the activities of the requirements analysis and validation process as shown Figure 2-4 is provided below.

2.3.2.1 Mission and Environment Analysis

The Systems Engineer helps the customer refine his needs, objectives, and measures of effectiveness in light of the initial and evolving results of the Requirements Loop. Questions such as, “What is the minimum/maximum operating time required to accomplish the mission?” and, “Are alternate existing capabilities available to provide backup?” are posed and answered. Needs that are design drivers are identified and characterized as desirable or mandatory. Constraints that limit solutions are identified and defined in detail, e.g., mission or utilization environments (extremes of heat or cold, or continuous on-line operation) or adverse impacts on natural or human environments (pollution or radiation). While this analysis is performed early in the process, it is not a once-and-for-all activity. Throughout the life of the program, the validity of mission and environmental requirements are analyzed and assessed for mission deficiencies and are revisited whenever they exhibit adverse impact on cost, schedule, performance, or risk.

Quite often customers define requirements as “thresholds” or “goals.” Thresholds are minimum requirements customers need to perform their missions. Goals are advanced qualities that provide added benefit. Achievement of a threshold is of utmost importance, since the customer has indicated he may not be able to perform the mission without it. Goals are less critical and the System Engineer should make the customer fully aware of any cost, schedule, performance or risks involved in their attainment before proceeding. Find out if the customer is willing to accept the added penalty associated with the benefit. Maybe it makes sense to put the goal on hold for later implementation. This is the customer’s choice, but the Systems Engineer has an obligation to provide all the information necessary to make that decision.

2.3.2.2 Identify Architectural and Functional Requirements

The major functions that the system needs to perform are identified and the appropriate system-level attributes (requirements) are assigned to them. In this activity, a system hierarchy is established and a system-level specification tree developed. Where a function involves more than one requirement, the requirements are apportioned over the affected function. For example, the function to provide spacecraft stability may be primarily influenced by spacecraft attitude pointing error, spacecraft pointing error rate, and spacecraft translation acceleration limits. Further allocations of each requirement will then be necessary. Continuing with our example, the requirement statement is to achieve an overall spacecraft pointing error of less than 250 micro-radians for each orthogonal axis. The allocations to the onboard instrumentation might be stated such that the operation of each of 2 instrumentation units shall contribute less than 100 micro-radians to total spacecraft attitude pointing error.

In this example, a derived set of attributes is assigned to a function because the system-level attribute cannot be allocated directly. The assembly of all allocated or derived functional requirements must equate to the originating specific and overall system requirements, and the traceability of functional-to-system requirements must be recorded and maintained. Individual requirements must be characterized in terms of the degree of certainty in their estimate, criticality to system success, and relationship to other requirements. Again, this is not a one-time process. Re-balancing of functional requirements may be necessary when system requirements change or when analyses indicate that requirements assigned to a specific function might be more advantageously met in another.

2.3.2.3 Define and Refine Requirements

2.3.2.3.1 Architectural Requirements

Architectural requirements are typically derived from enterprise and SoS mandates, joint concept of operations, or prior art. Each of these elements may require a specific method of system assembly and its corresponding internal or external interface. These requirements provide

- Common internal interfaces for modularity in system design that is consistent with prior art
- Common external interfaces for enterprise and SoS level interoperability
- Common functional hardware for reusability
- Ability to use open standards and protocols
- Ability to use existing software

2.3.2.3.2 Functional Requirements

Functional requirements capture and define the intended behavior of the system. These requirements, referring to a specific or discrete action that is necessary to achieve a given objective, document what a system must do to fulfill the operational capability needs. A functional requirement may involve calculations, technical details, or data manipulation to provide some part of the desired functionality.

2.3.2.3.3 Non-Functional or Dependability Requirements

These requirements do not directly contribute to the functional and performance needs of the system, but are necessary for the proper and dependable use of the system. Most are derived from specialty engineering mandates and considerations. For example, a system must meet mandated safety, environmental, and force survivability KPPs before it can be deployed in the field.

2.3.2.3.4 Performance Requirements

The mission/environments analysis and the functional requirements identification result in an initial set of performance requirements and design constraints assigned to major system functions. In the Functional Analysis and Allocation activity, this set is further divided and allocated as the first step in arriving at specifications suitable for the acquisition of hardware and software, and for recruiting and training of necessary personnel. These requirements are documented in a System Requirements Document (SRD) or system level specification. As this process of decomposition to lower levels progresses, the nature and validity of the original assignment of attributes to the functions is more fully understood. With this understanding, more efficient or effective functional divisions and requirements assignments may become apparent, necessitating a reassessment and modification of the original assumptions of the Requirements Analysis. This feedback completes the Requirements Loop.

2.3.2.3.5 Constraints

The constraints are either physical or are imposed as mandates to achieve a purpose that, at times, is necessary to achieve functionality or objectives that are beyond the development of the system at hand.

The physical constraints are generally based on the physical laws and the properties of materials that are required or available to build the system. For example, SV parts and materials are constrained in form and mass to survive the harsh thermal and vibrational environment of the launch.

The mandated constraints are more notable as they may be derived from a number of usability considerations. For example, they include enterprise or SoS level mandates for interoperability to provide superior execution or emergent capability to the warfighter.

2.3.2.4 Requirements Baseline

The requirements baseline is established when capability needs and other early SE activities including concept and technology development phases culminate in well-formed and feasible requirements for the system that are approved and put under configuration control. The requirements baseline consists of documented performance, functional, architectural, dependability, and constraints specifications. A requirements baseline includes:

- All necessary functional, architectural, performance, dependability, and constraint characteristics
- All necessary physical, legal, component standardization, open-systems, and interoperability constraints
- The verification required to demonstrate achievement of the specified characteristics
- The necessary interface characteristics with associated CIs, other system elements, and other systems
- Identification of lower level CIs, if any, and the configuration documentation for items (such as items separately developed or currently in the inventory) which are to be integrated or interfaced with the CI
- Integrated database that captures the entire program data, models and tools used, trade studies, metrics, changes, design rationale, traceability, verification, and other pertinent information on decisions or clarifications

2.3.2.5 Requirements Loop

Requirements loop as shown in Figure 2-4 provides feedback to iteratively refine and improve requirements as knowledge is gained and alternative solutions are discovered through functional analysis and allocation step in the SE process to be discussed below.

2.3.3 Functional Analysis, Allocation, and Validation

Functional analysis helps define functional areas, sequences, and interfaces. It supports requirements development, enhances understanding of the system through a common language, provides graphical representation, supports development of the system configuration and its operational use, and draws out any additional requirements the system must meet to represent a “complete” system.

Functional analysis is a top-down development process that:

- allows development of the system hierarchy

- facilitates identification of trade-studies and requirements flow-down
- allows most difficult area to be attacked first throughout the entire system hierarchy
- allows iteration to achieve complete system definition through functional decomposition and allocation
- supports development and sharing of a common understanding of stakeholder Objectives

The Functional Analysis and Allocation bridges the gap between the high level set of system requirements (from the Requirements Analysis) and the detailed set required (in Synthesis) to develop or purchase systems and implement programs. It is an integral part of both the Requirements Loop and the Design Loop. During this activity, an integrated functional architecture is defined in sufficient depth to support the synthesis of solutions in terms of people, products, and processes, and to allow identification and management of attendant risk. It is an iterative process, interacting and reacting to the on-going activities in both the Requirements and Design Loops.

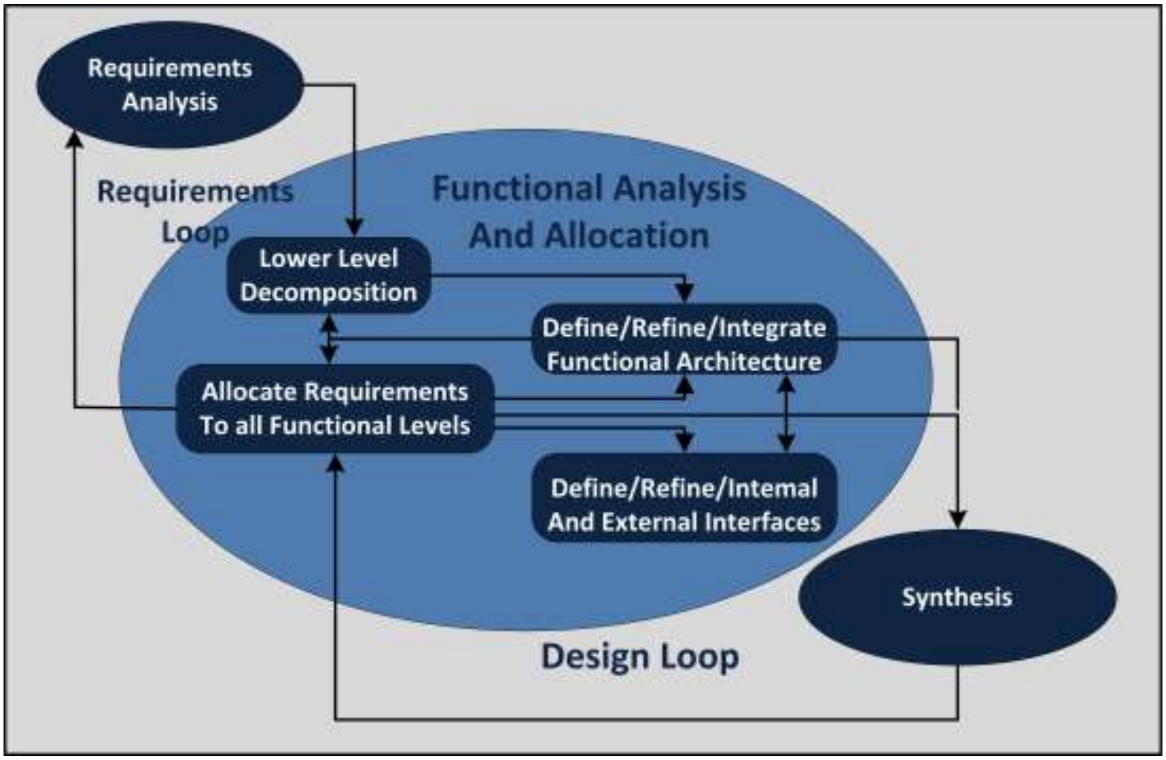


Figure 2-5 Functional analysis & allocations–create lower level requirements to aid synthesis of solutions

The initial step is to identify the lower-level functions required to perform the various system functions. As this is accomplished, the system requirements are allocated and functional architectures) are developed. These activities track and interact so that as details evolve, they are continually validated against each other. For example, GPS user equipment signal processing may require greater receiving sensitivity, or different decomposition may appear more advantageous. Perhaps, detection is simpler with increased processing rather than greater signal strength, leading to a re-evaluation of the driving requirements. Decisions may not be clear-cut. Consequently, alternate architectures and allocations may be carried through early stages of this activity until the optimum approach becomes apparent. The internal and external functional interfaces are defined as the architecture matures. The functional architecture(s) and their companion functional requirements are the input to the Synthesis activity. Completing the Design Loop, the detailed results of the

Synthesis are compared to the candidate architecture(s) and allocated requirements to help zero in on the optimum approach and to assure that all proposed solutions meet established requirements.

Functional analyses and concomitant trade-off studies lead to validated logical solutions or functional architectures. It is a logical process where top-level requirements are decomposed and assigned or allocated to constituent parts of the architected system. The end-products of the process, the functional architecture baseline, include:

- A necessary and sufficient representation of required operational capabilities consistent with concepts of operation, system behavior, and required functionality
- A complete representation of the functional and performance requirements in the requirements baseline
- An accurate model of the system behavior to include all sequencing, concurrency, and timing requirements
- An accurate and timely data flow relationships that provide data associations necessary to derive requirements from the functional or logical analyses
- A documented system architecture sufficiently decomposed to the point that each can be related to elements of the physical hierarchy to form the allocated baseline, and the allocation of the system performance requirements and design constraints to the lower levels
- A decision database including relevant trade-off studies for the selected allocation and architecture
- Documented and well-defined vendor-neutral open internal interfaces for modular reusable physical and logical design
- Documented and well-defined external interfaces using open industry standards and protocols to enable enterprise-wide interoperability
- A two-way traceability between each element of the requirements baseline and each element of the functional architecture
- A two-way traceability between each element of the functional architecture and the functional and physical elements of the system-level architectures
- Assessments that all requirements can be met and are consistent with a balanced approach to risk, cost, and schedule constraints

A number of methods, techniques and tools are available to perform functional analysis and allocation. They include:

- Mandated DoDAF architecture templates
- Functional Flow Block Diagrams (FFBD)
- IDEF0
- System Modeling Language (SysML™)
- Unified Modeling Language (UML)
- NxN Diagrams
- Physical decomposition or Work Breakdown Structure

Chapter 5 provides more information on these and other tools relevant to SE. Detailed descriptions of the activities of the Functional Analysis and Allocation are shown in Figure 2-5 and discussed below. (A more detailed step by step functional analysis and decomposition, allocation, assessment of requirements development process is described with examples in Appendix C.

2.3.3.1 Lower Level Decomposition

Decomposition to lower-level functions is the incoming interface for the Requirements Loop. The functions identified in the Requirements Analysis are analyzed to define successively lower-levels of functions that accomplish the higher-level functional requirements. Alternate lower-level functional solutions covering all anticipated operating modes are proposed and evaluated to determine which provides the best fit to the parent requirements and best balance between conflicting ones. The initial decomposition is the starting point for the development of the functional architecture and the allocation of requirements to the lower functional levels. Adjustments to the decomposition strategy may be necessary as details are developed.

2.3.3.2 Allocate Requirements to All Functional Levels

All requirements of the top-level functions must be met by the aggregate of those for all lower-level functions. This is often difficult to prove when an upper-level performance requirement is achieved through a number of derived requirements. (For instance, system accuracy is composed of derived functional attributes that in sum determine its value.) Consequently it is extremely important not only to ensure higher-level requirements are allocated properly, but also that traceability to the originating requirement and rationale for the allocation be recorded and maintained. Traceability is an on-going record of the pedigree of requirements imposed on system and subsystem elements. Expressed in terms of “parents” and “children” and recorded on a suitable database, traceability allows the Systems Engineer to ascertain rapidly what effects any proposed changes in requirements may have on related requirements at any system level.) Because requirements are derived or apportioned among several functions, they must be traceable across functional boundaries to parent and child requirements. Design constraints defined in the Requirements Analysis must also flow down to the lower functions. The allocated requirements must be defined in measurable terms, contain applicable go/no go criteria, and be in sufficient detail to be used as design criteria in the subsequent Synthesis activity.

Time dependent operations are also allocated to the functions. If the total time required for the system to perform an operation is critical, the time allowed for each function to perform its portion of the process must be allocated and the sequence specified. For each sequence, the characteristics of the inputs and outputs between functions must be identified.

In completion of the Requirements Loop, as the functional allocations are established they are continually evaluated against the original requirements. In addition, the functional allocations are one of the criteria used in parallel activities of functional architecture and interfaces definition. If required, the allocations may be modified as a result of these activities. In some cases, this may reflect into reassessments of the Requirements Analysis results.

The allocated requirements along with the associated architecture form the input to the Synthesis activity. Results of the Synthesis are validated against the allocated requirements and occasionally necessitate re-allocation.

2.3.3.3 Define, Refine, and Integrate Functional Architecture

The functional architecture defines how the functions will operate together to perform the system mission(s). Generally, more than one architecture can satisfy the requirements. Usually, each architecture and associated set of allocated requirements have different cost, schedule, performance, and risk implications. Not only is it difficult at this point to ascertain which is the optimum solution, it is usually prudent to carry along low-cost, low-risk, lower-performance alternatives as insurance in case the higher-performance solution proves not feasible, too costly, or not possible to achieve in time for the need. In the Design Loop, synthesized designs are compared with the originating architectures and allocated requirements to assure compliance or to initiate re-evaluation.

Sometimes it is necessary to drive toward optimal solutions by presenting various functional views including those that depict functional relationships with existing assets to enable more thorough assessments of plausible solutions. For example, we might choose to consider the NASA's Deep Space Network (DSN) to provide communication capabilities for our system under consideration. Further decomposition of the functional elements would also greatly assist in interface definition between the system and existing assets. Inherent in the process of establishing the architecture is the definition of the boundaries of the various functions and sub-functions. This leads to the definition of the internal and external interfaces.

2.3.3.3.1 Define and Refine Interfaces

System interfaces are both physical and functional. Interface definition and control are two SE activities that begin in parallel with the development of functional architectures. Typically, a system is initially depicted by a System Diagram that bounds the system by depicting the system along with its external elements. Source documentation, such as ICDs, CDDs, external element specifications, and interface documents, might also provide interface requirements to ensure interoperability between systems and make sure required capabilities are achieved. An operational concept may also provide descriptions, interactions, and requirements between the system and the external elements. An interface definition process will evolve interface architectures and requirements in conjunction with the overall systems definition process. The interfaces will mature as the operational and system requirements mature. First, an initial top level interface architecture is created. This architecture is also a reflection of the system concept. If alternative concepts are under consideration, alternative interface architectures are also developed. The functional decompositions of the interfaces are performed in concert with that of the system since the interface elements must be clearly identified with system architectures. This one-for-one correlation initiates the interface architecture that is triggered by and traceable to known system functions and any associated source and derived requirements. This procedure significantly reduces requirements conflicts and supports a more rapid interface design change process.

Often, interface architectures focus on the system communications. For example, protocol and data segments define the communications interface between the system functional groups. Standards are often selected to ensure the interfaces are sufficiently defined and interconnected between 2 elements. A design solution for a communications interface may include a bus interchange unit, signal lines, transceivers for the nodes, and possibly memory devices to physically represent a communications interface. Design solutions are the subject of the next section.

2.3.3.4 Functional/architectural Allocated Baseline

Allocated baseline describes functional, performance, interoperability, and interface requirements that are allocated from those of a system or higher level configuration item. The allocated baseline correlates to the second and lower levels of the work breakdown structure. The allocated baseline is established and put under configuration control at each configuration item's (hardware and software) PDR, while the entire system's allocated baseline is established at the system-level PDR. The end product of this activity defines the allocated baseline that includes:

- assembly specifications
- assembly and component interface specifications
- subsystem specifications
- an item or software requirements specification.
- Interface control documents,
- Interface requirements specifications,
- Item/software requirements specifications for lower-level CI, if any
- design restraints

- the verification required to demonstrate the achievement of specified functional and interface characteristics
- Integrated database that captures the design, data, models and tools used, metrics, changes, design rationale, and other pertinent information on decisions or clarification made to subsystem requirements.

2.3.4 Design Solutions (Synthesis)

Design solutions part of the SE process translates functional architectures and their associated requirements into physical architectures and one or more physical sets of hardware, software and personnel solutions. It is the output end of the Design Loop. As the designs are formulated, their characteristics are compared to the original requirements, developed at the beginning of the process, to verify the fit. The output of this activity is a set of analysis-verified specifications that describe a balanced, integrated system meeting the requirements, and a database that documents the process and rationale used to establish these specifications.

The first step (Figure 2-6) is to group the functions into physical architectures. This high-level structure is used to define system concepts and products and processes, which can be used to implement the concepts. Growing out of these efforts are the internal and external interfaces. As concepts are developed, they are fed back in the Design Loop to ascertain that functional requirements have been satisfied. The mature concepts and product and process solutions are verified against the original system requirements before they are released as the SE Process product output.

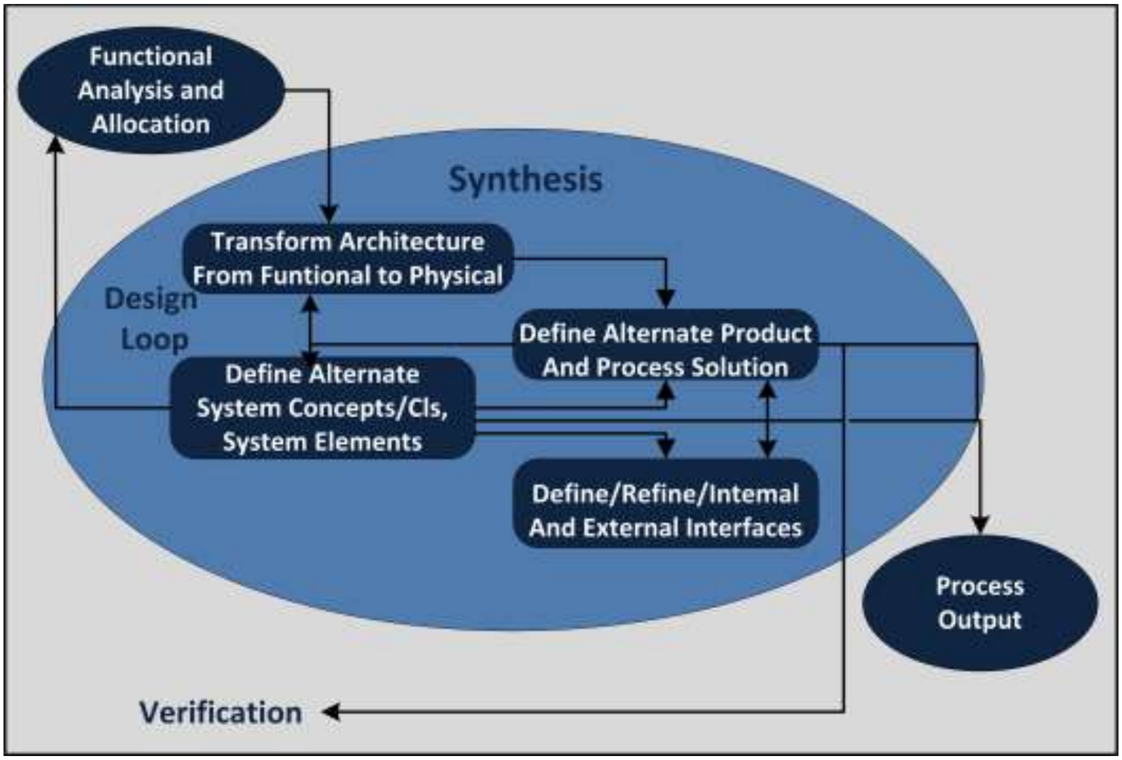


Figure 2-6 Synthesis–developing detailed solutions

Using functional architecture baseline, the system element design solution and validation activities begin to group and translate functions into physical architectures, allocating requirements to system constituents. The end-product of this part-process is an allocated baseline. This is achieved iteratively with the Functional Analysis, Allocation, and Validation process through the design loop, but can possibly reach back to

Requirements Analysis and Validation process through the requirements loop when problems arise. Detailed descriptions of the activities of Synthesis as shown in Figure 2-6 are provided below.

2.3.4.1 Transform Architecture

For each set of inputs from the Functional Analysis and Allocation, like functions are grouped together to form major physical system elements, an integrated physical system architecture is developed, and the interaction of the elements established. As a part of this process, the completeness and adequacy of the input functional and performance requirements are established and if additional ones are necessary, the Functional Analysis and Allocation is revisited. The physical architectures as well as composite (functional and physical architectures) are used as the basis for defining system concepts. Data fed back from the concept development may result in refinement of the architectures.

In the development of physical architectures (and composite physical and functional architectures), it is important to retain and enhance any open-systems features built-in during Functional Analysis and Allocation. Failure to do so may result in sub-optimized design, loss of opportunity to incorporate on-going technology advancements or replacements during development or subsequent sustainment, and even reduce the effective life of the system. Recent emphasis has been placed on open systems architectures. Such architectures facilitate use of COTS solutions for system implementation, later incorporation of advanced or replacement technologies, expansion of system capabilities, and interoperability with existing or prospective related systems. The flexibility provided by open systems architecture during all phases of system development, recommends its consideration in making all SE decisions.

2.3.4.2 Alternate System Concepts and Elements Definition

The elements of the various architectures must be developed in sufficient detail to permit verification of the design against the requirements and constraints of the Requirements Analysis, and to eventually lead to detailed system design. In defining system implementation concepts, functions are assigned to system constituents which can be the subsystems or components to perform various system functions. Functions might be distributed among several system constituents. Likewise, there usually are several ways in which the boundaries of each system constituent can be defined, for example, pre-amplifiers might be mounted with an antenna, or included in a receiver. Consequently, several system implementations are usually proposed and further analysis performed in the Design Loop to determine which best fits the requirements.

Another important aspect of this activity is identification of the critical parameters of each alternate concept and the sensitivity of the concept's performance, cost, schedule or risk to each parameter. The sensitivity may weigh heavily in trade studies performed in the System Analysis and Control activity, and may help decide which concepts are carried further in the Design Loop.

The output of this activity is integrated logical sets of systems, configuration Items (CIs), and system element solutions. They are evaluated repeatedly in the Design Loop to shake out those that do not meet the requirements. The remaining sets are further verified to arrive at the optimum solution(s). The design concepts are handed off for definition of the interfaces and product/process solutions. Results from these parallel activities are fed back to refine the system concepts.

2.3.4.3 Alternate Product and Process Definition

Just as there are several ways to implement system configurations, there are also many ways in which these configurations may be accomplished. The Alternate Product and Process activity addresses such questions as the use of COTS (commercial off-the-shelf) products versus new or modified development, LSI (large scale integration) versus discrete or hybrid circuitry, human versus machine operations, and new versus existing technology. As alternates are developed, design simplicity approaches are incorporated to take maximum advantage of standardization, modularity, existing support equipment and facilities, and production techniques.

Another major consideration in this activity is the determination of how much automation to incorporate. Where the man-machine interface is drawn may cause large variations on the workloads on both sides of the interface. This could have considerable impact on the cost, performance, schedule and/or risk of alternate configurations. Many times the decision is deferred until later in the program. Costs of automation for all possible configurations may be prohibitive, so human operations may be incorporated during the concept demonstration phase of the program with the idea of automating later when the system has been defined in more detail.

The Alternate Product and Processes activity reacts interactively with the architecture development, systems concept definitions, and interfaces definition activities. Where appropriate, the results, complete with all applicable tolerances and variables, are included with the associated system concept in the process output database.

As described earlier, SE has both technical and management aspects. One of the management tasks of the Synthesis function is developing a Work Breakdown Structure (WBS), which is used in managing the development of the system described in Synthesis.

2.3.4.4 Define and Refine Physical Interfaces

This is a continuation and extension of the work began in the Functional Analysis and Allocation and is the foundation of the Configuration Management operations that continue through the life of the program. The functional and physical characteristics of the inputs and outputs at the boundaries identified during Synthesis activities must be identified and documented in a set of Interface Control Documents (ICDs). In addition to this accounting, methods must be established for tracing requirements across the interfaces and aggregating them as necessary to permit comparison with the original driving requirements and constraints resulting from the Requirements Analysis.

This activity has both engineering and legal ramifications. The interfaces are an important factor in establishing contracting and subcontracting agreements and in assuring that items made by various suppliers play together as a system.

The interface definition is iterated as the system concepts are developed, and as alternate product/process solutions are defined. For each surviving system definition, the associated final set of interfaces is included in the database of the process output.

This is also the time when implementation processes and related considerations drive refinement of design for cost-effective manufacturing, dependability, and sustainment.

2.3.4.5 Design Loop

Design loop provides feedback to iteratively refine and improve allocated and design release baseline requirements as knowledge is gained and alternative physical design solutions are discovered in the synthesis part of the SE process to be discussed below.

2.3.4.6 Design Baseline

The design baseline is described by the build-to item detail specification and its technical data package. The design baseline is established and put under configuration control at each configuration item's (hardware and software) critical design review (CDR), while the system design baseline is established at the system-level

CDR. Requirements and allocated baseline are implemented in detailed design to the component and assembly level. Design baseline represents a fully verified product configuration.

2.3.4.6.1 Design for Implementation, Deployment, and O&S

At this juncture the system design has matured. Both the product and process specifications are validated, approved by the customer, and maintained in controlled configuration. The design release baseline

- Fully reflects the allocated baseline
- Identifies all additional system products necessary to manufacture, code, author, or buy; integrate, verify; deploy; train; operate; support/sustain; and dispose of the system and its constituent products over the life cycle.
- Is designed to implement interoperability with both internal and external interfaces
- Systematically derives functionality from the operationally stated interoperability constraints.
- Integrates the functional and physical interface designs and associated functions and requirements across systems.
- Parts, materials, and processes are fully characterized including consumables
- Dependability (reliability, availability, maintainability, and safety) data is complete and acceptable

2.3.5 Implementation

The system is implemented based on the design release baseline (see Figure 2-3). The product configuration baseline details the system to its lowest level components that are fabricated or coded in a manner that fully satisfy design release baseline. It also documents customer approved design deviations, if any, maintaining a transparent two-way traceability all the way up to the SE input CBA and other documents.

2.3.5.1 Product Baseline

The product baseline is the approved technical documentation which describes the as-built configuration of a CI, subsystem, and ultimately the entire system during the production, fielding, deployment, and operational support phases of its life cycle. The product baseline prescribes:

- All necessary 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

The product baseline is verified through the system verification review and the functional configuration audit. It is finalized and validated at the physical configuration audit (PCA).

2.3.6 Integration

This is the crucial part of the SE process where design becomes reality (see Figure 2-3). System parts, components, and software are assembled and integrated in a systematic fashion to build the system from ground up, using well-defined verification and validation procedures. Complex systems require a written integration plan. All discrepancies are reported and analyzed for corrective action. Approved corrective action procedures are integrated with standing assembly and verification procedures. Any necessary changes to the parts, materials, and processes are configuration controlled and correlated to the earlier baselines.

2.3.7 Verification and Validation

Verification and validation are concurrent SE activities that begin at the inception of the program (see Figure 2-3). Concepts are analyzed for their feasibility. Requirements analysis is not complete without identifying a verification method and procedure. Tests are developed, and at times are built into or integrated with the hardware and software to verify and validate system capability.

Requirements verification process is used to see if they are necessary and sufficient to meet the stakeholder's needs. The verification process is used to show that the as-built system, sub-system, and components meet all of stakeholder requirements. In short, it answers the question if the system was built right. The verification plan and procedures have a two-way traceability to all system, subsystem, and component requirements to ensure that all requirements are verified. Engineering verification activities verify that the system design meets requirements, ensure suitability to stakeholder's needs, and help characterize design margin and dependability of the system.

Validation activity provides proof that the system is performing as expected. The validation process is used to ascertain if the right system was built. Typically, a validation plan is developed to systematically assess the system, sub-system, or component in operation. Validation plan and procedures are traceable to the concept of operations.

A more detailed discussion of verification and validation activities, methods, and procedures is provided in Chapter 7.

2.3.7.1 Verification Loop

Verification loop shows that the synthesized design meets all requirements. It provides iterative feedback on the health of the SE process. Requirements, architectures, and design aspects of the systems are continuously tested for their sufficiency and necessity to satisfy stakeholder capability needs cost-effectively.

2.3.8 Deployment, Sustainment, and Disposal

A validated system is deployed in the field to provide its intended capability to the users (see Figure 2-3. Complex systems have a well-defined deployment strategy. Typically, such systems are deployed incrementally. The system may be installed at a single site and tested in the field conditions for final adjustments before full deployment to multiple sites over an extended period of time. A complex deployment also may require post acceptance testing at each site.

All systems require regular maintenance and consumables need to be replenished as needed. Sustainment involves planning and executing activities that include operating the system, monitoring system performance, making repairs, hiring and training operators, testing the system after any changes are made, and keeping track of the system's configuration the system. Preventive maintenance involves inspection and proactive actions, such as cleaning, replacement of components prior to the end of their rated life, backing up software, storing data, and replacing components that have become obsolete and unsupported. Reactive maintenance involves correcting faults when they occur. Software maintenance involves correcting malfunctions when they are discovered, upgrading components that become obsolete and unsupported, and making minor modifications as needed to improve functionality.

Disposal activities are needed to determine when a system or major sub-system needs to be retired or replaced. Disposal plans for complex systems generally provide guidance on system element disposition and replacement of lost capability. High value items and hazardous materials generally require special handling.

2.3.9 SE Technical Management Process

System Analysis and Control is the glue that holds all the other SE Process activities together. It is the activity that spans the whole life of the program. It involves the initial analysis of system requirements to deliver the expected capability, the management of the activities necessary to develop the required capability and its interactions with other enterprise systems, the review and measurement of work progress, and the documentation of work actions and results.

System Analysis and Control (Figure 2-7) interacts with all the other activities of the SE Process. Because it is so extensive, this interrelationship has been mentioned only briefly in the previous discussions of the other activities to allow a more comprehensive review at this point. The initial analyses performed in this activity are the basis for the Systems Engineering Management Plan (SEMP) and the SE entries in the Integrated Master Plan (IMP) that define the overall SE effort. The SEMP is a process-oriented document, which describes what has to be done; the IMP is event oriented, identifies the significant accomplishments to complete each event, and defines the criteria for successful completion of each accomplishment. From the SEMP and IMP, the Integrated Master Schedule (IMS) is developed to relate the IMP events and SEMP processes to calendar dates.[20] Once the SEMP, IMP, and IMS are in place, the “control and manage” activity shown in Figure 2-7 directs their accomplishment.

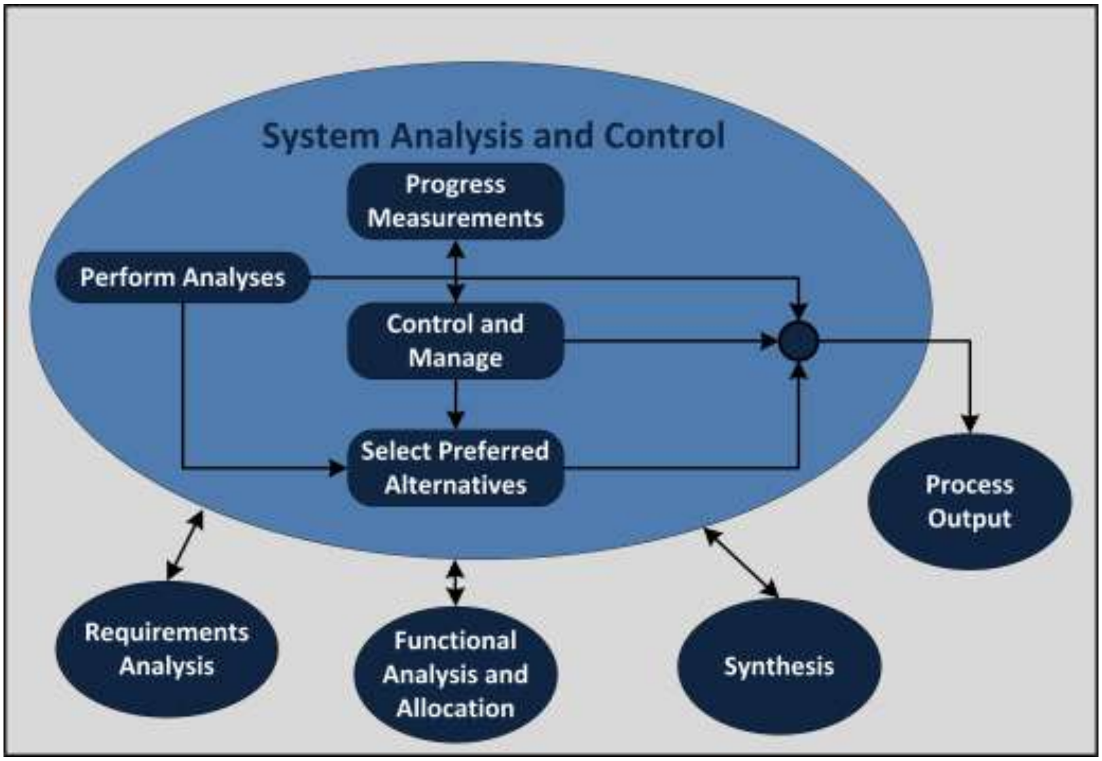


Figure 2-7 System analysis and control

As the process progresses, trade-off studies and system/cost effectiveness analyses are performed in support of the evaluation and selection processes of the other activities. Risk identification/reduction studies are conducted to aid in risk management. Analyses also identify critical parameters to be used in progress measurement.

---

[20] The IMP and IMS are used by programs applying Integrated Product and Process Development (IPPD) to plan the systems engineering activity as an integrated part of the overall work necessary to complete program. The draft MIL-STD 499B and the early EIA/IS-632 and IEEE P1220 standards (all issued in the mid-1990s) used the term Systems Engineering Master Schedule (SEMS) for a plan equivalent to the IMP but covering only systems engineering and Systems Engineering Detailed Schedule (SEDS) for a schedule equivalent to the systems engineering elements of the IMS. In the ANSI/EIA-632-1998, the SEMP is called an Engineering Plan. In the IEEE Std 1220-1998, the corresponding terms are the system engineering management plan or engineering plan, the master schedule, and the detailed schedule.

The management activity directs all operations and also performs configuration management (CM), interface management (IM) and data management (DM). It specifies the performance parameters to be tracked for progress measurement. It conducts reviews and reports progress.

The information from the System Analysis and Control activity is a major part of the SE process database that forms the process output. The control and manage activity contributes a record of the process as well as CM, IM and DM data. The analysis activity provides the results of all analyses performed, identifies approaches considered and discarded, and the rationales used to reach all conclusions. The selected preferred alternatives are recorded with the associated criteria and methodology for selection. Detailed descriptions of the activities of System Analysis and Control are provided below.

2.3.9.1 Trade Studies and Analyses

Initial analyses identify the salient factors of the program and its requirements providing the basis for planning the SE effort. Subsequent analyses support the selection and refining operations of the other activities of the SE Process. These analyses include trade-off studies, system/cost effectiveness analyses, and risk identification. Trade-off studies analyze the differences between alternate approaches. System analyses look at aggregate systems solutions and determine their performance characteristics. Cost effectiveness analyses establish the costs and associated benefits of candidate system concepts, functional configurations, products and processes. Risk identification analyzes all parts of candidate approaches and their associated program elements to isolate and evaluate the risk involved in their use. As the SE Process advances from Requirements Analysis through Synthesis, the analyses become more detailed.

The trade-off studies supporting the other SE activities are discussed in the following subsections.

2.3.9.2 Alternative Architecture Analysis

The Analysis of Alternatives (AoA) evaluates the operational effectiveness, operational suitability and estimated costs of alternative systems to meet a mission capability. The analysis assesses advantages and disadvantages of alternatives being considered to satisfy capabilities, including the sensitivity of each alternative to possible changes in key assumptions or variables. The AoA provides the basis for choosing a specific concept and for the JCIDS to refine the capabilities to be provided in the Capability Development Document (CDD) to support the initiation of a formal acquisition program.

2.3.9.3 Requirements Analysis

Trade-off studies establish alternate performance and functional requirements. Often these studies identify major cost drivers to assist the customer in refining his requirements to obtain the most effective cost/performance mix. These studies may also influence changes to architecture concepts.

2.3.9.4 Functional Analysis and Allocation

Trade-offs provide evaluations of alternate functional architectures, help define derived requirements and resolve their allocation to lower levels, and aid in selecting the preferred set of performance requirements at functional interfaces.

2.3.9.5 Synthesis

Trade studies support decisions on use of new versus non-development products and processes; establish system and CI configurations; assist selection of system concepts, designs, and solutions (based on people, parts and materials availability); support materials/processes selections and Make-or-Buy decisions, examine proposed changes; investigate alternate technologies for risk/cost reduction; evaluate environmental and cost impacts; establish standardization to reduce life-cycle costs; and evaluate and select preferred products and processes.

System Analyses are performed to assist in the development of candidate functional and physical configurations and to determine the performance of each candidate. The analyses also provide a methodology and mechanism to establish, track and control analytical relationships and measures of effectiveness, and permit traceability across functional and physical interfaces. Integral to this process is the identification of critical factors to support decisions and permit technical performance measurement.

Cost-effectiveness analyses determine the cost/benefit characteristics of candidate systems approaches to assist in selecting the preferred alternative(s). These analyses support the three other SE Process activities and are a major factor in selecting the preferred alternative(s).

Risk analyses identify critical parameters that might be risk drivers. Potential sources include both individual items and groups of items where interrelationships may contribute to risks. For example, a product might itself be low risk, but because it must be matched to a high-risk new development item, use of the product might be high risk also. Risks are quantified for cost, schedule and performance impact. Also examined are design, cost and schedule uncertainties, and the risk sensitivity of program, product, and process assumptions. The analyses pinpoint areas that require risk management in the control and management activity.

2.3.9.6 Control and Manage

This activity interfaces with all other activities of the process. It plans and manages the activities, monitors and reports status, coordinates actions, and documents in the process output database all progress, results, decisions, and rationales for decisions. It promulgates the SEMP, and the SE entries into the IMP and IMS, and any lower order plans or schedules required to implement them. It also includes the activities of Risk Management, Interface Management, Data Management, and Configuration Management. It is responsible for the conduct of technical reviews and audits. It identifies the items to be tracked for technical performance measurement. The Control and Manage activities are addressed in more detail in Chapter 4.

2.3.9.7 Selected Preferred Alternatives

Based on analyses performed within the System Analysis and Control activity and within the Functional Analysis and Allocation and the Synthesis activities, preferred alternates are selected. The selections are made at increasingly fine-grained levels of system description. In support of the Functional Analysis and Allocation activity, these selections are made to determine which functional architecture and definitions should undergo continued development and which should be discarded. Technology, process, and product selection revolves around physical systems architectures. The selection process helps determine technologies necessary to prove concepts and identifies technologies that may be inserted later as design matures.

2.3.9.8 Make Progress Measurements

The Control and Manage activity determines which measures of effectiveness will be tracked and reported. Once this has been accomplished, the other activities are directed to supply the requisite data. The Progress Measurement compiles and analyzes the data for use by the Control and Manage activity to direct the program and report progress.

3 DoD Space Systems Acquisition Life Cycle

As introduced in Chapter 1, the DoD has three overarching and interactive management systems to implement the acquisition process. These are the JCIDS, The Defense Acquisition System (DAS), and the Planning, Programming, Budgeting, and Execution (PPBE) process. These processes and their interaction for space programs at SMC are discussed in more detail in this chapter.

The JCIDS process provides for early SE. Service operators and program planners sponsor new or incremental development of programs to bridge recognized capability gap(s), under JROC Oversight as documented in CJCSI 5123.01 and CJCSI 3170.01. This process is joined by the Developmental Planning at SMC. Capability-based analysis, concept development, architectures, and analysis of alternatives is performed to ascertain technical and financial feasibility of the project. A number of foundational documents are produced to include ICD, CCTD, and AoA. An MDD is required to proceed with the program into the MSA phase of the DAS process as documented in DoDD 5000.01 and DoDI 5000.02. A PPBE process manages and oversees funding of the program as documented in DoDD 7045.14 and DoDI 7045.7. All three of these processes are supported by SE activities that provide assessments of cost, schedule, and risk based on the evolving design of the system that is to provide the capability.

3.1 The JCIDS Process

JCIDS process provides for a structured methodology to identify and bridge capability gaps as seen by the operators, technology developers, and sponsors.

While the process is provided for a typical and normal development and fielding of new or improved capability, it makes room for the accelerated fulfilment of urgent warfighter needs. (For more details and current doctrine see JCIDS Manual as provided and continuously updated on the Internet.)

3.1.1 CBA and Developmental Planning

Primarily based on and governed by the JCIDS process, CBA develops potential materiel and non-materiel concepts to address capability gaps and shortfalls, or to exploit new capabilities provided by new technologies. CBA/JCIDS initiates the early SE efforts, sponsored by Major Command (MAJCOM). It is a team effort to identify any capability shortfalls, perform trade analyses, and consider potential alternative solutions.

The acquiring command(s) lead concept exploration and development, typically through XR organizations, and technical solutions. Air Force Research Laboratory (AFRL) also assists in identifying the projected availability of technologies to help overcome the capability shortfalls. The MAJCOM is responsible for submitting JCIDS documentation; all team member organizations participate in development of supporting material and in reviews.

CBA process starts with concept exploration and refinement that can help bridge the perceived capability gap. Both materiel and non-materiel solutions are considered. Each concept is researched and evaluated against capability needs. Various architecture products are produced to document analyses that include capability, all view, operational, and standards viewpoints as described in the DoDAF. This is followed by the AoA and the CCTD documents which form the basis for an SRD. If a materiel solution is necessary, and MDD is sought on the basis of these documented solutions. Common iterative SE methods as shown in Figure 1-7 in Chapter 1 are used to perform these activities. These activities are rigorously controlled and documented by the SE management process. The CBA and DP processes output the following:

- Identified materiel solutions that fulfill validated capability gaps
- Identified functional and data interoperability requirements within the enterprise
- Definitions of future capability needs and operational requirements
- Defined and evaluated alternatives concepts
- Assessed technology maturity and risk drivers
- Developed achievable system requirements through trade studies and M&S
- Identified supportability and sustainment requirements
- Developed architecture viewpoints to include all view, operational, capability, and standards viewpoints
- Defined preferred concepts or candidate solutions
- Developed executable acquisition strategies and funding profile

A summary of CBA/JCIDS and DP studies and products from the early SE activities is summarized in Highlight Box 3-1.

3.2 The DAS Process

DAS is the management process used by DoD to provide effective, affordable, and timely systems to the users as introduced in Chapter 1.

A summary of the DAS process phases, based on DoDD 5000.01 and DoDI 5000.02, is shown in Figure 3-1. Figure 3-2 provides an overview of the interaction between the DAS and the JCIDS processes. A DAS process is initiated when the ICD demonstrates the need for a materiel solution. A Directorate is funded to field a new, improved or continuing weapon or information system or service to fulfil an approved capability need. The capability need decision is documented in an MDD at the end of the CBA and in related early SE activities conducted and completed at the behest of MAJCOM or other sponsors. This can involve startup of a new program or extension of an existing program structure charged with the task of development of materiel solutions and technology to produce and field the required capability.

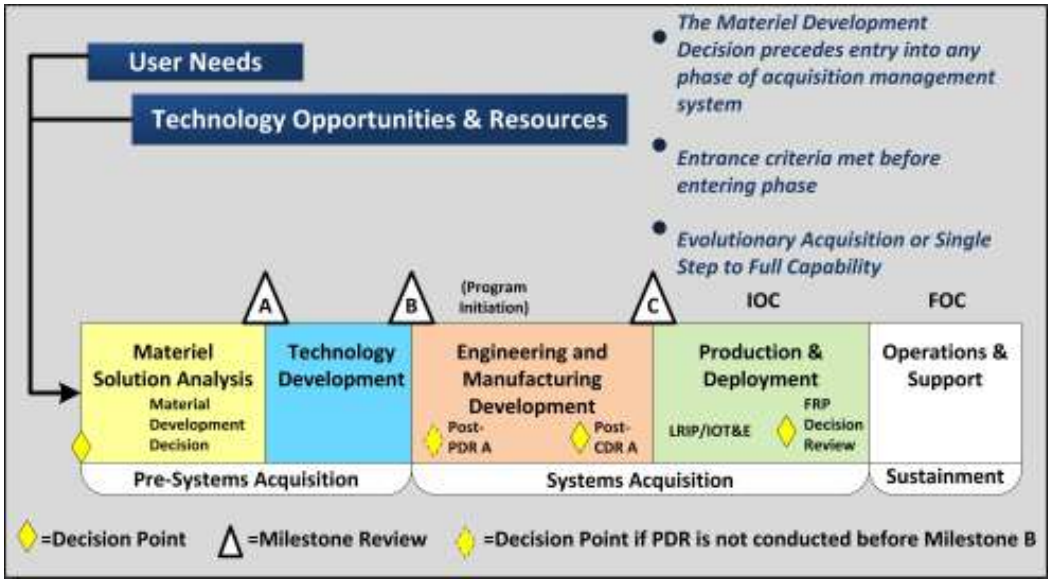


Figure 3-1 Phased DAS process

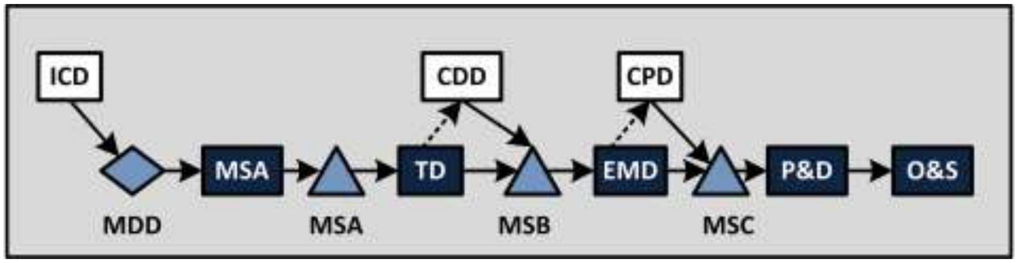


Figure 3-2 Interaction between DAS and the JCIDS process

DAS is an evolutionary acquisition system, allowing DoD to field mature technology to the user rapidly. Capability is delivered in increaments as it becomes affordable and available. Each increment, with its own well-defined objective, is a militarily useful and supportable operational capability that can be developed, produced, deployed, and sustained.

Highlight Box 3-1 summarizes CBA and Developmental Planning products.

Highlight Box 3-1

CBA/Developmental Planning Products

Decomposed ICD/CONOPS requirements
DoDAF Architectures
AoA Study Guidance
CCTD
Alternative solution space
Interoperability and Supportability Concepts
MDD

3.2.1 Materiel Solution Analysis (MSA)

The purpose of this phase is to assess potential materiel solutions and to satisfy the phase-specific entrance criteria for the next program milestone designated by the MDA. An approved ICD, supporting CBA and related concept and architecture documents, and an MDD is required to initiate the MSA phase. Funding is limited to MSA objective.

MDD review includes JROC recommendation and ICD which contains the preliminary concept of operations, a description of the needed capability, the operational risk, and the basis for determining non-materiel approaches which will satisfy the capability gap(s). Furthermore, MDD review provides for study guidance for the AoA to assess preliminary solutions, identify key technologies and estimate life-cycle costs.

Highlight Box 3-2

MSA Products

Draft CDD
Approved Materiel Solution
Support & Maintenance Concepts & Technologies
Systems Engineering Plan
AoA Report
T&E Strategy
System Safety Analysis
ISP / AoA / TDS / IBR
Cost/Manpower Estimates

ICD and AoA study guidance are used as major inputs to AoA and other MSA phase activities. MSA ends when (i) the AoA has been completed, (ii) materiel solution options for the capability need have been deemed feasible and affordable for recommendation, and (iii) the entrance criteria for the initial review milestone has been satisfied. Highlight Box 3-2 shows a list of major program products and documents at the successful conclusion of the MSA phase.

3.2.2 Technology Development (TD)

The purpose TD phase is to reduce technology risk by identifying and, if necessary, maturing appropriate set of technologies to be integrated into a full system. Extended M&S and prototypes may be necessary to demonstrate readiness. TD requires close and continuous collaboration between the S&T community, the user, and the system developer. It is an iterative process designed to assess the viability of technologies while simultaneously refining user requirements. A completed AoA, a proposed materiel solution, and appropriate funding for the planned activities are required to begin the TD phase.

When the cost estimate increases by 25 percent or more during TD, the MDA can decide whether the program is consistent with the priority level assigned by the JROC based on the interest of national defense or cancel the Milestone A approval.

Highlight Box 3-3 shows a list of major program products required at the completion of the TD phase.

Highlight Box 3-3
TD Products

System Allocated Baseline
PDR Report / Test Reports
SEP / TEMP / PESHE / PPP
TRA
NEPA Compliance Schedule
Risk Assessment
Validated Systems O&S Requirements
System Safety Analysis
CDD / ISP / STA / IBR
Acquisition Strategy
Affordability Assessment
Cost/Manpower Estimates

3.2.3 Engineering and Manufacturing Development (EMD)

The purpose of the EMD Phase is to develop a system or an increment of capability. This includes: (i) development of an affordable and executable manufacturing process, (ii) ensuring optimal logistics and operational supportability, (iii) implementing human systems integration, (iv) design for producibility, (v) ensuring affordability, (vi) protecting CPI by implementing appropriate techniques such as anti-tamper; and (vii) demonstrating system integration, interoperability, safety, and utility. EMD effort is based on TD knowledgebase that includes the CDD, Acquisition Strategy, SEP, and Test and Evaluation Master Plan (TEMP).

For a new program or an evolutionary increment, the EMD begins at Milestone B, after the maturity of the technology has been established and appropriate funding has been secured.

EMD activities are intended to produce an integrated system design that provides necessary system capability and an affordable manufacturing process. EMD integrates acquisition, engineering, and manufacturing processes with T&E.

Highlight Box 3-4
EMD Products

Initial Product Baseline
Test Reports
SEP / TRA / PESHE / TEMP
NEPA Compliance Schedule
Elements of Product Support
Risk Assessment
Life Cycle Sustainment Plan
System Safety Analysis
CPD / STA / ISP / IBR
Cost/Manpower Estimates

Highlight Box 3-5
Prod & Deploy Products

Product Baseline
Test Reports
SEP / PESHE / TEMP
System Safety Analysis
IBR
Cost/Manpower Estimates
Product Support Package

Highlight Box 3-4 shows a list of major program products required at the completion of the EMD phase.

3.2.4 Production and Deployment (P&D)

The purpose of the Production and Deployment Phase is to field specified system operational capability at the end of EMD. Highlight Box 3-5 shows a list of major program products required at the completion of the P&D phase.

The MDA makes the decision to enter P&D by authorizing LRIP after a successful Milestone C, with subsequent reviews and descision to authorize full rate production. Entry to P&D further includes: (i) acceptable performance in developmental test and evaluation and operational assessment, (ii) mature software capability, (iii) no significant manufacturing risks, (iv) an approved Capability Production Document (CPD) and refined integrated architecture, (v) acceptable interoperability, (vi) acceptable operational supportability, and (vii) a demonstration that the system is affordable throughout the life cycle, and fully funded.

P&D has two major components: An LRIP followed by a full-rate production decision review authorizing a full-rate production and deployment. In case of programs with no production components like software systems, a full deployment decision review is required.

3.2.5 Operation and Sustainment (O&S)

The purpose of the Operations and Support Phase is to execute a support program that meets materiel readiness and operational support performance requirements, and sustains the system in the most cost-effective manner over its total life cycle. Planning for this phase shall begin prior to program initiation and shall be documented in the Life-Cycle Sustainment Plan (LCSP). Operations and Support has two major efforts, Life-Cycle Sustainment and Disposal.

Entrance into the Operations and Support Phase depends on (i) an approved CPD, (ii) an approved LCSP, and (iii) a successful Full-Rate Production (FRP) Decision.

Life-cycle sustainment planning and execution commences as the system is pressed into service. The sustainment activities continue until the system is retired and disposed as planned. Life-cycle sustainment planning is considered early in the program during Materiel Solution Analysis, and is matured over time. A LCSP shall be prepared at Milestone B. The planning should be flexible and performance-oriented and reflect an evolutionary approach that accommodates modifications and upgrades as necessary. The LCSP is updated and executed during Production and Deployment and Operations and Support.

At the end of its useful life, the system is disposed of in accordance with all legal and regulatory requirements and policy relating to safety, security, space debris, and the environment. These conditions are documented and maintained in the Programmatic Environment, Safety, and Occupational Health Evaluation (PESHE).

Highlight Box 3-6
Operations & Support Products

Data for In-Service Review
Input to CDD for next increment
Modifications / upgrades to fielded systems
SEP
Test Report
System Safety Analysis
Product Support Package

Highlight Box 3-6 shows a list of major program products required at the completion of the O&S phase.

3.3 The PPBS Process

The objective of the PPBS shall be to provide the operational commanders-in-chief the best mix of forces, equipment, and support attainable within fiscal constraints. The PPBS process runs parallel to the JCIDS and DAS processes. It ensures proper oversight over funding of capability acquisition activities. It is an on-going activity that establishes a decision-making framework for planning, programming, budgeting, and execution of future programs as detailed in the DoDD 7045.14. The process allows for periodic review and examination of decisions in the light of changing threats, political environment, economic situation, technological advances, and available resources.

The PPBE decisions are based on a consistent set of objectives, policies, priorities, and strategies derived from National Security Decision Directives. The Directorates plan, program, and budget their proposed tasks to meet their objectives and schedule. The program budget is submitted to the DoD. The DoD forwards the budget in summary to the President. The President's budget then is submitted to the Congress for authorization and appropriation. The Directorate executes on the program as funded.

Key PPBS documents are shown in Highlight Box 3-7. A more detailed description of the PPBS can be found in DoDI 7045.7.

Highlight Box 3-7
Key PPBS Documents

Joint Long Range Strategic Appraisal (JLRSA);
Joint Strategic Planning Document (JSPD);
Defense Guidance (DC);
Program Objective Memoranda (POMs);
Joint Program Assessment Memorandum (JPAM);
Issue Books (IBs);
Program Decision Memoranda (PDMs);
Budget Estimates;
Program Budget Decisions (PBDs);
President's Budget.

4 Systems Engineering Management

4.1 Introduction

SE management process runs parallel to the SE technical process. Its function is to review and assess technical status of the program, balance effectiveness or capability needs against fiscal, technology, or timeliness constraints, and provide direction for further development to minimize mission risk. The DoD 5000 series of acquisition directive and instructions provide the program decision makers substantial flexibility, depending on the maturity of the technology and other risk factors as well as the urgency of the military need. SE management offers integrated technical processes to define and balance system performance, cost, schedule, and risk within a systems-of-systems context. It is embedded in program planning and is meant to support the entire acquisition life cycle.

Each SMC program defines their business model and business and technical approach to meet their program objectives, program and technical challenges, organizational structure, as well as program and engineering planning. SE technical process provides full support to define the program objectives, establish a business model, develop program planning and schedules, and define and implement the program. SE must ensure the technical components of the program are appropriately represented in the program plans, program schedules, work breakdown structure, and cost estimates. SE also ensures the timely reporting and integrity of the technical performance and development progress. SE shares in the risk management responsibilities to identify, assess, and propose mitigating actions of technical risks. SE supports the program manager’s problem identification, resolution, and decision making processes.

The SE Management function has the responsibility for the design of the complete system’s architecture. It develops and maintains system requirements and its internal and external interfaces. SE management interacts with all other activities of the SE process as discussed in Chapter 2 under the “Control and Manage” element of Systems Analysis and Control. It integrates the outputs of the other activities, and conducts independent studies to determine which of its alternate approaches is best suited to the application. It is responsible for the conduct of technical reviews and audits. It includes the planning of day-to-day program activities.

The functions of SE management include:

- Planning and management of a fully integrated technical effort necessary to achieve program objectives,
- Instituting and managing all necessary integrated product and process development mechanisms to ensure that the information channels are always open, team activities are coordinated, and the conflicts are resolved in a timely manner at the proper level,
- Ensure that a comprehensive and systematic "lessons learned" database is available to guide the engineering process,
- Provide for the application of a systematic engineering approach for each phase of the program from the concept definition to the design and deployment to the eventual decommissioning of the system,
- Provide mechanisms to control and assess the progress by conducting technical reviews, configuration management, data and product management, interface management, risk management, and test and verification,
- Support analyses, trade studies, modeling and simulation, prototyping, and research to help optimize system design and minimize program risk,

- Support development of all necessary methods, processes, and data products to ensure that the system can be built, tested, deployed, operated, supported, and properly disposed of at the end of life, and
- Exchange all necessary data and information with the project management to assist decision process at both the system and the program level.
- Assure that the program and its components are implemented within the overall DoD joint enterprise construct, and that the system is capable of interoperating with current and future enterprise assets.

The success of the SE management can be measured by the completeness and accuracy of the decision database and the degree of balance among capabilities, cost, schedule, and risk in the system solution. The decision database includes:

- Trade-off and other analyses,
- Requirements and requirements allocations,
- Specifications,
- Verification requirements, and
- All the decisions made to arrive at the design,
- The design, and
- Traceability of design features to imposed specifications, requirements, constraints, and standards.

The balanced system solution meets all the final requirements and is one for which all driving design decisions were made by Government or Contractor managers at a level that encompassed all products and factors affected by the decision based on comprehensive trades of cost, schedule, and risk.

DAS process requires that SE management prepare a SEP. The SEP describes the program's overall technical approach, including key technical risks, processes, resources, metrics, and applicable performance incentives. It also details the timing, conduct, and success criteria of technical reviews. The SEP supports the Technology Development Strategy (TDS) at Milestone A, and the acquisition strategy for later Milestones. Other DAS components of the SE management include technical reviews, configuration management and control, management of risk, management of interfaces and interoperability in an SoS/Enterprise environment, data and data rights management, ESOH and programmatic ESOH evaluation (PESHE) including National Environmental Policy Act (NEPA), corrosion prevention and control, and spectrum supportability.

The classical management tasks include planning, organization, staffing, top-level direction, project monitoring, and control of resources and schedule used to produce desired capability for the customer at affordable cost. These tasks must usually be carried out iteratively and in close cooperation with the SE organization as the system to be acquired is better defined, especially given the complexities of DoD acquisition programs. In most cases, the distinction between the program and the SE management is blurred. While traditionally, the program offices at SMC perform managerial duties on a project or program, many of these activities may be delegated to support-contractors and/or the prime system contractor through one or more contracts. The allocation of responsibilities between the Program Office, the prime Contractor, and the support-contractors varies from program to program. The program management activities include:

- Program planning based on integrated master plan and other associated program phases, milestones, and forecasts,
- Estimation and management of cost, technology, and schedule, and to monitor program activities and trends,
- Procurement of necessary materials, data, and services to ensure smooth running of the program,
- Assessment, management, development of policy, and implementation procedures to minimize program risk,

- Configuration management, through Configuration Control Board (CCB) process, to control technical baseline and to assess,
- Change management to assess change proposals and their impact on technical baseline, and to plan and budget for change, and
- Contract monitoring, control, and accounting of vendor activities and deliverables by devising proper acceptance procedures.

The primary function of organizations at SMC is to fulfill the program management function for its various projects to improve Warfighter effectiveness. The SE process, described in this handbook, governs the technical effort on the program as a contributory process to facilitate the program management process. The program director, usually a government functionary, is responsible for the implementation of both the program management and the SE processes. SPO manager holds Program Office personnel responsible and delegates to them certain authority (1) to ensure that the technical requirements in the contract accurately reflect the capabilities to be provided based on the decisions of the program Milestone Decision Authority and are complete and verifiable and (2) to monitor the Contractor's progress. Using a legally binding contract, the program director also holds the contractor program manager responsible to meet all the requirements of the contract to include the technical requirements.

Within the program office as well as within the Contractor's organization, it is important to distinguish between the SE process and the SE organization. Although the majority, of an organization has responsibilities associated with implementation of the SE process. Only a few organizational entities have "Systems Engineering" in their titles. For example, in an organization implementing IPPD, teams within the Program Office and the Contractors organization with names along the lines of Systems Engineering and Integration Team (SEIT) may be directly responsible to the Government program director/program manager and the Contractor program manager, respectively. The SEITs may be held responsible for day-to-day management of the overall process as well as conducting certain tasks such as allocation of the system level requirements to the teams responsible for the products at the next lower level in the product tree. Lower tier SEITs (or individuals) may have the analogous responsibilities to the corresponding integrated product team leaders or similar organizational entities at lower levels.

4.2 Current Management Practice

4.2.1 Six Sigma

Employed at many space and military contractors, Six Sigma started as a statistical method to control variation or defects in manufacturing. Six Sigma today is used as an all-encompassing business performance methodology. Developed at Motorola, Inc. in 1986, Six Sigma can be thought as[21]:

- **A metric –** Sigma is used as a scale for levels of goodness or quality. Six Sigma equates to 3.4 defects per one million opportunities.
- **A methodology –** for business improvement that focuses an organization on understanding and managing customer requirements (capability need or opportunity), aligning key business processes to achieve those requirements (analyze opportunity), utilizing rigorous data analysis to minimize variation in those processes (measure performance), and driving rapid and sustainable improvement to business processes (improve and control performance)
- **A management system –** that ensures process metrics and structured methodology are applied to improvement opportunities directly linked to organizational strategy. It is a top-down high performance solution to help programs align their business strategy to critical improvement efforts, mobilize teams to

---

[21] Based on work from Motorola University, Six Sigma training and consultancy division

attach high impact projects, accelerate improved business results, and govern efforts to ensure improvements are sustained

Inspired by Deming's Plan-Do-Check-Act cycle, Six Sigma methods for improving an existing business process, known by its acronym DMAIC, has five phases:

- **Define** the problem, the voice of the customer, and the project goals, specifically.
- **Measure** key aspects of the current process and collect relevant data.
- **Analyze** the data to investigate and verify cause-and-effect relationships. Determine what the relationships are, and attempt to ensure that all factors have been considered. Seek out root cause of the defect under investigation.
- **Improve** or optimize the current process based upon data analysis using techniques such as design of experiments, poka yoke or mistake proofing, and standard work to create a new, future state process. Set up pilot runs to establish process capability.
- **Control** the future state process to ensure that any deviations from target are corrected before they result in defects. Implement control systems such as statistical process control, production boards, visual workplaces, and continuously monitor the process.

Six Sigma method for creating new product or process, Design for Six Sigma, also has five phases:

- **Define** design goals consistent with stakeholder needs and the enterprise strategy.
- **Measure** and identify product capabilities and risks.
- **Analyze** design alternatives, evaluate design and perform trade studies to select the best design.
- **Design** details, optimize design to target values, and plan for design verification.
- **Verify** the design and implement the production process

Six Sigma Management System leverages the foregoing methods to drive clarity around the business strategy and the metrics that most reflect success with that strategy. It provides the framework to prioritize resources for projects that improve the metrics, and it allows SE managers to orchestrate the efforts for rapid, sustainable, and improved business results.

4.2.2 Lean Management

Lean[22] offers a principled approach to help an organization systematically identify and eliminate waste from its business processes. It is a tightly focused process aimed at delivering value to the stakeholder. Although Lean originated in Toyota's manufacturing operations, the tools have been successfully applied in organizations across all sectors. It is of note that product development processes are quite distinct from factory manufacturing. There is certain uncertainty in engineering processes since the exact design is not known whereas the factory is ideally asked to produce a well-defined part or component. Furthermore, for product development the raw material is information and the end product is typically a specification on how to build a product.

Activities that do not contribute to value as defined by the customer are considered waste and are candidates for elimination. The value to the customer can be quantified in terms of product

- performance and quality required to satisfy capability needs,
- schedule or timeliness of availability, and the

---

[22] James Womack, Daniel Jones, and Daniel Roos coined the term "Lean" in their 1990 book The Machine that Changed the World to describe Toyota Motor Company's manufacturing paradigm (Toyota Production System)

- cost of ownership.

Highlight Box 4-1 shows a number of useful value metrics for space systems. Value stream mapping (VSM) in lean is a fundamental tool used to identify waste. It starts with customer needs as stated in requirements, schedules, and cost. It maps out all end-to-end linked actions, processes and functions necessary for transforming inputs (materials, information) to outputs (product, service) to help identify and eliminate waste. As stated earlier, any activity that does not contribute to customer value is waste. VSM encompasses both material and information flows, as they move toward product or service delivery. VSM makes the search for non-value-added activities intentional and deliberate. Non-value-added activities can be pure waste that are candidates for elimination. However, some non-value-added activities can be necessary even though they do not directly contribute to value. Figure 4-1 from MIT's Lean Aerospace Initiative (LAI) company studies shows a major part of effort and time in engineering and product development is wasted.

As waste is minimized, workflow moves from value-added activity to value-added activity, transforming materials and information to achieve the end result.

Highlight Box 4-1
Value Metrics
(Examples from Space Acquisitions)

Performance:
- LV lift capability,
- SV orbit characteristics,
- Downlink/Uplink data rates
- Component MTBF or MTTF,
- Interoperability characteristics

Schedule:
- Acquisition timeline,
- Product development cycle time,
- Time to repairs

Cost:
- Development cost,
- Production cost,
- Maintenance (recurring) costs,
- Upgrade costs,
- Disposal costs

Lean offers a holistic approach to business. The concept of pull makes business operations transparent to all stakeholders: customer, community, government, company, supplier, operator, and user. Lean enterprise pull engages all stakeholders. Everyone knows the objective, as there is no lean without a fully informed community. People understand their job and know how it contributes to achieve the objective. It fosters cooperation and creativity through established policies that promote stable enterprise-wide relationships. It encourages learning and avoids "blame game" for performance errors typically caused by processes. It establishes a horizontal organizational structure that empowers people and encourages decision-making at the point of knowledge and the need to minimize expensive rework later in product development cycle.

Lean pursues perfection. A well-informed enterprise community is empowered to strive for continuous improvement in processes, products, and people.

In short, fundamental objectives of lean are to:

- Minimize waste
- Strive to have the right thing at the right place and at the right time
- Be responsive to change
- Maintain and nurture effective skills and relationships
- Continuously improve processes and products

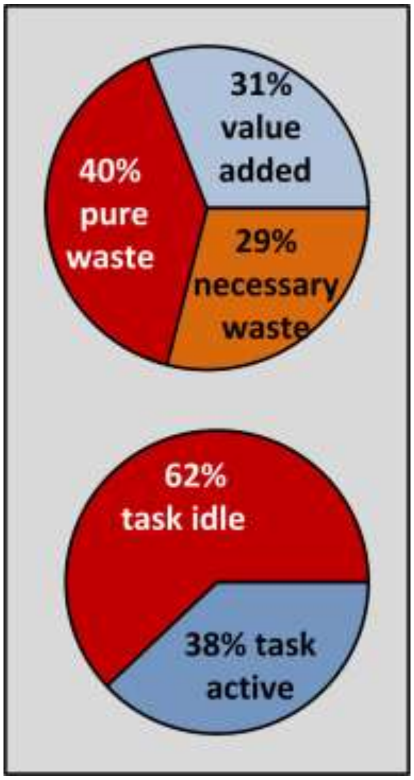


Figure 4-1 Waste in engineering and product development (Source: MIT's LAI company studies)

- Deliver exactly what the customer wants (performance, schedule, cost)

4.2.3 Adapting Lean and Six Sigma for Systems Engineering

Most enterprises today base their transformation initiatives on elements of Lean and Six Sigma. Lean delivers value to all stakeholders by optimizing VSM and flow to eliminate waste, and strives toward perfect quality through holistic and evolutionary improvement. Six Sigma minimizes defects to increase customer satisfaction by process centric and continuous quality improvement through the elimination of variation using quantitative methods in all enterprise processes.

SMC can benefit from a holistic view of the enterprise. A flat supply chain with adequate pull and knowledge sharing can help identify technology and other issues in time to identify workarounds or a reassessment of requirements. Recent attempts in the aerospace industry, notably from the MIT's Lean Aerospace Initiative (LAI), a consortium of government and industry leaders, has shown that application of Lean Six Sigma methods can greatly improve cost and performance of space programs. They have also developed tools and other guidance to implement lean six sigma for its member organizations.

Emphasis on elimination of non-value-added steps and continuous improvement based on quantitative analyses can help reduce waste and save cost. Data driven decisions provide an insight into the cost of requirements, tempering capability need reassessment among all stakeholders.

Today's space systems are complex and must interoperate with other existing and future complex systems. Lean Six Sigma SE places emphasis on environment where people are encouraged to share, learn, and cooperate can help balance the needs of stakeholders across traditional boundaries. Highlight Box 4-2 shows key contributions of Six Sigma and Lean to SE management process.

Highlight Box 4-2
Lean Six Sigma in SE
(Process improvement concepts at SMC)

Six Sigma:
- Data driven decisions in requirements analysis and design
- Quantitative methods to eliminate defects in design
- Prescriptive infrastructure for repeatable results

Lean:
- Focus on increasing the velocity of the processes by VSM to identify and eliminate unnecessary steps (waste)
- Define and refine stakeholder value flow
- Think of the system at hand as an integral part of the enterprise to add (emergent) value for stakeholders
- Holistic and flat infrastructure to foster cooperation and creativity over the entire supply-chain (enterprise view)
- Focus on value to the stakeholder

And, of course, building the perfect system first time cannot be overstressed in the space environment.

4.3 Managing Cost, Schedule, and Risk

It is the responsibility of SE to provide the tools, analyses, and technology trades required to help decision-making by balancing the desired user capabilities against the program cost, schedule, and risk. In addition, the overall program cost, schedule and risk reflect the technical plan and technical execution of the plan for the program. Essential SE management tasks that may be assigned to Government or Contractor organizations, include (i) verification that the design provides the needed capabilities (or meets the requirements), (ii) estimation of all elements of program cost, (iii) monitoring adherence to the schedules, and (iv) assessment of risk. Stated a different way, the assessment of all those factors is essential to monitoring the implementation of the SE process on the program and the contract(s).

Earlier, the Government management systems for establishing capabilities (the Capabilities/Requirements Generation System), for overseeing the acquisition programs (the Defense Acquisition System), and for establishing the budget (Planning, Programming, Budgeting, and Execution, PPBE, System) were described.

Other Government agencies also provide key data to the program including the threat assessment provided by the intelligence community and environmental data/phenomenology from a variety of laboratories and agencies. The risk to the program is minimized when the requirements derived from the capabilities-based assessment, direction given by DAS process (including acquisition strategy, technologies, and schedule), and the budget approved by the PPBE system are balanced. Typically, the relationship between these factors is as shown in Figure 4-2.

The concept selection is usually made during a program phase prior to detailed design. The environmental constraints are then predicted and the threat is then assessed based on the concept selected. The SE process prepares the contract requirements accordingly. The design that complies with the contract requirements then follows from the SE process. Cost, schedule, and risk are all consequences of the development, verification, manufacture, deployment, support, and disposal of the design – none can be predicted with any certainty until the basic parameters of the concept and its design are understood. In other words, a different design will result in a different cost, schedule, and risk. Furthermore, the relationship between cost and the budget is a significant contributor to the risk – if the predicted cost rises above the budget, the risk obviously increases a pace.

It should be clear, therefore, that the SE process has to interact closely with the JCIDS requirements generation or the capabilities needs assessment, the DAS system lifecycle process, and the PPBE to plan and balance capabilities, cost, schedule, and risk. In a program where such interactions are not effective, cost growth and schedule slippage are more common leading, at times, to program cancellation.

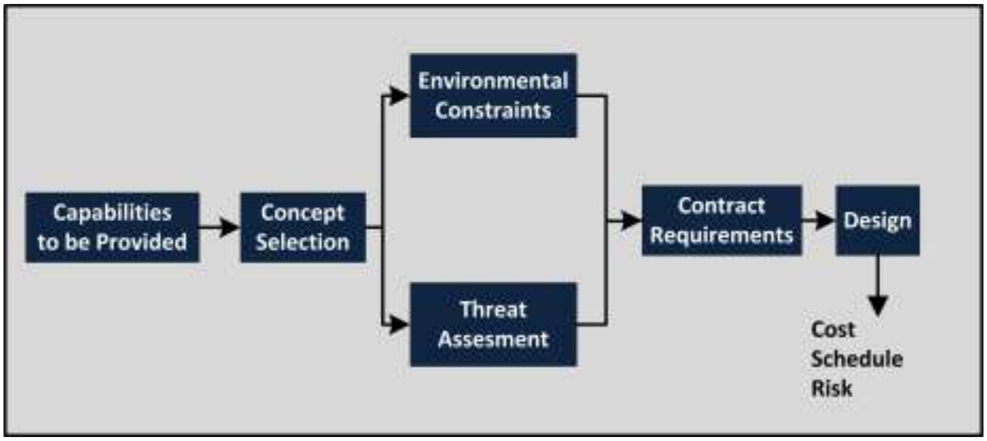


Figure 4-2 Typical relationship of capabilities and other program inputs to cost, schedule, and risk

To help understand the evaluation of capability (or performance), cost, and risk, later subsections of this Chapter address systems analysis, cost estimating, and risk management.

4.4 Planning and Organizing

The steps in planning and organizing for SE include the following:

- Selection of a proven process and the tailoring of that process to the next phase of the program life cycle to include the processes for risk management, interface management, configuration management (CM), and data management (DM),
- Assigning responsibilities for implementing the process,
- Outlining the work via the product-oriented Work Breakdown Structure (WBS),
- Defining the scope of the work via the Contract Statement of Work (CSOW),
- Structuring the next program phase to include the selection of major events such as reviews and audits,

- Establishing an organization to carry out the work (such as Integrated Product Teams or IPTs for each major product or work area in the WBS),
- Identifying what must be accomplished by each major event (such as in an Integrated Master Plan or IMP),
- Scheduling the tasks to achieve complete each major event (such as in an Integrated Master Schedule or IMS), and
- Planning and authorizing the detailed work/work packages to complete each task (such as in an Earned Value Management System or EVMS).

In most programs, the first and third items in the above list are specific to the SE process and its output and will be treated next. The remainder are usually conducted in an integrated fashion for all work and organizational elements and heavily tailored to both the management philosophy and the objectives of the next program phase so only a few additional points will be made in the subsequent discussions.

4.4.1 Systems Engineering Process Selection

Selecting a proven process is the critical first step described above. Considerable attention has been given to process development since the early 1990s starting with the publication of the draft MIL-STD-499B in 1994 that details requirements for both Government Program Offices and Contractors. Soon after, two standards-issuing organizations, the EIA and IEEE, issued standards based heavily on the draft MIL-STD-499B (EIA/IS-632 and IEEE1220). Subsequently, both EIA and IEEE issued standards more attune to the general industrial setting, i.e., not specific to Government contracting. These were ANSI/EIA-632-1998[23] and IEEE-1220.[24] Since then, many industrial firms including defense contractors have put in place corporate processes based on one or the other of these standards. It is important to note that the Program Office cannot enforce compliance with such corporate processes unless such is required by the contract.

4.4.2 Systems Engineering Plan (SEP) and Systems Engineering Management Plan (SEMP)

As discussed earlier, the SE process and responsibilities for its implementation are usually described in a SEP for the program. Its counterpart, the SEMP, is typically produced by the contractor.

All required technical specialties should be addressed as an integrated part of the SE process. At times, some of these are covered in separate plans; when this happens, the SEP or SEMP should show how they are integrated with and support the overall technical effort on the program. To support review of the Contractor's plans and programs in those areas, Risk Management, Interface Management, Configuration Management, Data Management, and Operational Safety, Suitability, & Effectiveness are addressed in separate subsections below. Still, other specialties are covered in Chapter 6, while verification and validation are covered in Chapter 7.

4.4.3 The Work Breakdown Structure

The WBS is a means of organizing system development activities based on system and product decompositions. It is a product-oriented family tree composed of hardware, software, services, data, and facilities, which result from SE efforts during the development and production of the system and its components, and which completely defines the program. The WBS is prepared from both the physical and system architectures, and identifies all necessary products and services needed for the system. This top-

---

[23] ANSI/EIA-632-1998, Processes for Engineering a System, available from Global Engineering Documents, 1-800-854-7179.

[24] IEEE Std 1220-1998, IEEE Standard for Application and Management of the Systems Engineering Process, The Institute of Electrical and Electronics Engineers Engineers, Inc., New York.

down structure provides a continuity of flow down for all tasks. Enough levels must be provided to properly define work packages for cost and schedule control purposes.

Since the WBS is a derivative of the physical and systems architectures, it is a direct output of the SE process. It can also be considered part of the synthesis process since it helps to define the overall system architecture. The DSMC Systems Engineering Fundamentals Book, December 2000, includes the WBS in the System Analysis and Control process as a tool to help represent and control the overall process. The WBS is thus not just about hardware or software but also is used to structure development activities, identify data and documents, organize integrated teams, and is used for non-technical program management purposes such as scheduling, and measurement of progress.

The WBS defines the total system of hardware, software, services, data, and facilities, and relates these elements to each other and to the end products. Though WBS is a product of the SE process, it impacts costing, scheduling, and budgeting professionals as well as contracting officers. An integrated effort including these stakeholders should be applied to develop the program WBS and monitor its application in the contract WBS.

WBS and WBS examples of a space system are discussed in Chapter 5.

4.5 Resource Management

Day-to-day monitoring of the Contractor's progress is by comparing progress against the plans and schedules. The IMP, IMS, and EVMS can be particularly effective for this purpose. Though formal EVMS reports can be a lagging indicator, the contractor may collect and be able to make available data that is timelier. For example, resources such as total manpower are usually available for a given week by early in the following week. Manpower levels higher than planned, especially if part of a trend, can be an indication of a technical problem. Levels lower than planned can be an indication of a staffing problem.

4.5.1 Staffing and Direction

Staffing the Program Office is primarily a responsibility of the Air Force manpower and personnel systems. Direction for the program usually comes in the form of decision memoranda approved by the Milestone Decision Authority for the program and program direction memoranda from the Air Force.

Staffing by the Contractor is usually carried out by a human resources function with little oversight needed unless staffing is not as planned or personnel are unqualified. Directing by the Contractor is unique to each corporation, but should be formal. It is often keyed to the Earned Value Management System and includes formal authorization to open or close work packages.

4.5.2 Earned Value Management (EVMS)

Earned value is a management technique that relates resource planning to schedules and to technical cost and schedule requirements. All work is planned, budgeted, and scheduled in time-phased "planned value" increments constituting a cost and schedule measurement baseline. There are two major objectives of an earned value system: to encourage contractors to use effective internal cost and schedule management control systems; and to permit the customer to be able to rely on timely data produced by those systems for determining product-oriented contract status.

The benefits to project management and systems engineers of the earned value approach come from the disciplined planning conducted and the availability of metrics, which show real variances from the plan. The values of the variances are the metrics indicators that may corrective actions.

A detailed discussion of the EVM is provided in Chapter 5.

4.5.3 Reviews and Audits

Requirements reviews, design reviews, and configuration audits provide an opportunity to assess program status in considerable detail. See SMC Standard, SMC-S-021, "Technical Reviews and Audits for Systems, Equipment and Computer Software," and System Engineering Critical Process Assessment Tool (CPAT) Handbook for details on reviews and audits and their expected entry and exit criteria.

4.6 Metrics and Measurement Assessments

Measurements can add value to improving program performance and risk assessments, mitigations and reporting. Typically, a well thought out measurement program is based on the objectives and goals of the program. Appropriate metrics are then attributed to each goal. For example, a SE goal is to establish clear traceability of system requirements to acceptable sources as well as reverse traceability to ensure that all source requirements are being captured. In fact, this goal is only attained when 100% of the system requirements have (funded or mandated) sources and 100% of the source requirements are sufficiently captured in the defined system. The systems engineer may be required to maintain an accounting of progress to meet this goal. Two possible measurements that the systems engineer may be required to periodically report may be the percent of source requirements that are defined (trace to the system) and the percent of system requirements that have sources. For this example, we have applied the metrics development and reporting process represented in Figure 4-3. Often, management is more interested in overall progress and not so much detailed measurements. For instance, the engineer may be required to report requirements development progress in terms of maturity levels. It can easily be predetermined the requirements development maturity is based on a set of factors such as traceability, allocations, supporting analyses and trades, verifiability, etc. Hence, the systems engineer collects a set of measurements. Then based on the predefined definition of maturity levels, he/she reports a roll-up maturity metric. Surely, management will want to see whether progress is being made so the systems engineer also provides the previous month's maturity metric as well.

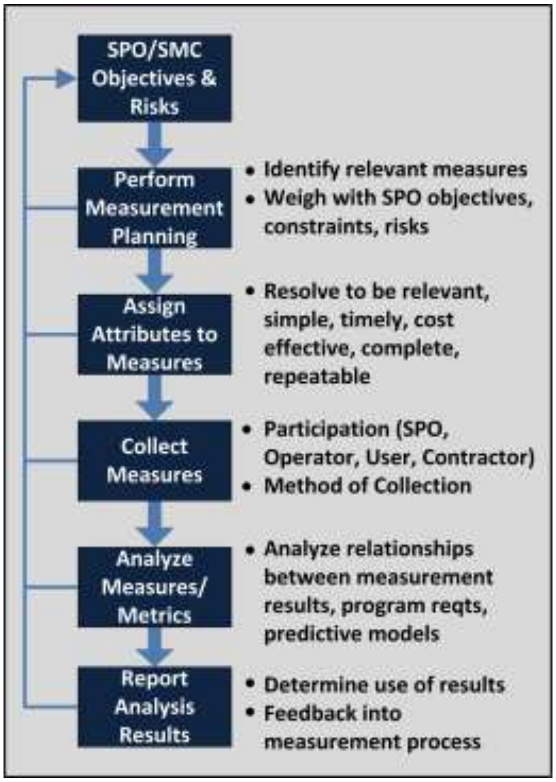


Figure 4-3 Metrics development and reporting process

4.6.1 Technical Performance Measurements (TPM)

Technical Performance Measures (TPMs) provide an assessment of key capability values in comparison with those expected over time. TPM is an evolutionary program management tool that builds on the two traditional parameters of Earned Value Management and cost and schedule performance indicators. A third dimension is also added – the status of technical achievement. By combining cost, schedule, and technical progress into one comprehensive management tool, program managers are able to assess the progress of their entire program. TPMs are typically established on those programs complex enough where the status of technical performance is not readily apparent. TPMs can also be valuable for risk monitoring – levels below that forecast can indicate the need for an alternate approach.

A more detailed discussion of TPMs is provided in Chapter 5.

4.6.2 Trade Study Process

Trades are performed throughout the concept definition, development, and design phases to select operational concepts, originating capabilities and requirements high level system architecture, systems functions and requirements, and design solutions. For space systems the focus of trade studies is to perform objective trade comparisons of all reasonable alternatives and to choose the alternative that best balances performance, cost, schedule, and risk. (We might add safety, reliability, weight, and other constraints.) Also for space systems, the trade study process is often controlled using models.

A detailed discussion of the trade study process is provided in Chapter 5.

4.6.3 Cost Estimating

Figure 4-4 taken from a GAO report shows that the relationship between the cost and the required capabilities for SMC space projects is not well understood. It Includes data from Advanced Extremely High Frequency (AEHF), Global Broadcast System (GBS), Global Positioning System (GPS) II and III, Mobile User Objective System (MUOS), Space Based Infrared System (SBIRS), and Wideband Global SATCOM (GBS). A disciplined and careful approach is needed to estimate cost and schedule to avoid program funding risk.

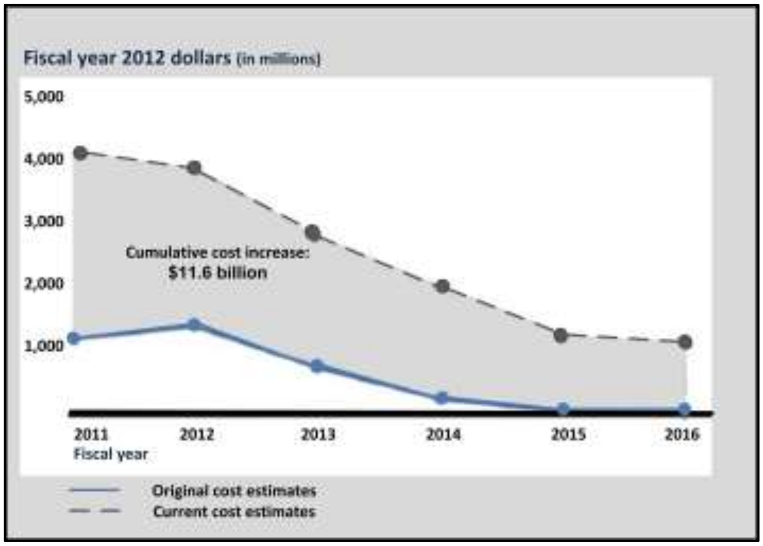


Figure 4-4 Comparison between Original Cost Estimates and Current Cost Estimates for Selected Major Space Acquisition Programs for Fiscal Years 2011 through 2016. (Source: GAO analysis of DoD data)

In any SE selection process, reliable cost estimates are critical in avoiding expensive design solutions. There are presently several commercially available cost models that give fairly accurate relative hardware and software cost indications of competing approaches with even the most fragmentary design information. These models have been supplemented with more customized models developed by individual organizations and aimed at the types of systems with which they have specific interest. Most models require some training in their use and experience in interpreting results. While there is much disagreement on their absolute accuracy in predicting costs, models are especially useful to Systems Engineers in establishing relative costs in order to choose between candidate approaches. Running several models and then comparing outputs can increase confidence in model results.

Cost estimators can provide meaningful results soon after candidate system architectures begin to emerge. As the designs firm, models become less important and the estimating function turns increasingly to those in manufacturing versed in process and materials estimating. The SE should be aware of this transition. As the development phase of a project ends and EMD begins, cost estimates should be firmly based on actual cost data.

4.7 Risk Management

Risk is a measure of the potential inability to achieve overall program objectives within defined cost, schedule, and technical constraints and has two components: (1) the *probability/likelihood* of failing to achieve a particular outcome, and (2) the *consequences/impacts* of failing to achieve that outcome. Risks or potential problems are items that may occur in the future.[25]

4.7.1 Risk Management Process

Risk management[26,27] is an important and often critical activity for DoD systems acquisition. It is the act or practice of managing risk. Risk management includes planning for risk, assessing (identifying and analyzing) risk areas, developing and implementing risk-handling strategies, monitoring risks to determine how risks have changed, and documenting the overall risk management program. An example risk management process is shown in Figure 4-5.

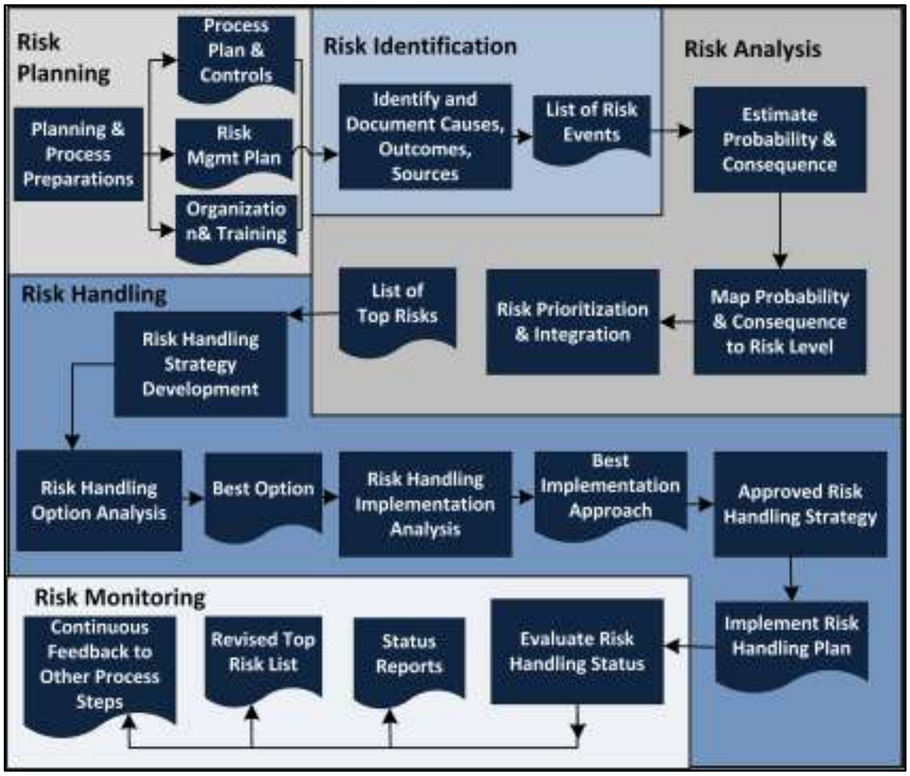


Figure 4-5 Example Risk Management Process

Risk planning is the process of developing and documenting an organized, comprehensive, and interactive strategy for identifying and analyzing risks; developing risk handling plans; and monitoring how risks have changed. A key risk planning output is the Risk Management Plan (RMP). Typically, the risk manager (or equivalent) working with SE and program management personnel will perform the initial risk planning and develop the RMP. The program manager should review and approve the RMP. Risk management training should be performed after the release of the RMP.

Risk identification is the process of examining the program areas and each critical technical process to identify and document the associated risk. A variety of risk identification approaches exist that are suitable for examining different types of risk—two are mentioned here. One approach is based upon the program

---

[25] Department of Defense, Risk Management Guide for DoD Acquisition, Defense Acquisition University, Fifth Edition, Version 2.0, June 2003.

[26] Ibid.

[27] Edmund H. Conrow, Effective Risk Management: Some Keys to Success," Second Edition, American Institute of Aeronautics and Astronautics, July 2003.

WBS and used to evaluate elements/products. A second approach evaluates key processes (e.g., design and test) and is often best applied midway in the development process.

Risk analysis is the process to examine identified risks, isolate causes, determine the relationship to other risks, and express the risk in terms of probability of occurrence and consequence of occurrence. It also includes risk rating and prioritization.

in the example that follows). The resulting probability and consequence values are then transferred to a risk mapping matrix (typically 3x3 or 5x5) which converts the values to a risk level. For cost or schedule risks, a Monte Carlo simulation is typically used to estimate cost or schedule risk at the desired confidence level (e.g., 70th percentile). The RMB then prioritizes risks across the program based upon the estimated risk levels. They may also use the frequency of occurrence, time to impact, relationship with other risks, and other considerations in performing the prioritization.

Risk handling is the process that identifies, evaluates, selects, and implements strategies in order to reduce risks to acceptable levels given program constraints and objectives. This includes the specifics on what should be done, when it should be accomplished, who is responsible, and implementing the associated cost and schedule.

For those risks authorized by the RMB (e.g., medium and higher), the risk focal point develops a Risk Handling Plan (RHP) with the assistance of the IPT lead and others. The RHP includes a primary and sometimes one or more backup risk handling strategies. A structured method is used to first select the most desirable handling option (from assumption, avoidance, control, and transfer), then choose the best implementation approach for that option. Suitable metrics should also be selected and included in the RHP to allow subsequent tracking of results as part of risk monitoring. The RMB approves the RHP and ensures that the IPT has sufficient resources to implement it.

Risk monitoring is the process that systematically tracks and evaluates the performance of risk handling actions against established cost, performance, schedule, and risk metrics throughout the acquisition process. This permits an evaluation of actual vs. planned progress in reducing risks to an acceptable level in accordance with implemented risk handling plan activities, budgets, schedules, resources, testing, etc. Risk monitoring also provides information to update risk handling strategies, risk analyses, risk identification, and risk planning as warranted through continuous feedback to the other risk management process steps (see Figure 4-5). Monitoring results may also provide a basis for reducing a risk to watch list status or closing a risk if the level is at an acceptable level.

Risk documentation includes recording, maintaining, and reporting risk planning, identification, analysis, handling, and monitoring results. It includes all plans, reports for the program manager, and reporting forms.

4.7.2 Operational Risk Management

Operational risk management (ORM) is a decision-making process to systematically evaluate possible courses of action, identify risks and benefits, and determine the best course of action for a given situation. ORM is a subset of project risk management and includes risk identification, risk analysis, and risk handling process steps, but no formal risk planning or risk monitoring steps[28,29]. While this may be suitable for a number of operational situations, the acquisition risk management process previously described is more suitable for the development and production of space and other systems.

---

[28] Ibid.

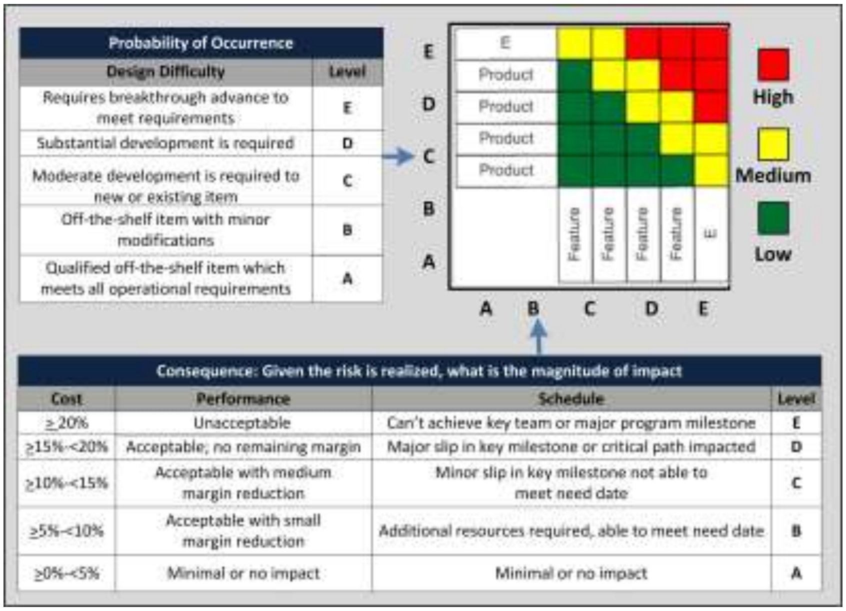
[29] Department of the Air Force, Air Force Pamphlet 90-902, Operational Risk Management (ORM) Guidelines and Tools, 14 December 2002.

4.7.3 Example for Implementing Risk Management

An initial task would be to prepare a RMP tailored to the program. An outline of a risk management plan based upon the Department of Defense "Risk Management Guide for DoD Acquisition" is included in Appendix C3, Example Risk Management Plan Outline. The RMP, among other things, identifies risk identification and analysis approaches and methods to be used. SMC does have a few tools that can be of benefit to establish and maintain a risk management process. See Chapter 5 on Tools for further discussion on risk management tools.

A simple risk analysis methodology is now presented that can be applied to a variety of technical risks. A single 5-level ordinal probability of occurrence scale, related to design difficulty is presented in Figure 4-6. Three 5-level consequences of occurrence scales for cost, performance, and schedule are given in Figure 4-6 Technical risk will generally encompass a number of additional risk categories in addition to design, such as: manufacturing, support, technology, threat, etc. Hence, the number and types of probability scales used for a technical risk analysis must be tailored to your program. Similarly, while cost, performance, and schedule are the appropriate consequence categories, the scale levels must also be tailored to your program.

For this example, assume that a digital data processor is being developed for a space application. This processor includes an architecture based upon an existing design that is deployed in a different operational environment than envisioned for the new application. A moderate level of development is anticipated to transform the processor design from the existing environment to the new environment. From Figure 29, this corresponds to a probability level=C[30]. It is estimated that a potential cost growth of 14% may occur due to design changes needed because of performance limitations of some existing parts for the required operating environment. Brass-board testing revealed that while the new processor would likely pass all performance tests, a small reduction in margin will be present in the flight units. The proposed schedule to deliver a flight processor can be achieved if additional resources are provided. In this example, cost consequence level = C, performance consequence level = B, and schedule consequence level = B. The conservative approach is to select the maximum value associated with the probability scales used and the three consequence scales. Thus the resulting probability and consequence values are levels C and C, respectively.



Probability of Occurrence	
Design Difficulty	**Level**
Requires breakthrough advance to meet requirements	E
Substantial development is required	D
Moderate development is required to new or existing item	C
Off-the-shelf item with minor modifications	B
Qualified off-the-shelf item which meets all operational requirements	A

	A	B	C	D	E
E	🟨	🟨	🟥	🟥	🟥
D	🟩	🟨	🟨	🟥	🟥
C	🟩	🟩	🟨	🟨	🟥
B	🟩	🟩	🟩	🟨	🟨
A	🟩	🟩	🟩	🟩	🟨

🟥 High

🟨 Medium

🟩 Low

Consequence: Given the risk is realized, what is the magnitude of impact			
Cost	**Performance**	**Schedule**	**Level**
≥20%	Unacceptable	Can't achieve key team or major program milestone	E
≥15%-<20%	Acceptable; no remaining margin	Major slip in key milestone or critical path impacted	D
≥10%-<15%	Acceptable with medium margin reduction	Minor slip in key milestone not able to meet need date	C
≥5%-<10%	Acceptable with small margin reduction	Additional resources required, able to meet need date	B
≥0%-<5%	Minimal or no impact	Minimal or no impact	A

Figure 4-6 Example technical risk analysis methodology

The next step is to convert these probability and consequence values into risk. This is performed using a risk mapping matrix such as the example matrix given in Figure 4-6[31]. For

---

[30] Letters are provided for scale levels instead of numbers to discourage readers from attempting to perform mathematical operations on the scale values. Numbers may be used but because the risk scales are ordinal rather than cardinal and the true coefficients are unknown, the resulting values are placeholders and have no numerical meaning other than being rank-ordered.

[31] Note: The risk assessment matrix can typically be as simple as a 3x3 or as large as a 5x5. Furthermore, which blocks in the risk matrix are low, medium or high is a matter of discretion. Whether or not the matrix is asymmetrical or symmetrical depends upon utility preferences of program decision makers. It is recommended that a symmetric matrix be used unless specific, quantitative information (rather than uncertain guesses) exists to develop asymmetrical boundaries.

the digital data processor example given above, corresponding risk level is medium. Finally, because the estimated risk level is medium, a Risk Handling Plan (discussed previously) should be developed and implemented.

In closing, successful acquisition programs will generally include the following regarding risk management:

- Integrate risk management into the program's culture, and do not treat risk management as an add-on, "check the box" or parallel activity;
- Develop an acquisition strategy consistent with the program risk level;
- Include industry, government program office, user, and other key stakeholder participation in risk management;
- Obtain risk management buy-in at all program-levels from working-level personnel through upper management and stakeholders;
- Establish the means and format to communicate risk information and to train participants in risk management;
- Use technology demonstrations/models/simulations and prototypes to assist in reducing risk;
- Develop and follow a program-specific Risk Management Plan;
- Identify candidate risks early and manage intensively those design parameters that substantially affect cost, performance, schedule, and risk;
- Evaluate program risk using a structured, iterative, continuous process; and
- Use test and evaluation as a means of quantifying risk handling results.

4.8 Interface Management

Interface and the types of interfaces between entities with examples are provided in Chapter 1. Interface management is a SE activity that begins in parallel with the development of architectures and continues for the life of the program. The evolutionary process to sufficiently design interfaces (DI) begins with concept definition and continues through the development and design process. Interfaces, both internal and external to the program, are documented and managed formally. This activity is intended to ensure compatibility among subsystems being designed and fabricated, and to ensure future interoperability between systems.

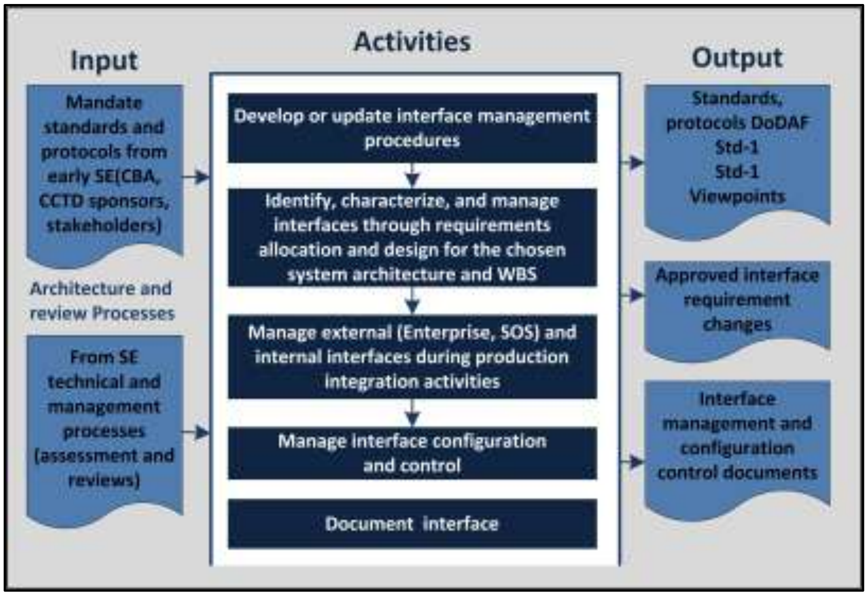


Figure 4-7 Major activities with inputs and outputs for interface management

Major activities with inputs and outputs for interface management are shown in Figure 4-7. Interface requirements from early SE, foundational, and stakeholder data are used as inputs. An on-going inter- and intra-system SE technical and management review and assessment process provides further inputs as negotiated and necessary changes to the interface.

Internal interfaces are those boundaries between subsystems and products that are controlled by the project as dictated by the system architecture and design. External interfaces are the boundaries between a system end product and another external system end product or a human and the operating environment in which the system products are used or operated.

To assist in managing systems development efforts when the efforts are divided between contracts, government programs, and geographically diverse teams within an organization, a formal interface management and control system is set up. The structure of an interface control system is influenced by the system and subsystem WBS, Contracts/subcontracts, interoperability requirements with other systems.

interface management activities include (i) preparation of interface management procedures, (ii) interact with architecture and design activities to assure and enforce proper interface, (iii) manage interface during implementation and integration, (iv) verify, validate, and test interface compliance for deployment, and (v) document all interface management activities.

Interface management activities output (i) negotiated and approved ICDs, (ii) negotiated and approved interface change requirements, and (iii) other interface configuration and control data as necessary.

Important activities in interface management include management of interface definition activities that define interface architectures, identify interface requirements/constraints, ensure sufficient trades and analyses support the defined interface requirements, and documenting the interface constraints in interface specifications, interface control drawings or other documents. Furthermore, early in the program, management decisions in consultation with sponsors, stakeholders, and joint-CONOPS owners an appropriate, often mandated, set of open and widely accepted standards are identified and applied to the system external and internal interfaces to assure interoperability with enterprise and SoS in a secure net-centric environment. The Interface manager also ensures sufficient review and approval of the documentation by those stakeholders responsible for affected products. The Interface Manager also manages the change process of preliminary interface engineering products before they are placed under formal configuration control. In summary, the management plan addresses the following aspects of the system over its lifecycle:

- Identification of subsystems or components that require an interface
- Identification of external interfaces to other systems, system of systems, and enterprise
- Identification of interface requirements including scope, design, implementation, integration, testing, and deployment of the system
- Identification of requirements for facilities, organizations including COI, and other assets need to implement and test interfaces
- Identification of necessary interface management and technical skills over the system lifecycle
- Technical strategy for interface architecture, design, and test
- Precise technical specification of the interface as documented in ICDs and protocols
- Configuration and quality management relevant to interface requirements, architecture, design, implementation, integration, test, and deployment
- Establishment of interface development schedules and resources
- Assignment of interface development roles and responsibilities
- Management of interface development risks
- Management of interface safety, hazard analysis, and related activities

4.9 Interoperability Management

The concept of interoperability was introduced in Section 1.6, including DoD definitions that extend typical IT view of interoperability to physical interfaces. Interoperability enables form, fit, or function reuse and interchangeability, as well as cost-effective enterprise operations and seamless availability, integrity, and confidentiality of timely information over disparate entities.

Interoperability is managed as an external interface, generally based on open, vendor-neutral, well-documented, and widely-accepted technical specifications, standards, or protocols. It is an interface that is typically dictated by enterprise or SoS considerations. For example, a standard interoperable interface between an SV and a LV can enable agile production and timely deployment of assets in space.

For typical IT systems interoperability is "the capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units."[32] It is essential to the development, architecture, design, and implementation of net-centric operations. A body of knowledge and guidance on enterprise-wide application of standards like IPv6, RF frequencies, and cyber security in net-centric environments is available through DISR mandates, applicable design profiles, and other DISA produced and managed facilities and guidance. NESI, a joint effort of SPAWAR, SMC, and other organizations also offers an exhaustive collection of information and implementation advice on interoperable solutions.

The programs also produce (or reuse) mechanical, electronic, and organizational interface documents to facilitate connections and integration between major subsystems as well as external systems like the interface between the SV and launch processing facilities and the LV.

In today's capability-driven environment, interoperability requirements impact the SMC systems in three major categories: Mission, Platform, and IT. These categories are summarized below.

4.9.1 Mission Impact

Joint capability-based analyses lead to overarching enterprise or SoS requirements that can help efficient use of DoD resources. As a forcing function, these interoperability requirements ensure that sets of systems to work together to provide a broader or emergent capabilities for the execution of the mission.

Interoperability implementation commitments essential for Warfighter/User mission success extends beyond information sharing to include compatible functional and physical interfaces of the SMC space system with other systems, deployment, operational and support assets, equipment, as well as deployment, operational and support processes, procedures, and organizations to effectively build, test, integrate, deploy, operate and dispose SMC systems

4.9.2 Platform Impact

SMC SVs and supporting ground and user systems deploy a number of common functions and technologies like the SV bus, communications subsystem, or even weapon sensors. Such subsystems, functional HW or SW can be reused or shared. While joint policy and requirements in this area are less developed, major gains in interoperability, effectiveness, and rapid deployment can be made through targeted communication between various SPO management and interaction with appropriate COIs.

---

[32] ISO/IEC 2382-01: Information Technology Vocabulary, Fundamental Terms.

4.9.3 Information Technology Impact

This category of mandates is partly driven by evolving joint mission areas and DoD NR-KPP and interoperability mandates to radically improve information sharing by employing robust modern information technologies. Net-centric operations focus on rapid gathering and sharing of accurate and trusted war-fighting information, thus dramatically improving situational awareness, collaboration, speed of command, and mission effectiveness. In practical terms, this capability relevant to SMC space systems translates into a service-oriented internet-like secure global information grid (GIG) that offers high quality information to the authorized warfighter, when and where it is needed. This net-centric environment allows information to be processed, analyzed and disseminated more quickly than ever before while maintaining data confidentiality, integrity and availability.

SMC programs are required to produce and maintain ISP, IAS, and obtain CCA certification over the lifecycle of the system, as described in detail in the SED handbook.

4.9.4 SE lifecycle view of Interoperability Implementation

DoD mandates (DoDD 5000.01, DoDD 4630.05, CJCSI 6212.01) allow for prescriptive sharing of information by applying IT standards and protocols (logical interfaces) and extend interoperability considerations to incorporate hardware and operational interfaces between intra-system, inter-systems, and beyond to the DoD Enterprise. Interoperability pervades form, fit, and function over disparate systems, services, environments, organizations, activities, processes and procedures, migration strategies, and preplanned upgrades to achieve enterprise interoperability over the life cycle of a system. In simpler terms for the Systems Engineer, interoperability is a property of a system (or product), whose interactions and interfaces are fully understood that allow the system to work with other existing or future systems. To ensure the required interoperability is achieved, the SE identifies, analyzes, defines, and determines the technical interface and integration solutions beginning with concept development and continuing through qualification testing, production, deployment, operations, and support.

4.9.4.1 System Concept Development and Architecture

Interoperability planning begins with early SE activities with the CBA analysis and XR developmental planning process. Requirements to provide system capability in certain form and fit amenable to enterprise-wide sharing and reuse of materiel, parts, and processes are developed. This may include:

- System capability MOEs and MOPs including those for net-ready KPPs that identify as-is and to-be interoperability objectives
- Insertion of enterprise/SoS level interoperability constraints
- Incorporation of basic elements of an open architecture into all technical solution considerations and trades: open-standards, interoperable, interchangeable, portable, modular, and scalable.
- Forward-looking anticipated future requirements to accommodate technology insertions.
- On-going interaction with enterprise level COI maintain widest possible distribution of system capability through availability, compatibility, transportability, interoperability, reliability, safety, human factors, documentation and training requirements among others.

A number of these system concepts for SMC (and other DoD systems) are documented in DoDAF architectures. CJCSI 6212.01 has a list of mandated architectures that a program is required to produce and maintain over the lifecycle of the program.

4.9.4.2 Requirements Analysis and Allocation

The system performance requirements and system interoperability requirements constraints are derived from the JROC validated JCIDS products mentioned above, a complementary functional analyses, and system level trades that provide the engineering justification for the system solutions. The initial requirements document for a program is usually referred to as a System Requirements Document (SRD) or Technical Requirements Document (TRD) and is formally baselined following a successful System Requirements Review (SRR). Much effort is expended by the SPOs to determine the full set of mature requirements for the TD and EMD acquisitions. This requirements definition process is further delineated in the SMC SE Guide SMC-G-001, the SMC SE Handbook, and the SMC SED Handbook. The requirements to ensure the rigors of the requirements development process is adhered to are delineated in the SMC-S-001 Systems Engineering standard.

The SE requirements management activity is also described in the referenced documents. This activity ensures bidirectional traceability of the interoperability related requirements is retained. This activity also ensures that these requirements are systematically matured and controlled: No widows, no orphans. Requirements implementing the USER defined interoperability are written in concise contractual language, parameters are appropriately stated, adjunct verification requirements are established, requirements are appropriately allocated, and each requirement is supported by analyses. The Requirements Management activity also ensures firm baseline control through the configuration management process. The Requirements Management activity also plans for and executes all activities to support required certifications such as interoperability requirements certification, requirements components of the IA certification, requirements components of the space flight worthiness certification, and others. Here we delineate the requirements definition at the system, allocated, and design levels to implement user defined interoperability

4.9.4.3 Interface Analysis and Design

Interface analysis and design is the heart of enterprise-level interoperability and net-readiness.

Traditional SE has long understood the need and desirability of clean and testable interfaces, both internal between subsystems and external to other known or required systems. Interface is viewed as a classic coupling problem between systems. SMC-S-001 defines interface as “The boundary between two or more systems, functions or other logical representations, or system products or between a system and a facility at which interface requirements or constraints are set. Interfaces can be physical or functional.”

Interoperability is the property of an entity that allows for well-defined but unrestricted interaction with other existing, transitioning, or future entities. Entities can be enterprise, system of system, system, subsystem, organization, process, force, or mission – when they have an interface to share. Extant and terms of interoperability within the enterprise are fully documented in open physical, service, or functional interface(s). The standardized and open interfaces are adopted and implemented for the purpose of sharing, reusing, or merging data and materiel, thus providing enhanced emergent capability of the combined interoperating entity. Typically, the use of a fully documented and pervasive interface obviates the need of specialized and unique adaptation devices, equipment, software, or hardware. Interoperability enables form, fit, or function reuse and interchangeability as well as cost-effective enterprise operations over disparate entities.

Interfaces for net-readiness and interoperability are viewed as indispensable ingredient to enable the enterprise. A new or transitioning system, in effect, is conceived as a member of a community of existing, transitioning, and future systems. It is born (or reborn) with enterprise level standard interfaces as mandated constraints. The system joining the enterprise offers its capability to support military operations by entering an appropriate enterprise network where it is managed by the network to share and exchange information. The entities within the enterprise offer to share information, materiel, and services to improve operational effectiveness and performance that can be quantified and measured.

When a new or transitioning system's interfaces are designed for net-readiness and interoperability, they are designed within the context of the enterprise and with the help of the communities of interest. COIs can be standing bodies that acquire, initiate, develop, and maintain necessary knowledgebase for certain interoperable interface. Interoperability is the outcome when key entity (component, system, SoS) interfaces use widely supported open standards that are published and maintained by an authoritative standards organization. Alternatively, ad hoc COIs can be formed to tackle specific or unique problems in interoperability and net-readiness.

Interoperability is predicated on widely supported consensus-based standards that are published and maintained by a recognized standards organization. Such interfaces are characterized by:

- Well-defined and supported
    - Vendor-independent, non-proprietary, publicly available, and widely accepted.
    - Leverage community investment
    - Reduced acquisition cycle time and overall life-cycle cost
    - Ability to insert cutting edge technology as it evolves
    - Commonality and reuse of components among systems
- Interchangeable (within a system)
    - Allow for intermixing of hardware, software, and networks
    - Offer transparent replacements and upgrades for technology insertion
- Portable (among systems)
    - Allow exchange and use of information or materiel
- Modular
    - Physical or logical modularity to meet functional requirements.
- Scalable
    - The ability to grow (and interlink hardware and software) to accommodate increased loads.

Interoperable interfaces are defined at the highest level of integration for widest possible reuse and sharing across the enterprise. For IT and NSS interoperability, DoD CIO has the overall responsibility to provide oversight in coordination with the DoD Components and other mission partners. CJCS certifies NR-KPP and DISA/JTIC is responsible for final interoperability certification.

A capability-focused architecture-based approach is used to implement interoperability solutions across all DoD. Architectural data is developed and iteratively refined over the system lifecycle by DoD enterprise architects, program sponsors, and program managers to support interoperability analysis.

DoD provides extensive policy, instructions, and guidance to design-in widely available interfaces for IT and NSS interoperability to support military operations where information can be exchanged, shared, or reused over managed networks (e.g., SIPRNET and NIPERNET). Materiel interoperability, however, is rather specific to hardware and technology and, as such, is harder to achieve without on-going and close interaction within program elements and between programs over their asynchronous lifecycles.

4.9.4.4 Manufacturing Interoperability

Manufacturing engineering involvement and influences of an evolving item design is now an essential activity to ensure the design meets both manufacturing interoperability standards and a wide array of constraints that are imposed by the intended manufacturing environments. Production efficiencies as well as item quality levels are achieved when the item design is determined in conjunction with the design of the manufacturing process and product layout. This is more so the case for our high reliability space components where incompatibilities and inefficiencies between the item design and its manufacturing environment are identified and resolved early on through manufacturing and producibility considerations during item design, redesign; changes to manufacturing methods, facilities, production equipment and machinery; test equipment, as well as including effective quality controls prior to the start of production. The primary challenges for manufacturing interoperability include:

- Ensure the system or item design is compatible with the anticipated manufacturing environment and production design.
- Ensure the manufacturing technical and procurement information is interoperable throughout the supply chain

4.9.4.5 Integration and Test

Integration brings together related parts into a single system. Integration also is the arrangement of organizations and their systems in a way that allows them to connect, function, and exchange information effectively. Ideally, the integration and test phase provides the opportunity to validate that the launch and space vehicle are designed to interoperate with each other and with all external elements including the build-up, integration, test and checkout facilities, equipment, and procedures provided at the launch sites. Historically though, design and procedural changes are common to overcome integration and interoperability issues that arise during this phase. Such changes must be minimized by designing each system to interoperate with its intended integration and test environment. A brief summary of the space and launch systems integration and processing facilities follows:

- **Payload processing facility (PPF) –** A PPF is used for final assembly and checkout of the satellite which may be referred to as the space vehicle, launch vehicle payload, or simply payload. The PPF houses payload clean-room high bays, control rooms, and offices.
- **Hazardous processing facility (HPF) –** A HPF houses explosion-proof high bays for hazardous operations including liquid propellant and solid rocket motor handling, spin balancing, bays for payload attach fitting (PAF)/payload fairing preparations, and pay-load encapsulation.
- **Horizontal and vertical integration facilities –** The horizontal integration facility is used to process (integrate and checkout) the launch vehicles after their transport from the receiving and storage facility. Work areas are used for assembly and checkout to provide fully integrated launch vehicles ready for transfer to the launch pad. Alternatively, a vertical integration facility is used to assemble, test and mate the launch vehicle with the encapsulated spacecraft. The vertical integration facility may include a clean work area, bridge crane, personnel, and platform to support integration tasks. Services to the launch vehicle include climate control, power, and data exchange through the range fiber network.
- **Launch Pad & Launch Control Center –** Each launch complex is designed and built or modified to meet a launch vehicle's requirements prior to and during the initial launch.

4.9.4.6 Operations and Support

The launch and space system components must also interoperate within the various environments they are placed in. The effects of these environments often are significant contributors to designs, packaging, and transportation, handling, integration, launch, deployment, operations, and disposal. For example, the transportation environments are defined uniquely for each transportation segment that a space vehicle and/or its elements must undergo starting from factory shipping up through launch site LV to SV integration. Design considerations for transporting space system components include shock isolation, temperature control, moisture / humidity isolation and control, electrostatic prevention and control, fire suppression systems, and other considerations.

Once the space system is in place, the stated system capability is made available to the users. It is the mission defined in the front end, and modified over the acquisition lifecycle as necessary, to deliver war-fighting needs and capabilities.

4.10 Data Management

Much of the data produced on a program is technical in nature and describes a technical result, a plan to achieve the result, and/or the basis for the result. Hence, the content, the control, and the archiving of the data are managed as a part of the SE process and with the oversight of the responsible systems engineers acting under the authority of the program manager. Specifically, data should always reflect the balanced consideration of all the products in the product tree that could be affected by the matters under consideration to include the interfaces between those products.

Data often has to meet other requirements and so may also come under the purview of contract, data, and other specialists. Such other requirements and oversight should not be allowed to detract from the technical content, and timeliness of the data.

4.11 Information Assurance

Most SMC programs provide critical time-sensitive information to the warfighter, increasingly in a net-centric environment. SE management is required to take all necessary engineering measures to protect and defend information and information systems by ensuring that they are available to authorized users with the essential integrity and confidentiality. Execution of the IA planning is defined through the IA Strategy document which is developed early in the systems' lifecycle and then maintained to the end. IA components of the program are appropriately represented in program plans, schedules, WBS, cost, and risk.

4.12 Configuration Management

Change management is an important responsibility of any acquisition program. Generally, Program Offices put in place formal change procedures for all requirements that are to be placed on contract such that the initial RFP and all subsequent contract changes are approved by the program director or program manager, the chief systems engineer, the director of financial management, and the contracting officer. Such change procedures would normally handle changes to the system requirements documents such as system specifications and system-level (system-of-systems) interface specifications. These are the top-level configuration documents for the system.

The contract must also require that the Contractor manage and control the configuration of lower-tier products.

Where the contract requires the formal identification and control of the configuration of certain products, the contractor should have procedures in place, as part of the SE process, for determining the corresponding configuration items and their configuration baseline as well as for managing their configuration in accordance with the contract. For all other products, the contractor's decision database should identify the configuration and include means for controlling changes.

4.12.1 Configuration Identification

Configuration identification usually refers to the selection of configuration items (CI) (see definition above), the determination of the types of configuration documentation required for each CI, the issuance of numbers and other identifiers affixed to the CIs, and to the technical documentation that comprises the CIs configuration documentation.

4.12.2 CM Monitoring and Control

Typically, there is one agency or contractor that is recognized to be the final authority over changes to a particular specification or ICD. It is that agency that must implement configuration control procedures for their documentation. In addition, during a developmental effort, lower tiered contractors will establish control procedures to document changes, then submit change request to the higher tiered contractors/government agencies for final approval of changes.

Regardless, each configuration control program is responsible to effectively perform the following:

- Ensure effective control of all CIs and their approved configuration documentation.
- Provide effective means, as applicable, for (1) proposing engineering changes to CIs, (2) requesting deviations or waivers pertaining to such items, (3) preparing Notices of Revision, and (4) preparing Specification Change Notices.
- Ensure implementation of approved changes.

4.12.3 Configuration Status Accounting

Each program and their respective contractors also put in place configuration management status accounting procedures. The typical attributes to a status accounting system includes:

- Identification of the current approved configuration documentation and identification number associated with each CI.
- Status record and reporting of proposed engineering changes from initiation to final approval/contractual implementation.
- Records and reporting of the results of configuration audits to include the status and final disposition of identified discrepancies.
- Records and reporting of the status of all critical and major requests for deviations and waivers that affect the configuration of a CI.
- Records and reporting of implementation status of authorized changes.
- Traceability of all changes from the original baselined configuration documentation of each CI.
- Reporting of the affectivity and installation status of configuration changes to all CIs at all locations.

4.12.4 Configuration Audits

Configuration audits are performed before establishing a functional and product baseline for a configuration item and eventually the system (if the audits are performed incrementally). Configuration audits consist of the Functional Configuration Audit (FCA) and the Physical Configuration Audit (PCA). Additional PCAs may be performed during production for selected changes to the item's configuration documentation or when contractors are changed.

4.13 Operational Safety, Suitability, and Effectiveness (OSS&E)

The OSS&E Assurance program implements AFPD 63-12, AFI 63-1201, and AFMCI 63-1201, “Assurance of Operational Safety, Suitability, & Effectiveness (OSS&E),” for space and missile systems, and addresses portions of AFI 10-1211, “Space Launch Operations.” It is also the guiding document for Draft SMCI 63-1202 “Space Flight Worthiness,” SMCI 63-1203 “Independent Readiness Review Teams,” and SMCI 63-1204 “SMC Readiness Review Process.” This policy applies to all USAF-developed space and missile systems and end items.

The OSS&E assurance program implements a process for establishing and preserving the OSS&E space, launch, and ground/ user baselines or end items over their entire operational life. The Program Office structures and manages the implementation of the OSS&E assurance process throughout the life cycle of the system. Prior to fielding a new system, the Program Office verifies that the system is operated in an operationally safe, suitable, and effective manner and that the OSS&E baseline is adequately maintained throughout its operational life.

The Program Office also certifies that the Space Flight Worthiness of the system at the Flight Readiness Review (FRR). Certification is made to the SMC/CC in accordance with established criteria. The Program Office documents the method of compliance with these criteria. Space Flight Worthiness measures the degree to which a spacecraft, launch vehicle, or critical ground system, as constituted, has the capability to perform its mission with the confidence that significant risks are known and deemed acceptable. Certification is intended to be granted to the "system as constituted" and occur at the FRR based on a best assessment that the system will perform as expected throughout its lifecycle.

The OSS&E Assurance Process for an SMC mission consists of two major portions; an initial assurance assessment and a continuing assessment. The OSS&E Assurance Assessment (OAA) includes processes leading up to the fielding of a system, end item or launch of a satellite. The Continuing OSS&E Assessment (COA) is concerned with continuing OSS&E activities throughout the operational life of the fielded asset. The OAA is a phased assessment of the system and consists of a series of programmatic and independent assessments performed during the acquisition, manufacturing, and mission preparation phases. The scope and type of reviews are based on a program level of maturity. Specific Program Reviews, System Program Director Reviews, and PEO/DAC portfolio reviews are conducted for these modernized systems or end items.

The readiness and mission reviews are conducted before launch. Specific readiness and mission reviews are tailored to meet program needs. The Space Flight Worthiness Certification is accomplished at the FRR. The PFR provides a connection between OAA and COA as lessons-learned from missions are fed back to subsequent pre-flight preparation activities. Detailed descriptions of the reviews are found in SMCI 63-1201.

5 SE Tools and Techniques

5.1 Overview

A number of common SE tool and technique categories were introduced in Chapter 1and listed in Table 1-2. The table also shows a selection of tools based on typical SE development processes, tasks to be accomplished, and the products required for delivery in the space systems development environment. These SE tools are discussed in detail in this chapter in alphabetical order. (Note: INCOSE website can also be explored to browse and identify SE tools that may be useful in a particular situation as it maintains a comprehensive list of open and proprietary tools and techniques.)

During early SE or concept definition phase, we are interested in modeling the mission(s) or capability needs that system is expected to support. We develop mission models and simulations or analyze alternative concepts and solutions in both expected threat and non-hostile operating environments. During the concept phase we also commence with concept and operational architecture definition. We use architecture tools to define the architectures and an assortment of modeling and analyses tools to assess and down-select the best conceptual choices.

As systems definition and development commences, we continue to make use of the modeling and architecture tools used in the previous phase to support analyses and conclude technical or design solutions. During systems definition and development, we now put much more emphasis on requirements development, requirements and design analysis and validation, cost modeling and analysis, and certainly program or project management tools.

Following deployment of a system, tools are also used to perform and manage operations and maintenance. In addition, many of the tools used during development are also used to support major modifications and upgrades. Examples of tools that are candidate to be transferred for continual use following deployment include configuration control and management, and some of the Modeling and Simulation (M&S) and analytical tools that help support system upgrades and modifications.

Maintaining and upgrading a smaller suite of tools is preferable to a larger suite. Hence, much thought and consideration must go into selecting the right set of tools. If tools are selected that have similar functions and databases, there may be the need to transfer data between the tools. Extensive and sometimes costly training requirements might be associated with some tools. Many specialized tools demand expert users that must also have an in-depth knowledge of their discipline to adequately use the tool. There are also many other things to consider such as system requirements (e.g., processing speeds, memory, operating systems), licensing, maintenance, peripheral software and hardware requirements. Any tool selection assessment should also consider 'lessons learned' on other SMC projects.

5.2 AoA and Case Scenarios

A scenario describes expected situations in which the system might operate. Applying these situations to a simulation help visualization of the system's response to multi-faceted operating environment. Using various quantitative and qualitative techniques, it is possible to closely characterize expected environment in which the candidate system will operate. The AoA and case scenarios are used as tools to perform capabilities based analyses and the development of viable system concepts in early SE.

Along with the CBA, the AoA is the focus of early SE. DoDI 5000.02 states "The purpose of the AoA is to assess the potential materiel solutions to satisfy the capability need documented in the approved ICD…The AoA shall focus on identification and analysis of alternatives, measures of effectiveness, cost, schedule, concepts of operations, and overall risk. The AoA shall assess the critical technology elements (CTEs) associated with each proposed materiel solution, including technology maturity, integration risk, manufacturing feasibility, and, where necessary, technology maturation and demonstration needs. To achieve the best possible system solution, emphasis shall be placed on innovation and competition." Using all available technical, affordability, and timeliness measures, AoA attempts to arrive at the best value solution to develop and field a desired capability from a competing set of alternatives.

Scenarios include outlines and synopses of proposed events concerning a customer's problem. One of the most common descriptions is the operations concept. The operations concept is a sequential description of event and functions in the use of a product. The term "mission profile" is sometimes used to include both operations concept and environmental profile. The questions answered by the case scenarios include:

- Why must these things happen?
- What is supposed to happen?
- Who or what is doing these functions or behaviors?
- When do these things happen, and in what order?

The scenarios can be outlined in charts. When a single chart is too confining for comprehensive information, several charts can typically be used to show the overall operations followed by the details for each major operation. Formal case scenarios can be produced using SysML™ or UML diagrams described later in the chapter. The information is then available for derivation of requirements. Case scenarios are often used in conjunction with modeling and simulation (M&S).

5.3 Audits and Reviews

Requirements reviews, design reviews, and configuration audits that include the SRR, SDR, PDR, CDR, IV&V, FCA, and PCA provide an opportunity to assess program status in detail. In particular, requirements and design reviews can be essential to monitoring progress of system development at critical points in the program lifecycle, prior to the availability of test and other verification data that provide a direct indication of contract compliance.

As indicated in Chapter 1, SMC standard SMC-S-021 provides a complete list of audits and reviews required for the program from inception to operations. The objective of audits and reviews is to provide guidance and additional information on how well the system development tasks are progressing over its lifecycle.

5.4 Change Management

SE has the task to document and track of requirements and manage changes to requirements of complex space and launch systems. Systems engineers use databases, spreadsheets, and other common office application software to perform requirements development and change management functions. These tools improve the efficiency to perform these activities, but still provide a restricted environment to a requirements development and change management environment. Comprehensive requirements management tools are available to support multi-user collaborative environments that provide data exchange capability between other common and specialized tools, and make use of modern computer technology and software.

These specialized tools assist us to more effectively collect, define, and decompose requirements, manage changes, and produce requirements specifications. The tool vendors provide us with a broad range of requirements tools capabilities and characteristics. Therefore, before we make a final choice, we are prudent to assess each tool and compare with our program needs. Common features of a requirements management tool may include:

- **Ability to capture and identify requirements –** document, classify, compare, and parse requirements
- **Ability to capture system element structure –** document system architecture
- **Provides traceability analysis –** requirements derivation, flow-down capability, allocation of performance requirements, bi-directional requirement linking to system elements, capture of allocation rationale, accountability, test, verification and validation, identify inconsistencies, criticality, and other issues related to requirements
- **Perform configuration management –** baseline and version control, track history of requirement changes
- **Provide documents and other output media –** specification output, quality and consistency checking, status reporting.
- **Machine interface –** interact with other selected engineering and office tools
- **User interface –** Consistent and intuitive for multiple concurrent users on multiple platforms
- **Support and maintenance –** warranty, network license policy, maintenance and upgrade policy, on-line help

A number of government and industry developed tools are available to manage change like the DOORS and IBM Rational Rose. INCOSE provides an up-to-date listing of such tools for change management on its website.

5.5 Cost Analysis Tools

Engineering plays a key role in the cost basis for space programs. Typically, space systems rely on an available historical statistical database on cost and costing techniques. However, new space systems are always pushing the development and technology envelope, are built in limited production quantities, and require significant customization in architecture and design. This poses a cost estimating challenge for the space engineering community. While an ideal solution to this challenge does not yet exist, a methodology for engineering input into the cost estimating process is well established. Cost estimating and analysis for space systems is required at program milestone reviews to establish a reference cost baseline and to gauge status for management insight and action.

The purpose of cost estimating is to “translate system/functional requirements associated with programs, projects, proposals, or processes into budget requirements [and determine] a realistic view of the likely cost outcome…”[CEBoK 2010]

In developing the engineering estimate for costs, two key documents are used:

- Work Breakdown Structure (WBS)
- Cost Analysis Requirements Description (CARD)

Cost estimates using a WBS in a bottom-up (or in a “build up”) method must be based on work planned, including labor, materiel, and duration. In space acquisitions, a WBS, mapped to the contracted work statement, e.g. Contract Statement of Work (CSOW), is utilized as a framework for capturing work to be performed. Figure 5-1 shows an abridged example of the Space Segment WBS. While ANSI/EIA 632 Processes for Engineering a System provides a general system engineering framework, in space acquisitions the standards used are MILSTD 881C, or the National Reconnaissance Office (NRO) WBS. The Unmanned Space Vehicle Cost Model (USCM) is now updated and incorporated in MILSTD 881C. While all these WBSs might capture the work planned, the SE element within each are broken out differently and suggests caution in comparing and using each of the WBSs. MILSTD 881C uses as common elements the following throughout each WBS Level: SE, Integration and Test, Program Management, and Support Equipment. At the 5th level of definition--"Level 5"-- SE and Program Management are combined into a single "SEPM" (System Engineering and Program Management) element. The SMC standard is MILSTD 881C.

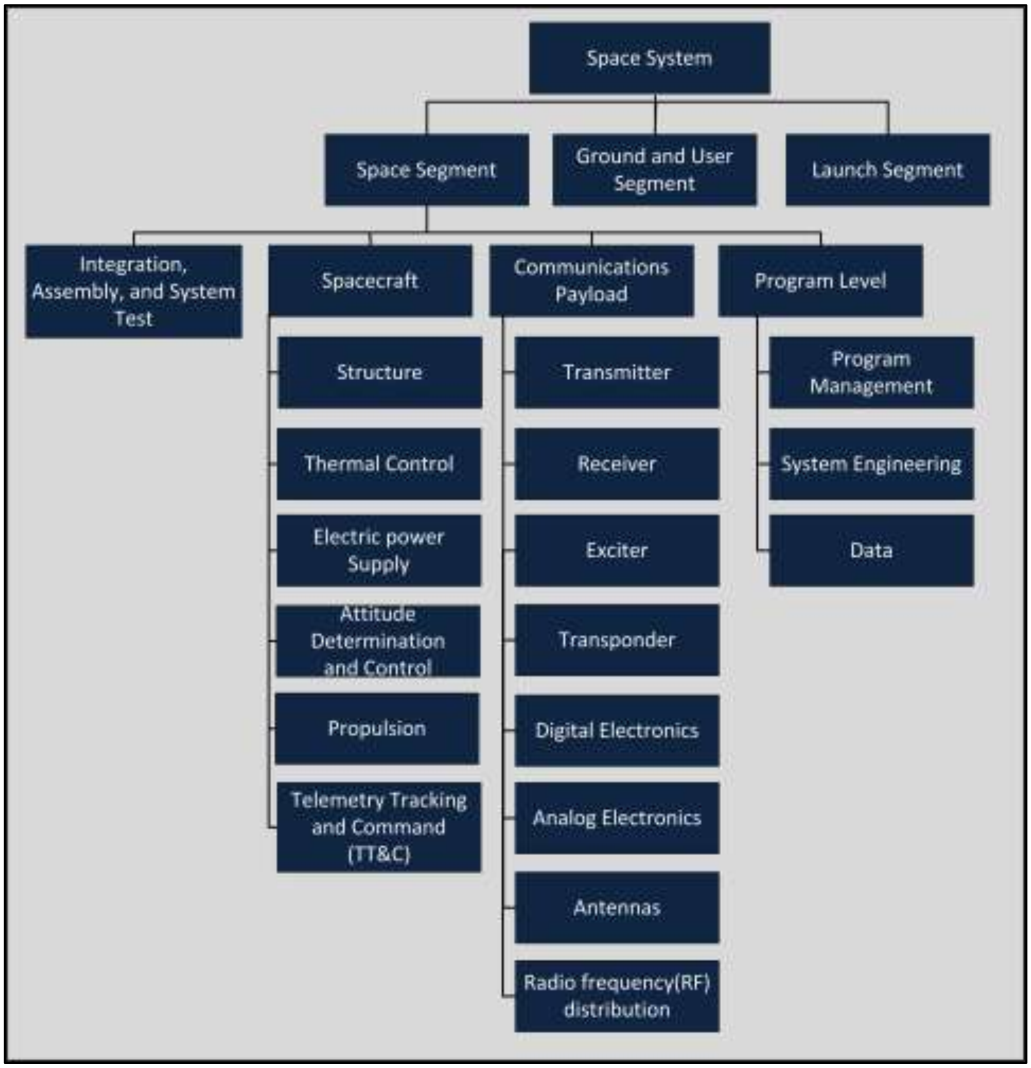


Figure 5-1 Example Space Segment WBS

Within the WBS, work packages at the elemental level are defined to identify what is to be worked and produced. Upon definition, aggregate work packages are bundled in control accounts and used as the basis for determining, bottom-up, the estimated costs of a project. The cost estimate associated with the work packages within the WBS is captured in a Basis of Estimate (BOE). The BOE helps establish project cost baselines, and for defining the work package earned value in an EVMS managed project. For comparison purposes, satellites are categorized in the following WBS categories:

- communication
- navigation
- meteorological (including environmental)
- experimental
- scientific
- surveillance
- radar

Projects can use various tools and techniques for analyzing and deriving estimated costs. A common set is shown in Table 5-1.

Table 5-1 Cost Tools and Techniques

Method	Description	Example	Accuracy/Comments
Analogy	Costs by the similarity of actual previous, similar projects/features	Cost of previous spacecraft bus now used in another program	Less resource intensive, but less accurate
Resource Cost Rates	Costs by going unit cost rate	\$/lb., \$/staff-hr.	Accurate if rates are current
Bottom-up	Costs by aggregated detailed work package and schedule activities	Cost of combined work packages for CDRLs development	Smaller the project, greater the accuracy
Parametric	Costs by statistical relationship between historical data and other technical/programmatic information	Number of solar panels at historical \$M each; \$/SLOC;	Accurate with greater sophistication of database
Project Management Software	Cost estimating tools, simulations, spreadsheets	Constructive SE Cost Model (COSYSMO®), Constructive Cost Model II (COCOMO ™ II)	Accurate with greater sophistication of database and applicable methods
Vendor Bid Analysis	Analysis of bids received on a solicitation and estimates of what the project "should cost"	Commercial launch service bids to International Space Station	Accurate with most credible data supporting costs
Reserve Analysis	Estimates of contingency resources needed	"known unknown" reserves for deliveries	Helps temper over-optimistic work package schedules
Cost of Quality	Costs for investments in preventing non-conformance to requirements; failure costs	Investments in risk management, QA	Includes cost of rework to correct failures

For complex projects, the CARD serves as the basis of cost estimates. The CARD is not an estimate in itself, but contains a technical, programmatic, and a schedule that are used as the basis for deriving a cost estimate.

Once a CARD is developed, cost analysts apply various methodologies and models to derive an estimate of program costs. To deal with complexity, parametric Cost Estimating Relationships (CERs) are established. CERs use information (physical, performance, operational, programmatic) to show the costs as a function of those parameters. Examples are cost projections as a function of space vehicle weight ($/lb.), parts ($/numbers of part type), Software Lines of Code ($/SLOC), and power ($/Watt). A typical outline for a CARD is shown in Highlight Box 5-1.

Highlight Box 5-1
Typical CARD Outline

1. System description and characteristics
2. System suitability and dependability factors
3. Predecessor and/or Reference System
4. PM's assessment of program risk and risk mitigation measures
5. System operational concept
6. System sustainment concept
7. Time-phased system quantity requirements
8. System manpower requirements
9. System activity rates (operating tempo or similar information)
10. Facilities requirements
11. Summary of security or program protection features
12. Summary of environment, safety, and occupational health considerations
13. System milestone schedule
14. Summary of acquisition plan or strategy
15. Plans for system disposal
16. Track to prior CARD
17. Approved or proposed CSDR plan

5.6 Cross Correlation Chart

Figure 32 is an example of a cross-correlation chart. It allows the analyst to relate customer requirements to product features to assure that all requirements are being met and that unnecessary features are not included without being addressed. In Figure 5-2, a dot at an intersection indicates that a particular feature contributes in part or in whole to the achievement of a customer requirement. Notice that Customer Requirement 8 is not satisfied by any product feature. The analyst should determine the importance of Requirement 8 and whether it is sufficiently important to launch a design effort to incorporate it. Likewise, Product Feature E has no corresponding customer requirement. The analyst should determine whether Feature E is required for performance of the system now or in the future and the additional costs incurred. If Feature E is expensive, tends to lower reliability, or is a commonality feature that would be costly to remove from present production, and the feature has no immediate requirement, the analyst might decide to eliminate it or incorporate it in a later version when the need arises.

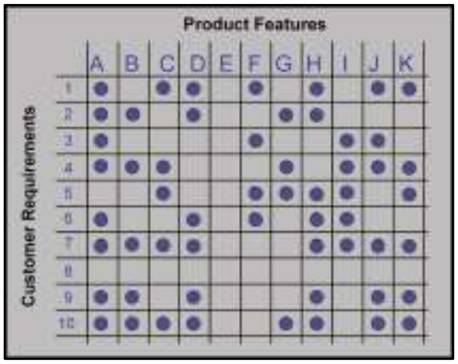


Figure 5-2 Cross correlation Chart

Self-Interaction Matrix is a related concept as depicted in Figure 5-3. It shows how different requirements impinge on each other, either positively or negatively. For example, an improvement in performance may adversely affect reliability or availability. Likewise, incorporation of a Built-In Test (BIT) may reduce Mean Time To Repair (MTTR). In Figure 5-3, Requirement 1 affects or is affected by Requirements 2, 4, 5, 7, and 9. On the other hand, Requirement 4 interacts only with Requirements 8 and 10. From such a chart, the analyst is reminded that when designing to satisfy one requirement, he must be aware of the effects on those related requirements.

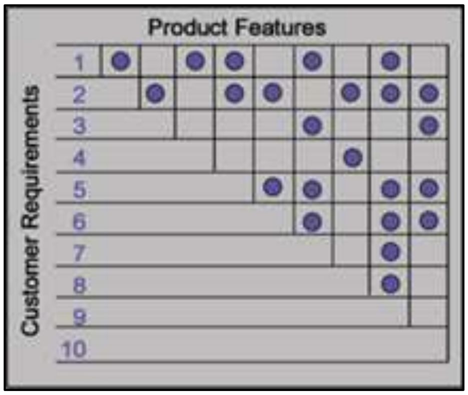


Figure 5-3 Self interaction chart

5.7 DoDAF Architecture

DoDAF is an architecture development framework. It is one of the many tools that capture an understanding of how a military system may be architected or organized for a particular mission or operational scenario. It is designed to meet development, design, and operational needs of the DoD by representing an enterprise architecture that allows stakeholders to focus on specific areas of interests in the enterprise, while retaining sight of the big picture. DoDAF offers a coordinated and consistent approach to architecture development, integration, and presentation.

DoDAF divides complex system description problem into manageable and focused artifacts called viewpoints. Each viewpoint has a particular purpose. The information is organized and presented that expressed a specific stakeholder point of view, typically within the overall enterprise architecture. They represent (i) high-level system or operational concepts and capabilities, (ii) specific interface, interoperability, interconnection requirements, or (iii) data model and flow. Taken together these graphical and tabular artifacts or viewpoints attempt to present a complete and consistent picture of the enterprise to enable effective technical and managerial decision-making.

5.7.1 DoDAF Meta-Model, DM2

DoDAF is a data-centric model. Its data model, the DoDAF Meta-Model (DM2), forms the basis of all DoDAF viewpoints. It enforces the need to have an enterprise-wide common vocabulary and a data hierarchy to ensure that architecture descriptions can be compared and related across organizational boundaries. Graphical and tabular DoDAF viewpoints, based on the underlying program data and its model, lead to common and widely understandable architecture descriptions for both technical and management activities.

DoD CIO states, “The purpose of DoDAF is to define concepts and models usable in DoD’s six core processes:

- Capabilities Integration and Development (JCIDS)
- Planning, Programming, Budgeting, and Execution (PPBE)
- Acquisition System (DAS)
- Systems Engineering (SE)
- Operations Planning
- Capabilities Portfolio Management (CPM)”

The purposes of the DM2 are (i) establish and define the constrained vocabulary for description and discourse about DoDAF models and their usage in the 6 core processes listed above, (ii) specify the semantics and format for Enterprise Architecture (EA) data exchange between architecture development and analysis tools and architecture databases across the DoD and with other authoritative data sources, (iii) support discovery and understandability of EA data using DM2 categories of information and its precise semantics augmented with linguistic traceability, and (iv) provide a basis for semantic precision in architectural descriptions to support heterogeneous architectural description integration and analysis in support of core process decision making.

5.7.2 DoDAF Viewpoints

DoDAF organizes the DoDAF-described Models into the following viewpoints:

- **The All Viewpoint (AV) –** describes the overarching aspects of architecture context that relate to all viewpoints.
- **The Capability Viewpoint (CV) –** articulates the capability requirements, the delivery timing, and the deployed capability.
- **The Data and Information Viewpoint (DIV) –** articulates the data relationships and alignment structures in the architecture content for the capability and operational requirements, system engineering processes, and systems and services.
- **The Operational Viewpoint (OV) –** includes the operational scenarios, activities, and requirements that support capabilities.
- **The Project Viewpoint (PV) –** describes the relationships between operational and capability requirements and the various projects being implemented. The PV also details dependencies among capability and operational requirements, system engineering processes, systems design, and services design within the DAS process.
- **The Services Viewpoint (SvcV) –** is the design for solutions articulating the Performers, Activities, Services, and their Exchanges, providing for or supporting operational and capability functions.
- **The Standards Viewpoint (StdV) –** articulates the applicable operational, business, technical, and industry policies, standards, guidance, constraints, and forecasts that apply to capability and operational requirements, system engineering processes, and systems and services.
- **The Systems Viewpoint (SV) –** for Legacy support, is the design for solutions articulating the systems, their composition, interconnectivity, and context providing for or supporting operational and capability functions.

Each of the viewpoints is captured in one or more pictorial, graphical, or tabular artifacts. Various diagramming techniques including FFBD, NxN, IDEF0, entity relationship diagrams, UML, SysML™, or other custom techniques can be employed to build DoDAF artifacts. Current DoDAF architecture documents and related advice available on the web from DoD CIO and other sources should be consulted for more details.

See current CJCSI 6212.01 for mandated DoDAF viewpoints for DoD programs over the program lifecycle.

5.8 Earned Value Management

Earned Value Management System (EVMS) is a program management technique to assess cost and schedule risk and take proactive action to deliver expected products or services on time and within cost. It is a type of metric applied to system engineering for reporting on present expenditures against planned work completion; and provides insight on future resource needs and task schedules against their program constraints. It relates resource planning to schedules and technical cost to encourage effective cost and schedule management. EVMS methods help provide timely data to all stakeholders on program status and direction. EVMS is applied to complex programs wherein program cost and schedule risks for expected deliveries must have strict active controls.

EVM integrates the technical, cost, and schedule parameters of a contract. During the planning phase, an integrated baseline is developed by time phasing budget resources for defined work. As work is performed and measured against the baseline, the corresponding budget value is “earned”. From this earned value metric, cost and schedule variances can be determined and analyzed. Variance measurements help the program manager identify significant drivers, forecast future cost and schedule performance, and construct corrective action plans to get the program back on track, if necessary. EVM therefore encompasses both performance measurement (i.e., what is the program status?) and performance management (i.e., what we can do about it?). EVM provides significant benefits to both the Government and the contractor. A fundamental requirement for managing any major acquisition system is insight into the contractors' performance specifically the program management and control. Proper EVM implementation ensures that the PM has access to contractor performance data that:

- relates time-phased budgets to specific contract tasks and/or statements of work (SOW)
- objectively measures work progress
- properly relates cost, schedule, and technical accomplishment
- allows for informed decision making and corrective action
- is valid, timely, and able to be audited
- allows for statistical estimation of future costs
- supplies managers at all levels with status information at the appropriate level, and
- is derived from the same EVM system used by the contractor to manage the contract.

DoDI 5000.02 requires EVMS for cost or incentive contracts, subcontracts, intra-government work agreements, and other agreements valued $20M or more. Contracts valued more than $50M require formal government contracting officer validation of the EVMS. EVM is typically not necessary, regardless of cost, on Firm-Fixed Price, Time and Material Contracts, and LOE activities. However, contractors may elect to use EVMS to ensure cost targets, such as for FFP, are on-track and corrective and timely action can be taken when deviations from planned expenditures and earned value are observed.

Key concepts and terms for EVMS for system engineering project assessments are listed in Table 5-2.

Fundamental to EVMS are accurate and well defined work packages (WP) which capture the task and products ("value") to be delivered against a schedule. The WPs reflect the work to be accomplished and flows from the Work Breakdown Structure (WBS) for the project. The WPs are costed and entered onto a schedule to indicate the level of work effort and other resources needed to achieve value against the deliverable. Work packages are usually aggregated for higher levels of management. The resources managed in these aggregations are managed in an EVMS control account (CA). The manager for the CA is referred to as the control account manager (CAM). Typically, the engineering product IPT lead serves as the CAM.

As scheduled work is performed, the budget tied to actual work accomplished or earned (referred to as the Budgeted Cost of Work Performed or BCWP), is referenced against the associated planned budget for that value (the Budgeted Cost of Work Scheduled, or BCWS). The difference or variance between BCWS and BCWP will show the relative earned value and further indicate whether the performed value of work is, in monetized schedule terms, ahead, on time, or late. In a similar way, the actual cost incurred (referred to as the Actual Cost of Work Performed, or ACWP) for the worked accomplished (BCWP) will indicate whether ongoing cost expenditures for the work package is, in monetary terms, under-run, at cost, or overrun.

Efficiency in schedule and cost are defined by the relative ratios of earned value against that which was scheduled and that which was actually expended. These are called cost and schedule performance indices, CPI and SPI, respectively.

Table 5-2 Key EVMS Concepts and Terms

Key EVMS	Concepts	Terms
ACWP	Actual Cost of Work Performed	Cost actually incurred in accomplishing work performed = ACTUAL COST
BAC	Budget At Completion	Total budget for total contract thru any given level
BCWP	Budgeted Cost for Work Performed	Value of completed work in terms of the work's assigned budget = EARNED VALUE
BCWS	Budgeted Cost for Work Scheduled	Time-phased Budget Plan for work currently scheduled = PLANNED VALUE
CA	Control Account	Lowest CWBS element assigned to a single focal point to plan & control scope / schedule / budget
EAC	Estimate At Completion	Estimate of total Cost for total contract thru any given level
MR	Management Reserve	Budget withheld by contractor PM for unknowns / risk management
TCPI	To Complete Performance Index	Efficiency needed from 'time now' to achieve a Cost Target = BAC, LRE, or EAC
WP	Work Package	Near-term, detail-planned activities within a CA

The Variance At Completion (VAC) provides a reference projection into the future. This information is useful for taking proactive engineering and management actions to keep a project within expected program constraints. VAC is the difference between the original Budget At Completion (BAC) and the projected Estimate At Completion (EAC). The EAC is the sum of actual costs to date, and the cost of remaining work factored-in with the ongoing cost efficiency index. Table 5-3 summarizes these and related EVMS relationships in simple mathematical terms.

Table 5-3 Key EVMS Equations

VARIANCES **Positive is Favorable, Negative is Unfavorable**			**EFFICIENCY: Favorable is > 1.0, Unfavorable is < 1.0**		
			Schedule Efficiency **SPI = BCWP/BCWS**		**Cost Efficiency** **CPI = BCWP/ACWP**
Cost Variance	Schedule Variance	Variance at Completion	To Complete Performance Efficiency TCPI = Work remaining/cost remaining = (BAC- BCWP)/(EAC-ACWP)		
CV = BCWP – ACWP	SV = BCWP – BCWS	VAC = BAC – EAC	OVERALL STATUS		
CV % = (CV / BCWP) * 100	SV % = (SV / BCWS) * 100	VAC % = (VAC / BAC) * 100	% Schedule = (BCWS_{CUM} /BAC) * 100	% Complete = (BCWP_{CUM} /BAC) * 100	% Spent = (ACWP_{CUM} /BAC) * 100
Estimate at Completion (EAC): Actuals to Date + [(Remaining Work) / (Performance Factor)]					
EAC_{CPI} = ACWPcum + [(BAC-BCWPcum)/CPIcum]					

5.9 Failure Analysis Tools and Techniques

The purpose of failure-mode effects analysis (FMEA) is to identify the ways a given system can fail to deliver expected performance. The failure analysis for space systems may include information from many sources and subsystems, such as data on fairing separation, solar panel deployment mechanism and algorithm, attitude control subsystem function, and possibly the interaction among their concurrent activities. In FMEA, a mathematical model is usually created and used in the analysis.

FMEA is used to evaluate systems, product designs, processes, and services. It offers help in identifying how a part, subsystem, or system might fail, or how it can impact system safety and effectiveness. In addition to identifying potential design flaws, FMEA may also help with the following:

- Identifies parts and components that need further design, testing, and analysis to improve dependability
- Identifies parts, processes, and operations where redundancies can help improve system availability
- Identifies parts, processes, and operations where education can help eliminate misuse of a product
- Offers a foundation for reliability assessment and risk analysis
- Offers a basis for effective communication and decision making

Both bottom-up and top-down approaches cab be used to conduct FMEA.

Failure Mode, Effects, and criticality Analysis (FMECA) is a bottom-up methodology. It starts at the component or parts level, and progresses toward the integrated system to identify overall failure modes and their criticality. FMECA should be performed as the system is designed, integrated, and tested. Iteration of the FMCEA is necessary when (i) design is changed or (ii) new data from testing or operations becomes available.

Fault-tree analysis (FTA) is a top-down methodology. The analysis is used to build a tree that links the cause(s) of a system failure to more basic or lower level parts, processes, and events. FTA offers a lessons-learned database to prevent problems and improve system reliability.

Root-cause analysis (RCA) is another approach to analyze major failures and accidents. It is a quantitative approach that starts with the creation of an expert interdisciplinary team that includes in part personnel not directly related to the failed system. The team (i) collects all data relevant to the failure, (ii) performs analyses to determine how and why the failure event occurred, and finally (iii) identifies what corrective actions need to be taken to redesign parts, processes, and operations to prevent reoccurrence of the accident.

5.10 Functional Flow Block Diagram (FFBD)

FFBDs portray the sequential relationships among functions at each given level, and provide a framework for deriving performance requirements for the system and/or all subordinate system elements. FFBDs are the means used to document the Functional Analysis. Figure 5-4 shows the typical symbols used in block diagrams. A detailed discussion of the symbols or conventions used follows.

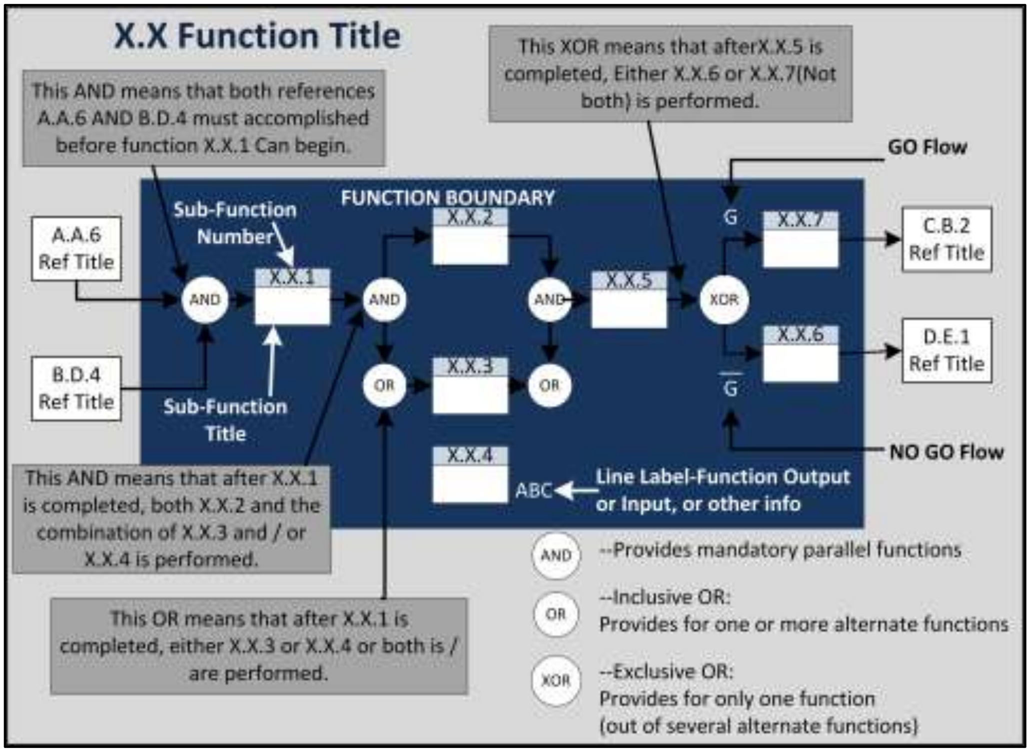


Figure 5-4 Sample functional flow block diagram (FFBD)–typical symbols used in FFBDs

Function Blocks on a FFBD are shown as a solid box having a number and a title. The traditional form contains the number in a separate "banner" at the top of the box, and the title in the major portion of the box. The number is unique to that function, and has nothing to do with the sequence in which the functions may be performed; it identifies the function's level within, and relationship to, the functional hierarchy. For example, the top-level system flow, FFBD 0.0, shows the sequential relationships among Functions 1.0, 2.0, 3.0, 4.0, 5.0, etc. When Function 5.0 is decomposed (i.e., broken into its component parts), relationships among Functions 5.1, 5.2, 5.3, 5.4, etc., and the functions/entities external to function 5.0 would be shown. Decomposing Function 5.4 would portray relationships among Functions 5.4.1, 5.4.2, 5.4.3, 5.4.4, 5.4.5, etc., and the functions/entities external to function 5.4. Using titles without a numbering scheme would make it extremely difficult to recognize where a particular function/FFBD would fit in the functional hierarchy.

Function titles must consist of an active verb and a noun. (Other parts of speech are optional and may be used to narrow or clarify the scope to the function). Ideally, the noun should be a measurable attribute, and the verb-noun combination something verifiable. Nouns should not be a part or activity. This can prove difficult at first. For example, "provide power" is better stated as "power electronics." Active verbs are something that can be demonstrated. Keep it functional, and avoid describing physical parts.

External Reference Blocks represent other entities or functions that are external to the function depicted by the diagram. On the 0.0 FFBD, the reference blocks are all entities that interact with the system but are external to it. These are shown as dotted boxes on the left and right sides of the FFBD. An alternate and more traditional way is to use "brackets" instead of a dotted box.

When a function is decomposed, it is important to depict accurately the preceding and succeeding functions and reference blocks that appear on the higher level FFBD as external reference blocks on the decomposed FFBD. Since the external reference blocks on the 0.0 FFBD (Top-Level System Flow) are shown to interact with the system functions on the 0.0 FFBD, that interaction must also be captured when those functions are decomposed. All of the external reference blocks on the 0.0 FFBD must appear on at least one of the FFBDs depicting decomposition of the 0.0 FFBD functions, and on down through the hierarchy. If they have no relationship to the parts of the decomposed functions, they could not have had any relationship to the functions at the 0.0 FFBD. On lower level FFBDs, functions from the higher level FFBD must appear as reference blocks on the left and/or right sides of the subject FFBD, and be linked by sequencing arrows to the appropriate sub-function(s), if they are precursors or successors to the subject function on the higher level diagram. Maintaining the relationships portrayed on higher level FFBDs at the next lower level is essential to ensuring the integrity of the functional analysis. If this is not done, the process breaks down. Functions do not exist in isolation; there is always at least one function or one reference (function or external entity) that precedes it, and almost always at least one that follows it. That is why functional flows flow. (The one exception that forces the use of “almost always” might be the function: Disposing of the System/Components.)

There is another instance where external reference blocks are used. That is when you utilize a function from an existing FFBD rather than identify a new function with the same performance as the already existing function on the other diagram. When this is done, it is essential to go back to the FFBD on which the reference block originally appears as a function block, and show the functions with which it interacts (from the FFBD where it is “borrowed” as a reference) as reference blocks on the left and/or right sides of the flow, as appropriate. This is necessary so that all functions with which the “borrowed” function interacts are portrayed in one location, its primary usage location.

Internal Reference Blocks also appear as dotted boxes or brackets. There are instances where, for the sake of clarity, a function within a FFBD is used in more than one location. This enables a clearer depiction of the functional relationships. The first time it appears it appears as a normal function block; for any subsequent uses on the diagram, it appears as a reference block.

Floating Block may be either a Function Block or a Reference Block. It is called a Floating Block because no sequencing arrows (see below) connect it to any other Function Block on that diagram. It may be used when the subject block is a precursor to, and/or a successor to, all the other Function Blocks on the diagram. In either use, the key consideration is that it relates to all the other functions;

As a Reference Block:

- If it appears as a Reference Block on the left edge of the diagram (along with the other Reference Blocks on the left side), it is a precursor to all the Function Blocks in the diagram.
- If it appears as a Reference Block in the right edge of the diagram (along with the other Reference Blocks on the right side), all the Function Blocks in the diagram are precursors to it,
- If it appears as a reference block in the bottom center of the diagram, it is both a precursor to, and a successor to all the Function Blocks in the diagram.

As a Function Block (Although a Floating Function Block cannot have any sequencing arrows connecting it to any other Function Block on the diagram, it may have sequencing arrows connecting it to reference blocks on either the left or right side of the diagram but not both.)

- If it appears as a Function Block towards the bottom-left of the diagram, it is a precursor to all the Function Blocks in that diagram.
- If it appears as a Function Block towards the bottom-right of the diagram, all the Function Blocks in the diagram are precursors to it.
- If it appears as a Function Block in the bottom-middle of the diagram, it is both a precursor to, and a successor to all the Function Blocks in the diagram. NOTE: Other programs may use the bottom-middle positioning to indicate that the Floating Function Block is only a precursor to all Function Blocks on the diagram.

Sequencing Arrows indicate the sequence in which functions are performed. An arrow leaving one function and entering another indicates that the function into which the arrow enters is performed after the one from which it exited. An arrow entering a function almost always enters from the left (never from the right) and almost always exits from the right (never from the left). The above statement is qualified with “almost always” because there are rare instances where arrows enter the top of a function block and/or exit from the bottom. Arrows are unidirectional; they never have two heads.

FFBDs are not data flow diagrams (DFD); they do indicate the sequence in which the functions are performed. If some of the functions being performed are involved with the processing or transferring of data (or some other product), some of the function sequences would correspond to a data (or product) flow. On a FFBD there is often a mix of functions that process/transfer product, and functions that perform other activities. So, in some instances the sequencing arrows may indicate an actual product transfer from one function to another; in other instances nothing more than an implication that “this function is/may be performed next.” This duality is sometimes difficult to grasp.

To help clarify the relationship of the functions connected by a sequencing arrow, arrow/line labels may be used. The label could indicate the “product” transferred from one function to the next function, or describe the conditions associated with each of the alternate paths. Both uses (the “GO – NO GO” alternatives, and “ABC Function Output/Input”) are portrayed within Figure 51.

Connectors. Any time it is intended to show that more than one function may be performed before a function, or may be performed after a function, a connector is utilized to join the sequence arrows linking the functions. The type of junction must be defined, and connectors are the means used to define the junction. The approach described here is not universal; some approaches do not distinguish between inclusive and exclusive ORs, while others do not use inclusive ORs at all. The former approach is workable, but may lose clarity; the latter is not really workable. It is not possible to describe all possible function relationships without the use of some form of inclusive OR.

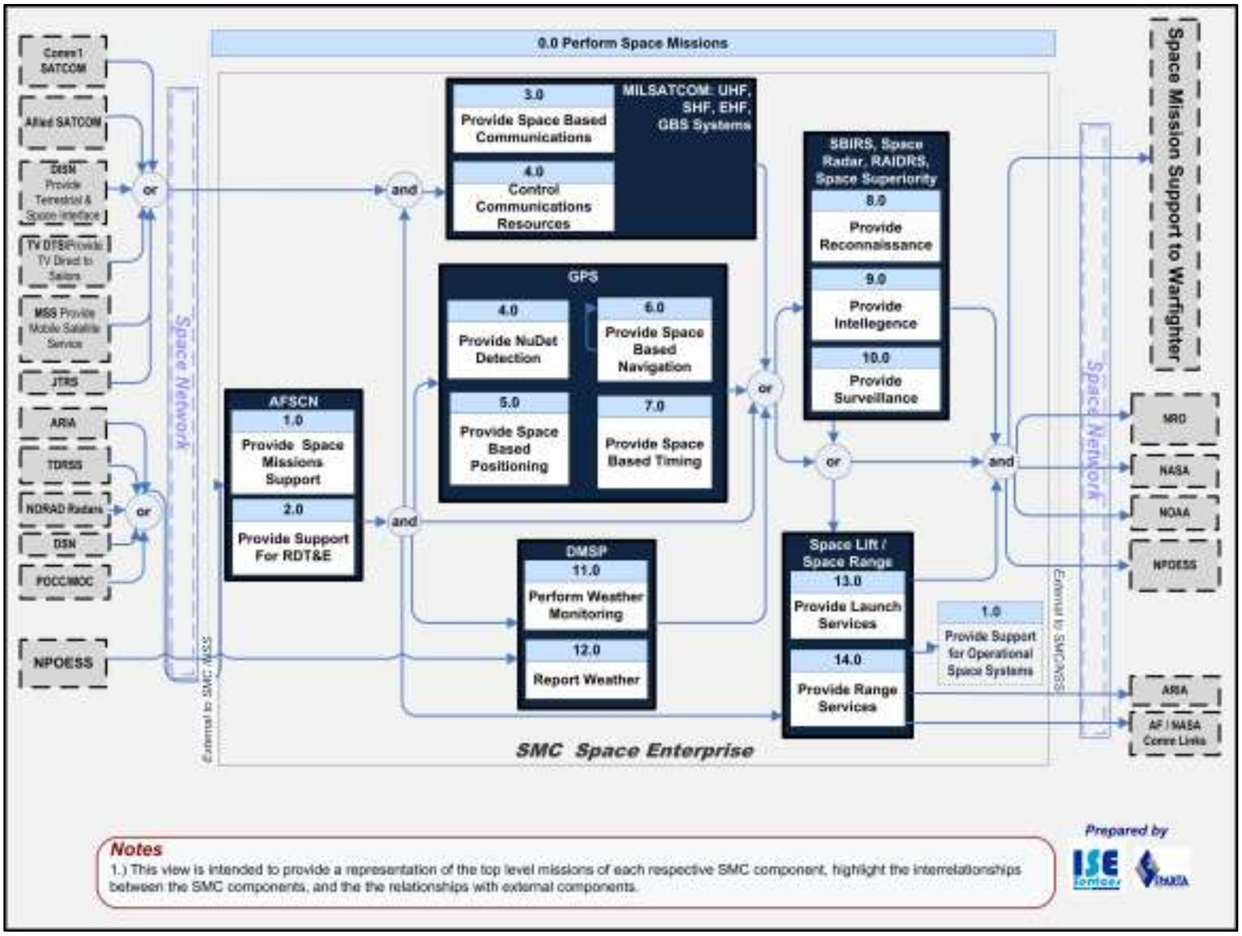
There are three types of connectors used: the AND, the OR, and the XOR. On a FFBD they appear as small circles with AND, OR, or XOR inside. The OR represents an inclusive or; the XOR represents an exclusive or. There are seven basic rules/conventions governing the use of ANDs, ORs, and XORs:

1. If two or more arrows enter an AND, all functions they originate from are always performed before the function following the AND is performed.
2. If there are two or more arrows originating from an AND, all functions to which they go to are always performed after the function preceding the AND is performed.
3. If there are two or more arrows entering an OR, at least one of the functions from which they originate is always performed before the function following the OR is performed.
4. If there are two or more arrows originating from an OR, at least one of the functions to which they go is always performed after the function preceding the OR is performed.

5. If there are two or more arrows entering an XOR, only one of the functions from which they originate is performed before the function following the XOR is performed.
6. If there are two or more arrows originating from an XOR, only one of the functions they go to is performed after the function preceding the XOR is performed.
7. Multiple inputs and multiple outputs to/from the same connector (AND, OR, or XOR) should not be used.

Function Descriptions may not be visible on the FFBD, itself, but are an essential aspect of Functional Analysis. The function description is a much more thorough explanation of what the function does than the title, alone. It bounds the function by limiting what is included within it: when it begins, when it ends, and what happens in the interim. It can also serve as an outline or checklist for the requirement developer(s) to insure that all aspects of the function are addressed by requirements.

An FFBD example representing the SMC enterprise functional flow is shown in Figure 5-5.[33]



Notes
1.) This view is intended to provide a representation of the top level missions of each respective SMC component, highlight the interrelationships between the SMC components, and the the relationships with external components.

Figure 5-5 SMC Enterprise (Source: AFSCN Architecture Description Document (ADD), 2008)

---

[33] First developed for AFSCN Architecture Description Document (ADD), 2008.

5.11 IDEF0

IDEF0 is a graphical representation of an enterprise, business, or a system to describe functionality or operation from a specific viewpoint to any level of detail. Developed by the US Air Force program for Integrated Computer Aided Manufacturing (ICAM) as part of a larger construct, IDEF0 provides rigorous and precise system description, and promotes consistency of usage and interpretation. IDEF0 is used to show data flow, system control, and the functional flow of life cycle processes.

An IDEF0 context or A0 diagram is always the highest level representation of the model and contains only one process, as shown in Figure 5-6, developed by NIST. The context diagram can then be decomposed to any level of finer detail by creating constraint diagrams as shown in Figure 5-7, also developed by NIST. Boxes in an IDEF0 diagram are numbered hierarchically. A0 is the top level context diagram whereas A1, A2, …, An, represent level 1, 2, …, n decompositions, respectively.

Each box is titled with an active verb or a verb phrase showing what needs to be accomplished. Figure 5-6 as developed by NIST shows the top level, typically labeled A0, context diagram. Each side of a function box has a standard box and arrow relationship:

- **Input –** arrow shows the data or objects that are provided as inputs to the function. It interfaces with the left side of a box (Issues and Operations data in Figure 5-6).
- **Control –** arrow, a form of input used to direct the activity, shows the necessary conditions required to produce correct output from input data or objects. It interfaces with the top side of a box (Program Charter in Figure 5-6).
- **Output –** arrow shows the data or objects produced by a function. It interfaces with the right side of the box (Program Plan in Figure 5-6).
- **Mechanism –** arrow shows the means (tools and resources) used to perform a function and includes the special case of a call arrow. It is associated with the bottom side of an IDEF0 function box, pointing upward to connect to the bottom side of the box (Program Team in Figure 5-6).

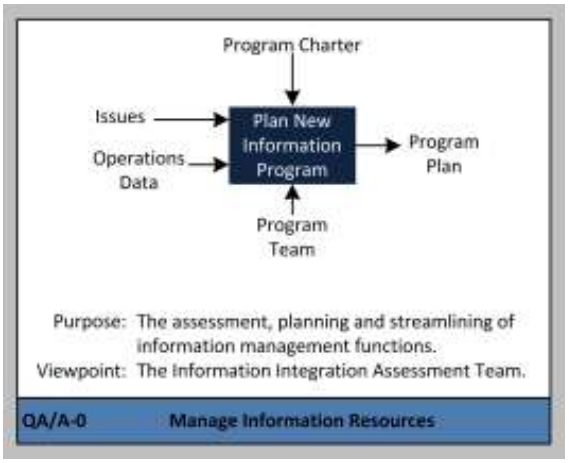


Figure 5-6 Example IDEF0 context or A0 diagram

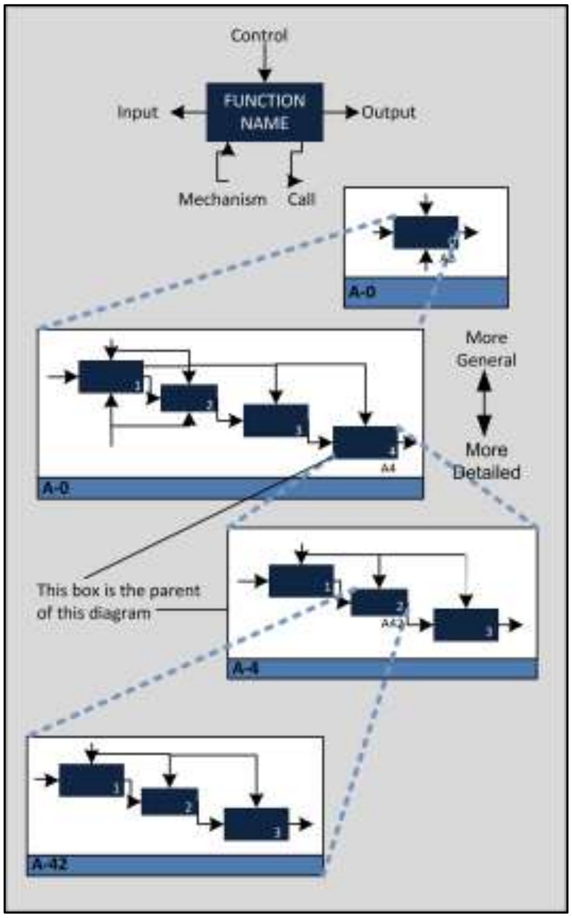


Figure 5-7 IDEF0 diagram hierarchy

- **Call –** arrow is a type of mechanism that enables sharing or linking of detail between models or within a model. It is pointed downward and connects to the bottom side of the box, and is labeled with the reference expression for the box which the subject box shares detail (shown in Figure 5-7).

5.12 Modeling and Simulation

Models and simulations allow for the study of effects of choices without actually building and testing a product. A model is a representation of a process or product that shows the effects of significant design factors. Simulation uses models to explore the results of different inputs and environmental conditions. Models or simulations may be actual hardware or scale replicas, mathematical programs that emulate system operation or processing response, or combinations of both hardware and programs. Often models are built to prove critical technology or to hone configurations. Simulations are used to optimize man/machine interfaces. Operational data may be fed into processing simulators to ensure proper data processing prior to committing to production software and firmware.

Models can be as simple as a picture or sketch. They can also be mathematical and statistical. Beginning models are simple and become more complex with time and improved understanding of the processes involved. The first step in modeling is identifying inputs that can be manipulated and then determining what outputs result for the process or product under study. The next step involves examining the effects of the environment on the product’s performance. Last, the internal transfer function of the product or process to complete the model is represented. When these are tied together, the model is ready to explore system design choices and their suitability.

Traditional optimization theory uses differential calculus, the simplex method, and other mathematical techniques. Computing power is readily available through desktop computers and spreadsheets. Spreadsheets have built-in numerical functions and iteration capabilities, making them ideal for small models. The references listed in the Bibliography are good starting points.

CJCSI 3010.02C, 15 January 2012, describes Modeling and Simulations (M&S) as techniques for testing or analyzing a logical representation of a system, entity, phenomenon or process. M&S is intended to provide readily available, operationally valid environments approved by warfighters to explore concepts and refine capability requirements in preparation for field experimentation. M&S tools that are used to accurately capture current and future Joint and Service capabilities, doctrine, and tactics.

DoDD 5000.59, DoD Modeling and Simulation (M&S) Management, establishes M&S policy including ensuring that M&S investments promote the enhancements of DoD M&S technologies in support of operational needs and the acquisition process; develop common tools, methodologies, and databases; and establish standards and protocols promoting the internet, data exchange, open system architecture, and software reusability of M&S applications. Guidance provided includes:

- Use verified, validated, and accredited models and simulations, and ensure credible applicability for each proposed use.
- Use data from system testing during development to validate the use of M&S.
- Support efficient test planning; pre-test results prediction; validation of system interoperability; and shall supplement design qualification, actual T&E, manufacturing, and operational support.
- Involve the Operational Test Authority (OTA) in SBA/M&S planning to support both developmental test and operational test objectives.
- DIA shall review and validate threat-related elements in SBA/M&S planning.

5.12.1 High Level Architecture (HLA) Tool

A number of powerful commercial off-the-shelf (COTS) modeling and simulation tools such as HLA, or High Level Architecture, are now available. The HLA is a general purpose architecture for simulation reuse and interoperability. The HLA was developed under the leadership of the Defense Modeling and Simulation Office (DMSO) to support reuse and interoperability across the large numbers of different types of simulations developed and maintained by the DoD.

The compliance testing process has been established as the means to insure DoD simulations are, in fact, HLA-compliant in accordance with DoD policy. HLA certification testing is available through a web-based interface which includes a reference library of documents, on-line help, e-mail, and a test registration.

5.13 N2 Diagrams

The N2 or NxN diagram, invented by Robert J. Lano[34] in the 1970s, is a matrix that represents functional or physical interfaces between system elements. The N diagonal matrix elements of the NxN matrix are occupied by the N system functions or nodes. Function outputs are shown in rows and the inputs are shown in columns. The matrix elements around the diagonal identify output-input linkage of the data and the physical interface to other node(s) in clockwise manner, as shown in Figure 5-8 If there is no interface between nodes, the corresponding matrix element contains a "–" or "x" or is left blank.

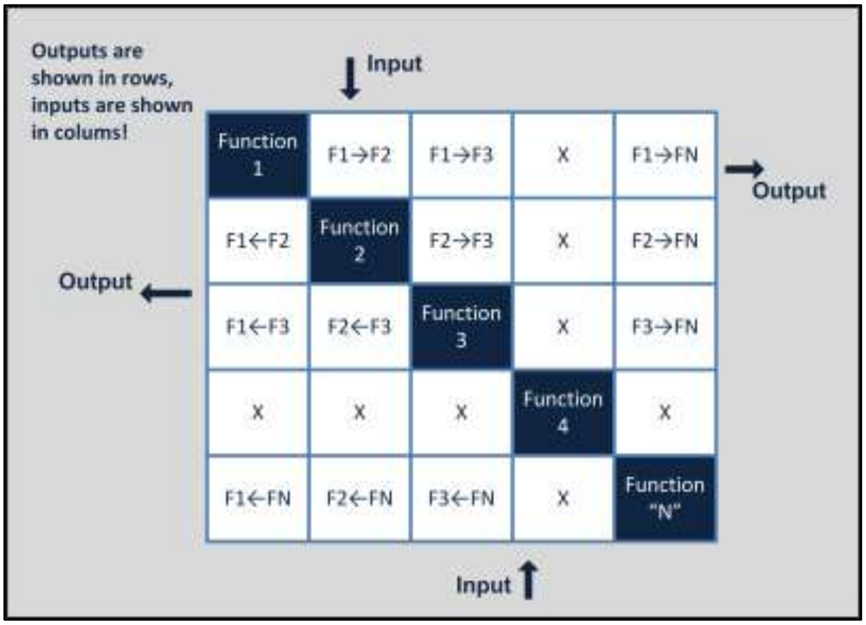


Figure 5-8 N2 Diagram

N2 diagrams can be used as a powerful tool to help analyze the intra- and inter-system interfaces and dependencies. For example Function 4 in Figure 5-8 has no inputs or outputs. This could mean that (i) function is not required, (ii) it can be combined with other functions, or most likely (iii) pertinent information about its inputs and outputs has not been identified or overlooked. Again, Function 3 is critical as it sends and receives information from most other functions that may become a bottleneck, and as such is a candidate for repartitioning. Similarly if some functions (Functions 1, 2, and 3 in Figure 5-8) are tightly coupled (share data back and forth) they could be combined together into a single function to minimize extensive input-output activity. In essence, N2 charts provide a way to balance internal function complexity against external interface complexity.

5.14 Optimization Tools

Optimization in SE is a multidisciplinary activity that seeks to build complex systems that are competitive in performance and lifecycle value. Multivariable optimization employs rigorous quantitative techniques, methods, and algorithms for performing system optimization for system architecting and design. The development of optimization techniques have grown in importance and scope with the advent of fast

---

[34] Lano, R. (1977). The N2 Chart. TRW Software Series, Redondo Beach, CA

computers. Elaborate computer models and simulations can be created to study and optimize system behavior before any hardware or software is built.

Optimization methods are typically based on recognition of an objective state that can be numerically expressed as a single number or a vector with more than one element. The ability of the system to achieve that state depends on a collection of variables that for a space system could include cost, schedule, and capability performance. The values of the variables can be manipulated in order to optimize the objective state. However, most variables cannot be manipulated without a set of constraints. For instance, a manufacturing process cannot require more resources than are available, nor can it employ less than zero resources. Within this broad framework, optimization problems can have different mathematical properties.

5.14.1 Robust Design and Optimization

An important consideration in the development and design process is to assess and strive for robustness of the design – even the 'value' of the robustness.

Optimal design is not always the best solution. Figure 5-9 illustrates this fact. Shown is a design characteristic with two possible design points. Point B is optimal because it produces the maximum Utility. However, the sensitivity of point B is such that small changes in x cause wild swings in Utility. Point A provides lower values, but it is more robust. Fairly wide variations of x cause very little change in Utility. If x is an unknown or uncontrollable factor, design point A is more desirable from an engineering and producibility viewpoint, because of its lower sensitivity to uncontrollable parameters.

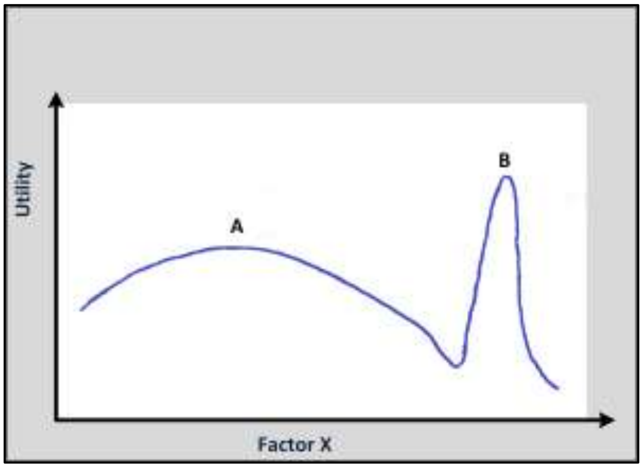


Figure 5-9 Robust design many be better than optimum

5.14.2 Analyzing Sensitivity

Analyzing sensitivity means the measurement of sensitivity of the proposed solution to changes in the value system, requirements, or functions, as well as identifying changes in weights or scoring that might reverse decisions. Utility curves often point out peaks of optimization that might not be stable, and analyzing sensitivity can prevent selecting an unstable design.

You might want to use optimization methods and designed experiments to determine sensitivities to changing environments and other noise. Manufacturing methods are another area you might want to cover.

5.14.3 Optimization Through Experiments

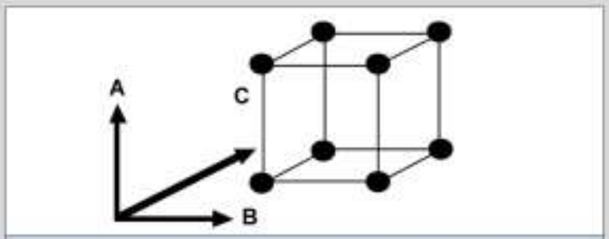
If experiments are used to obtain optimization data, using statistical methods can reduce experimentation time. The term factor is used to denote any feature of the experiment that can be varied, such as time, temperature, or pressure. The levels of a factor are the actual values used in the experiment. Experiments can be designed for best capture of data and reduced number of experiments required. Most engineers are taught to vary one factor at a time in an experiment or simulation, holding everything else constant. This allows observation of each factor's contribution. However, if the number of factors is great, this process requires much time and does not show interactions directly.

For an example of how a designed experiment might save time and cost, suppose two sample levels are proposed in a simulation involving three factors. A three-dimensional, orthogonal representation of the testing is shown in a, Figure 5-10. If each of the factors A, B, and C are exercised at every point, a total of eight simulation runs is required.

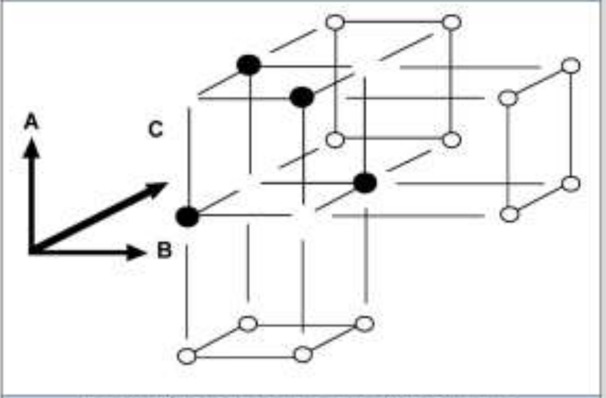
In an experiment of four balanced runs (Figure 5-10), you can extrapolate the other information statistically. The four samples can be projected onto three planes. Each of the planes contains the necessary information to extract other desired data. There are three advantages of designed experiments:

- It takes less time to run the simulations or experiments.
- Unknown biases are avoided.
- Variation from day-to-day and batch-to-batch are balanced out.

The statistical techniques are not difficult. For engineering work, you can use a cookbook approach to performing the necessary mathematics. Consider asking an experienced person in experiment design for help so that you measure the factors properly.



a. Eight Samples for Three Factors at Two Levels



b. Four Balanced Samples allow all Eight Points to Be Extracted Statistically

Figure 5-10 Balanced Experiments can Reduce Experimentation Costs and Schedule

5.14.4 Optimization in Manufacturing (Taguchi Method)

Dr. Genichi Taguchi's methodology for quality engineering optimization has been used in Japan for more than 30 years. It uses two tools, the Signal-to-Noise Ratio and the Quality Loss Function. The idea is to develop high-quality, low-cost products that incorporate robust designs that are insensitive to variability factors encountered in manufacturing and the field. This approach differs from the Go/No Go design and test methods normal to American operations. The Taguchi method borrows the Signal-to-Noise Ratio concept from communications engineering. Products with good signal-to-noise ratios are impervious to noise.

In this context, noise factors are anything over which the engineer has no control. Noise causes quality characteristics to deviate from the target, which results in a loss. The three types of product noise are:

- External noise - variables in the environment or conditions of use.
- Internal noise - changes that occur when a product deteriorates or ages.
- Unit-to-unit noise - differences between individual units that are manufactured to the same specification (manufacturing noise).

The engineer does not attempt to control the noise factors. Such control is usually expensive and may be impossible. The engineer designs around the noise factors, choosing parameters and values that minimize the effects of the noise.

The Taguchi method is not aimed at identifying cause-and-effect relationships. It is not necessary to understand the causes in order to produce a robust design that is not sensitive to variations. However, the method does place strong reliance on the product knowledge of the engineer. The Quality Loss Function describes the loss to the customer for deviation from the target values. American specifications call for a pass/fail test for conformance. Taguchi shows that any deviation from target is a loss to the customer, even an increase in quality if it comes at a price that is higher than the customer wants to pay. Taguchi uses a loss curve to establish the loss to the customer. The on-target loss is zero. The costs as the product moves away from target are based on tangible costs such as warranty costs. The curve can be fitted to pass through such identifiable cost points. The objective of the method is to minimize loss to the customer.

SE minimizes losses by selecting a low-cost system design. The key parameters that allow the least variation in the presence of noise are identified using experiments, usually in orthogonal arrays. The levels of the parameters are set for least variation, again using orthogonal arrays as previously described. The results are confirmed before engineering release. Concentrating on the "vital few," only those parameters that can be controlled in a cost-effective manner are used. The designer has to find solutions to quality and cost problems caused by many factors, including those about which he knows nothing. Statistics are used to analyze the main parameters to determine how to use of their interactions to minimize the effects of unknown causes. Mathematicians fault Taguchi methods as not mathematically rigorous. Taguchi's response is that engineering differs from science, using problem-solving short cuts to get practical, not perfect answers.

The Taguchi method requires low cost as a precondition to any increase in quality. Dr. Taguchi believes that price is the primary arena of competition. Even perfect quality cannot compete if the price is too high. His three-step process to producing a product is: a) design to lower product cost; b) improve quality as much as possible through parameter design (adjusting parameters for best combination of robustness and quality); and c) perform tolerance design (similarly adjusting tolerances) as necessary. Steps b and c allow the true costs of quality to be calculated. From these data it is possible to determine the best quality obtainable at the lowest cost. Taguchi considers the three steps in the engineering of both the product, and the manufacturing system to build the product.

In engineering the manufacturing system steps for the product are:

- **System design –** selecting the manufacturing processes from available technology.
- **Parameter design –** establishing the operational conditions, including materials and purchase parts sources.
- **Tolerance design –** setting the tolerances of the process conditions and sources of variability.

The results of the Taguchi methods have also been proven in the market place and are a potent SE tool for cost reduction and increased customer satisfaction.

5.15 Process Capability Models (CMMI®)

Process capability models are often used when well defined processes and process maturity influences the outcome of a development or production effort. Surely in the business of weapon systems, development having well defined and mature processes is critical to success.

There are a number of process capability models that have come into use over the last 30 years that include ISO9000 and Capability Maturity Model – Integrated (CMMI®). Here we limit our discussion to CMMI® developed by Carnegie Mellon's Software Engineering Institute (SEI).

Figure 5-11 CMMI® model components

The premise underlying the CMMI is that, if an organization that develops systems retains organizational maturity in controlling and managing software and hardware development efforts, that organization retains low risk to develop and deliver the products within cost. There are three core categories of CMMI® Process Areas that include Process Management, Project Management, and Engineering Support. Within each process area, goals and practices are defined as reflected in Figure 5-11. SEI publishes and maintains a large library of technical documents and guidance on its websites that can be consulted for a more detailed and thorough discussion of process areas and other CMMI® concepts.

There are five levels of maturity associated with the CMMI® model.

- **Level 1 (Initial) –** The process is ad hoc and chaotic and depends on individual efforts. There are neither project plans nor formal procedures. Change control is limited or lacking. Senior management is not aware of software development issues.
- **Level 2 (Managed) –** Basic project controls are in place to repeat project successes. Projects have planned processes and executed in accordance with policy; employed skilled people who have adequate resources to produce controlled outputs; are monitored, controlled and reviewed; involved stakeholders. Projects are performed and managed according to their documented plans. Edward Yourdon[35] suggests the following processes for software: software Planning, software cost estimating, configuration management, and management commitment.
- **Level 3 (Defined) –** Organization wide software development processes are standardized. An Engineering Process Group is in place. Yourdon recommends that the following criteria are necessary to achieve Level 3: formal standards, formal process models, formal processes for testing, inspections, configuration control, and establishment of an engineering process group.
- **Level 4 (Quantitatively Managed) –** This level emphasizes detailed quantitative methods to measure product and process quality. Objectives are based on the needs of the customer, end users, organization, and process implementers. In other words, an emphasis is placed in statistical terms and is managed throughout the life of projects.
- **Level 5 (Optimized) –** its business objectives and performance needs. The project's defined processes, the organization's set of standard processes, and supporting technology are targets of measureable improvement activities.

---

[35] Yourdon, Edward, Decline & Fall of the American Programmer, Englewood Cliffs, New Jersey, 1993

Based on surveys and assessments, SEI estimates that approximately 80% of the software development organizations are at level 1. This model may very well be a solid indication of software development risks. However, it is under discussion that measures of personnel capability and performance are also important to identify and assess potential risks.

5.16 Prototyping

A prototype is an early implementation of a system, component, or process to evaluate its technical feasibility to meet expected capability performance or suitability. Physical or virtual models are built to demonstrate (i) proof of principle, (ii) comparative assessment of alternate concepts, or (iii) form, fit, and function of the final design. Prototyping is used as a tool for:

- management of risk, uncertainty, and complexity
- technology maturity assessments,
- assessment and comparative performance of alternative concepts and designs,
- requirements development or refinement,
- assessment of technology maturity
- identification and resolution of integration risks
- assessment of manufacturing and sustainability risks
- minimizing cost growth due to unknowns in design, assembly and integration

Prototyping is typically employed during the early phases of the DAS process. DoD also employs Joint Capability Technology Demonstration (JCTD) and Advanced Technology Demonstration (ATD) processes to assess specific design or manufacturing risks. SPOs need to include prototyping needs to the subsystem level with both performance goals and expected cost of prototyping in their Technology Development Strategy (TDS).

5.16.1 Competitive Prototyping

DoD and other organizations use competitive prototyping, an acquisition tactic where two or more competing teams are asked to produce prototypes during the early stages of a project (MSA, TD). The prototype that satisfies stated measures of effectiveness best is chosen for further development. DoD requires competitive prototyping to assess technology maturity or program risk[36]. OMB identifies several advantages to competitive prototyping as a valid tool to mitigate acquisition risk that include[37]:

- Proves concepts are sound
- Allows efficient and effective communication among stakeholders to identify operational needs against market capabilities
- Provides for competition during the development effort
- Ensures focused technology development
- Facilitates firm fixed-price contracting for production

5.17 Requirements Analysis and Allocation Tools

Chapter 1 provides a list of characteristics for a good requirement. It is an important SE task to establish a structured requirements development process, and maintain a requirements trail that traces the pedigree of every allocated and derived requirement to the lowest level. Surely, somewhere along the line someone in

---

[36] Office of the Under Secretary of Defense for Acquisition, Technology and Logistics, September 19, 2007, Memorandum on Prototyping and Competition, Washington, DC: Pentagon.

[37] Office of Management and Budget, June 2006, "Competitive Prototyping," section II.3.3 of Capital Programming Guide, V 2.0, Supplement to OMB Circular A-11, Part 7

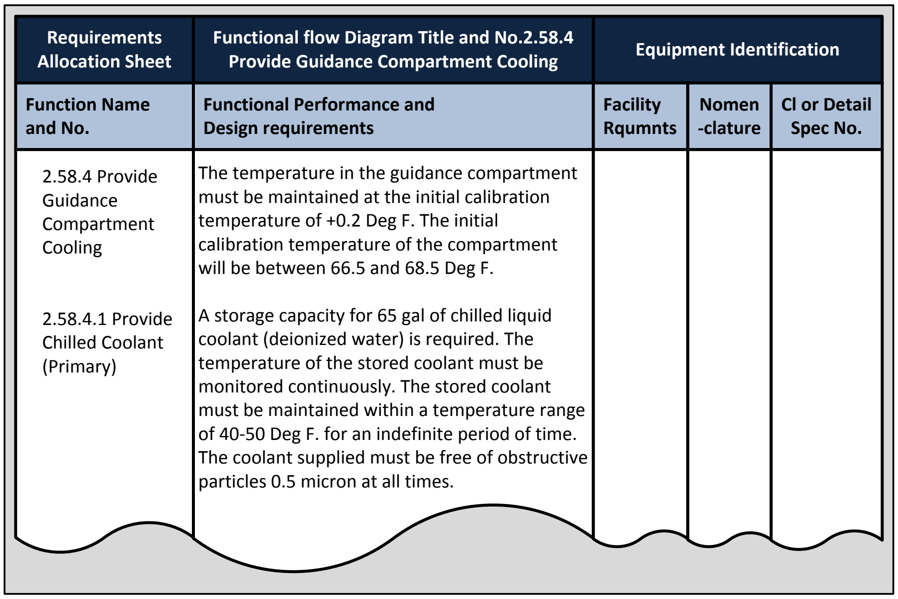
the design/production chain is going to question the need for a particularly sticky requirement that does not have to be met. One may be right! But if it is the case, unless one can trace the requirement, it is not safe in granting relief to determine its origin. Then too, this may not be the case. Likewise, without a secure guide, extraneous requirements tend to creep in when someone thinks it would be a "good idea," or "the way we did it last time." Traceability tools help alleviates this problem.

Such tools usually employ relational databases. A number of commercial and government developed tools are available to record, maintain, and trace system requirements and related technical data to include test and analysis methods needed to verify and validate design requirements.

As the system evolves from the top down, requirements, specifications, and constraints are attributed to each portion of the lower-level requirements and recorded in the database. Related trade studies, research, and analyses that lead to derived requirements are also registered. As the system design matures, designers and production management can validate or challenge any requirement. In this way, only those requirements that contribute to mission performance affect final design.

5.17.1 Requirements Allocation Sheet

Requirements Allocation Sheet is a simple but effective tool for requirements development and analysis. It documents a requirement's relationships between allocated functions, allocated performance, and the physical system. It records traceability between functional analysis, functional allocation to lower levels, and eventual design synthesis. It offers a structured tool that can help identify disconnects to maintain consistency between functional architectures and designs that are based on them.



Requirements Allocation Sheet	Functional flow Diagram Title and No.2.58.4 Provide Guidance Compartment Cooling	Equipment Identification		
Function Name and No.	**Functional Performance and Design requirements**	**Facility Rqumnts**	**Nomen -clature**	**CI or Detail Spec No.**
2.58.4 Provide Guidance Compartment Cooling	The temperature in the guidance compartment must be maintained at the initial calibration temperature of +0.2 Deg F. The initial calibration temperature of the compartment will be between 66.5 and 68.5 Deg F.			
2.58.4.1 Provide Chilled Coolant (Primary)	A storage capacity for 65 gal of chilled liquid coolant (deionized water) is required. The temperature of the stored coolant must be monitored continuously. The stored coolant must be maintained within a temperature range of 40-50 Deg F. for an indefinite period of time. The coolant supplied must be free of obstructive particles 0.5 micron at all times.			

Figure 5-12 Example Requirements Allocation Sheet (Ref. Systems Engineering Fundamentals, Defense Acquisition University Press, January 2001)

Figure 5-12 shows an example RAS. Each of the functions, developed as part of the functional analysis using FFBD, IDEF0, or other tools, is listed. These functions are then connected to the functional and design requirements allocated all the way to the piece-parts and design specifications as necessary. All functions must be represented in the allocated requirements and the design.

5.18 Risk Analysis

Effective and efficient implementation of system engineering in space systems acquisitions demands active risk management. Without it, programs can easily fall prey to continuous crises action and become a resource drain which compromise core engineering functions. Risk management avoids this situation by investing in plans, processes, and decision making to increase the probability of successful attainment of engineering goals within the constraints of cost, schedule, and performance expectations.

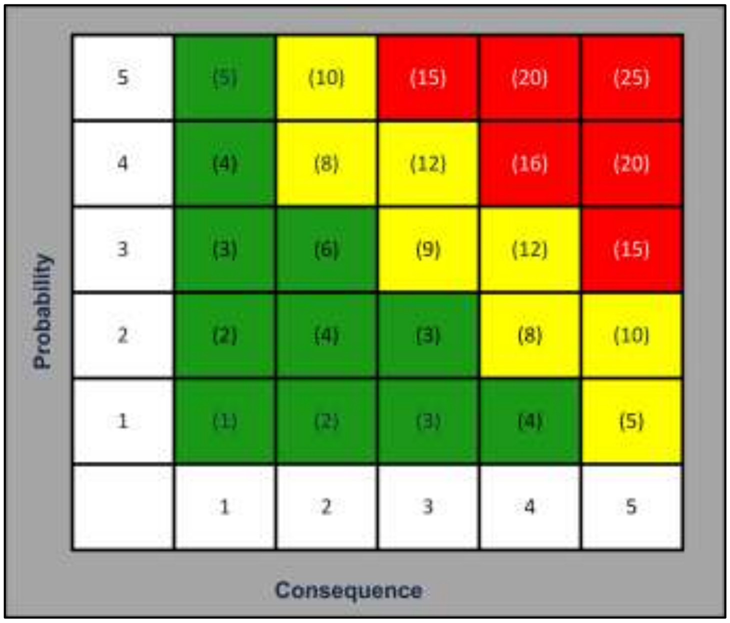
Carnegie-Mellon University's SEI defines risk analysis and management as the process that helps "…identify potential problems before they occur so that risk-handling activities can be planned and invoked as needed across the life of the product or project to mitigate adverse impacts on achieving objectives."

Risk management is, in effect, an engineering management tool for preventive action against foreseeable issues. Planning for the management of engineering risks is captured in individual program risk management plans and operating instructions. Documents such as the SMC Risk Management Process Guide provide a framework for the development and implementation of those documents.

The basic components of risks, as stated in DoD Risk Management Guidebook, consists of:

- A future root cause (yet to happen), which, if eliminated or corrected, would prevent a potential consequence from occurring,
- A probability (or likelihood) assessed at the present time of that future root cause occurring, and
  The consequence (or effect) of that future occurrence.

Key to risk analysis is the identification of the specific item which drives the existence of the risk, and that which may become the "future root cause" of a problem. Merely identifying threats may not be sufficient, since threats alone may only be indicators of symptoms, not of an underlying driver. A simple test for validating a root cause driver for a risk is if it is eliminated, whether the potential problem and their commonly rooted associated problems would go away. Note that a single driving root cause can have multiple symptoms. This is the reason why it is important to identify the root driver and refrain from chasing its symptoms or resultant threats. A technique useful in future root cause analysis is to layout the future root causes and their interrelationships in a Fishbone/Ishikawa diagram. This will help identify key drivers of risks and help focus handling efforts.



5	(5)	(10)	(15)	(20)	(25)
4	(4)	(8)	(12)	(16)	(20)
3	(3)	(6)	(9)	(12)	(15)
2	(2)	(4)	(3)	(8)	(10)
1	(1)	(2)	(3)	(4)	(5)
	1	2	3	4	5

Figure 5-13 USAF risk probability-consequence diagram

The probability of risk is that likelihood for the risk realizing itself, in other words, the chance of it becoming a problem. Quantitative databases and validated statistics should be leveraged to the extent meaningful to ascertain the likelihood of risk realization. Absent a hard quantitative basis for probabilities, an informed subjective assessment must be rendered. In such cases, the insights gained from subject matter experts, lessons learned, trend analyses, and experience in similar past situations might be used.

Similarly, for determining the impacts of when a risk realizes itself, quantitative data and leveraged insights should be utilized. To facilitate decisions on potential cost-benefits, the impact assessments are done in terms of the programmatic or engineering factors managed, e.g., cost, schedule, and performance.

Figure 5-13 shows the mandated Air Force risk probability-consequence diagram for SMC programs. Risks are placed in the color-coded (low=green; medium=yellow; high=red) locations. The numbers in the parentheses indicate the relative weight of the risk. Higher consequence and probability risks have higher numbers. The weighting is allows for risk prioritization and select management/engineering visibility and action. The standard (Air Force mandated) definition of the probability and consequence variables are published in the SMC Risk Management Guide.

Once a risk is assessed for its probability and impact, it must be "handled." Risk handling strategies fall in four general categories: assumption, avoidance, transference, and mitigation. Assumption is the acceptance of the risk, as is, without special action, except perhaps monitoring. Risk assumption decisions are based on the level of program risk tolerance—how much risk a program is willing to take. Avoidance is simply changing the basis of the risk's existence, so that by a change in initial conditions, it no longer poses a risk. An example is eliminating a requirement where its implementation poses a risk. Transference is moving the ownership of the risk outside the scope of the managing organization's area of responsibility. Though this might pass the risk to another organization, it does not eliminate the risk. Mitigation is taking active measures against the risk; reducing the probability and/or the severity of impact. Figure 5-14 summarizes the Risk Management Process Steps.

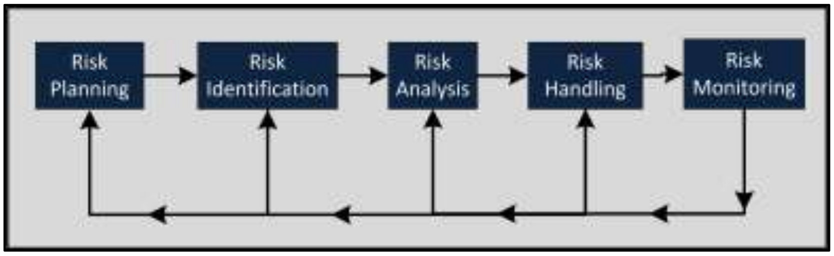


Figure 5-14 Risk Management Process Steps

To ensure risk mitigation steps are resourced, have the required management visibility and that there is due accountability, they should be entered into a formal schedule, such as an integrated master schedule (IMS). Also, since risks within an enterprise can be related, the linkages between risks and their association ought to be captured, especially as they are "rolled-up" to higher levels of engineering management.

The last key step is monitoring. To ensure desired effects are achieved, new conditions and progress against the risk must be reviewed; and any underlying assumptions re-validated for continued applicability. Findings will determine whether and how the risk needs to be re-worked.

The standard risk management tool at SMC which helps programs implement the process, and capture their risks and the associated metrics is Active Risk Manager (ARM).

While risk management is often treated as a separate entity in itself, the idea of problem prevention—the essence of risk management—is already an integral part of several program disciplines. Engineering tasks which take on calculated risks, and avoid, transfer, or mitigate against future problems are exercising a form of risk management. Inherent risk activities within program system engineering, for example, include: Reliability, Maintainability, Availability (RAM); Diminishing Manufacturing Sources and Material Shortages (DMSMS); Integrated Baseline Reviews (IBRs); PMRs/Milestone Reviews; FRACAS/FMECA; Test and Evaluation (T&E); Manufacturing and Producibility; Parts, Materials and Processes (PMP);

Spectrum Management; Architecture Engineering; System Safety; System Protection /Program Security; Information Assurance (IA); Prognostics and Health Management; and Quality Assurance.

As a general rule, engineering plans, processes, and decision making are required to have built-in features of risk management. Risk management when effectively employed, increases the engineering project's chances of success.

5.19 States and Modes Analysis

States and Modes analysis provide a means to identify different sets of conditions that will be encountered by the system/element, and the corresponding sets of performance requirements that the system/element must meet for each of them. They are only useful if they help clarify what performance is needed or expected. As with other SE terms used in this handbook, definitions and examples for the terms state and mode are provided below (Source: James Martin's Systems Engineering Guidebook):

- State: The condition of a system or subsystem when specific modes or capabilities (or functions) are valid.
  For example, states of a system may include Off, Start-up, Ready On, Deployed, Stored, and In-Flight.
- Mode: The condition of a system or subsystem in a certain state when specific capabilities (or functions) are valid. Each mode may have different capabilities defined. For example, of modes within the Ready state may include Normal, Emergency, Surge, Degraded, and Reset.

From the above definitions, it should be noted that according to this interpretation, modes are included within states. This is the most common and accepted relationship. However, the reverse convention is sometimes used. The important point is to be consistent in the use of the terms within the proper context.

States and modes identify different sets of performance requirements for different sets of conditions that may be encountered by the system. It may not be obvious, but once states and modes are introduced, it is imperative that all the performance requirements for each mode (within each state) be delineated. Often the specification developer only thinks in terms of the requirements that may have driven him/her to identify the mode in the first place, and neglects to consider all the other requirements that would need to be performed in that mode. For example, while concentrating on the key requirements for the Autonomous Mode, the ability to receive, interpret, and execute commands needed to transition out of the mode may be overlooked. This is another instance of the "tip of the iceberg" approach that is seen all too often. The danger of not explicitly stating all the performance requirements for each and every state/mode should be readily apparent. If the requirement isn't clearly delineated, the finished system/element won't perform as expected.

Remember that once states and modes are introduced, all the performance requirements must be included within the states/modes structure; there cannot be any performance requirements that are not associated with at least one state/mode combination. Put another way, performance requirements cannot exist outside the state/mode structure. If the states/modes defined cannot include all the performance requirements, there is something fundamentally wrong with that set of states and modes, and they should be revised. In some instances, it may be that requirements that appear to exist outside the state/mode structure are really common to all states/modes, or common to some subset of the states/modes. If either is the case, it should be clearly stated that the requirements are common to whatever states/modes that share them. The author may know that the requirements are common to all or some subset of all and assumes everyone else would also. Such an assumption does not facilitate clear understanding of what the system/element is supposed to do. One shortcut sometimes employed to implement states and modes is, instead of organizing the performance requirements within the state/mode structure; a matrix is included in the specification that indicates the states/modes applicability for each performance requirement. That procedure does convey the information, but not as clearly as having all the requirements for a given mode in one place.

The use of states and modes in system level requirements documents probably came into widespread use as a result of Data Item CMAN 80008A. This was the document that specified the format, content, and structure for A-Specs (system and segment level specs). However, trying to apply states and modes to an entire system may not have been a great idea. Often, while states and modes may make sense for a subsystem or element of a system, they would be difficult to apply (or meaningless) to the entire system. Although no longer mandated, some engineers still use states/modes within their requirements documents. If states and modes are going to be used, the following structure prescribed by CMAN 80008A is still a good one to follow:

- 3.2.1 Performance Characteristics
  - 3.2.1.1 State 1 Name
    - 3.2.1.1.1 Mode 1 (within State 1) Name
      - 3.2.1.1.1.1 Performance Capability (1)
      - 3.2.1.1.1.n Performance Capability (n)
    - 3.2.1.1.2 Mode 2 (within State 1) Name
      - 3.2.1.1.2.1 Performance Capability (1)
      - 3.2.1.1.2.n Capability (n)
    - 3.2.1.1.n Mode n (within State 1) Name
      - 3.2.1.1.n.1 Performance Capability (1)
      - 3.2.1.1.n.n Performance Capability (n)
  - 3.2.1.2 State 2 Name
    - 3.2.1.2.1 Mode 1 (within State 2) Name
      - 3.2.1.2.1.1 Performance Capability (1)
      - 3.2.1.2.1.n Performance Capability (n)

In practice, the actual performance requirement title would replace "Performance Capability (n)" in the above outline. It should be readily apparent the intent of CMAN 80008A was to define all performance functions/capabilities within the structure of the states and modes. Even though CMAN 80008A may no longer be the governing directive for A- Specs, the concepts it put forth regarding states and modes are still valid.

It is not uncommon for performance requirements to be applicable to more than one mode. A satellite operating in its Autonomous Mode would perform many (but not necessarily all) of the same functions that it would in its Normal Mode. In addition, it may perform some functions in the Autonomous Mode that it does not perform in its Normal Mode. Where capabilities/ requirements existed in more than one mode, CMAN 80008A prescribed identifying the performance requirement by title and referring back to the first appearance of the capability/requirement for the actual text, rather than repeating it.

Care must be exercised in considering transitioning between modes. It may not be necessary/possible to transition from each and every mode to each and every other mode. Allowable/ required transitions need to be specified. It is also necessary to consider that the transitioning begins from the current mode. Transitioning from the Autonomous Mode into the Normal Mode would be a function/capability required of the Autonomous Mode. The satellite is not in the Normal Mode until the transition is completed, so transitioning into the Normal Mode is not a capability, function, or requirement of the Normal Mode.

5.20 SysML

SysML™[38] is a critical enabler for model driven SE. It is based on and extends UML for SE as introduced in Chapter 1. In a complex and rapidly changing environment, SysML™ offers a model-driven approach to SE for cost-effective scalable solutions. SysML™ provides artifacts for modeling structure, behavior, requirements and performance characteristics of a system. A number of appropriate SysML™ structural and behavioral views of the system are available for various stages of the system lifecycle form requirements definition, to analysis and design, implementation, testing and deployment. Figure 5-15 shows SysML™ artifact categories as an extension of the UML. SysML™ provides for two new and three modified diagrams. These are discussed below. The standard UML diagrams, adopted by SysML™ without modifications, are discussed in the UML section in this chapter. (Note: SysML™ uses lowercase acronyms for its artifacts.)

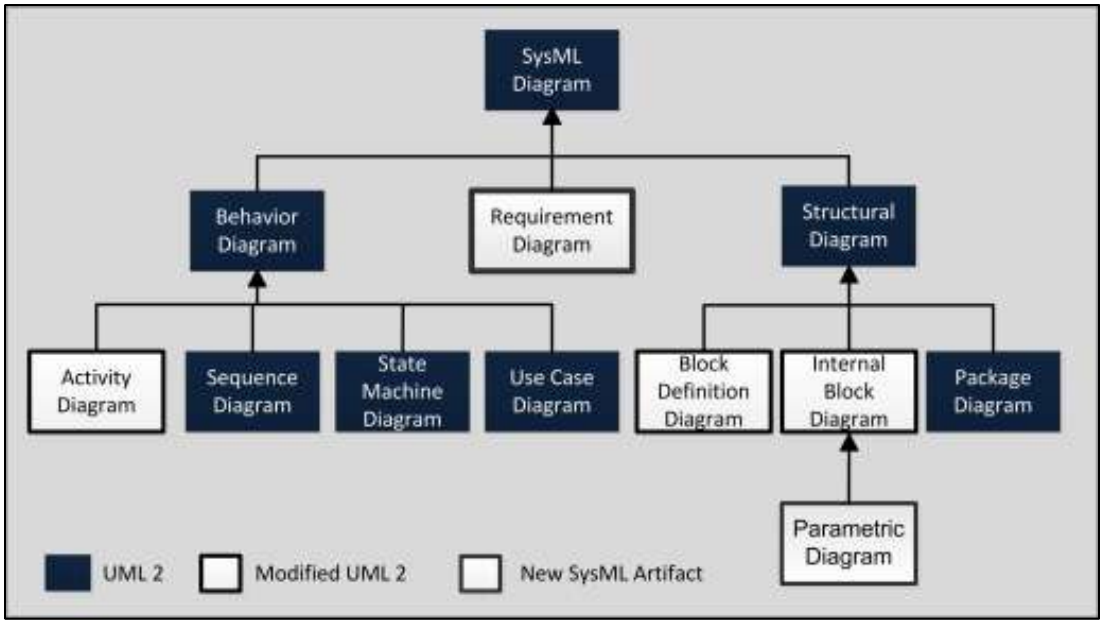


Figure 5-15 Complete set of SysML diagrams and how they are related to the UML

- **Requirements Diagram (rd) –** describes all the product functions, as well as the constraints under which these functions should be realized based on the analysis of stakeholder needs that follow the concept development stage. SysML™ allows the representation of requirements as model elements that can be related to other modeling elements. The requirements diagram depicts the requirements in textual, graphical, tabular, or tree structure format with their unique identifiers. Requirement properties and relationships to include verification methods and status, traceability, and cross relationship to other

---

[38] Latest SysML™ specification is available at the Object Management Group's (OMG) website

requirements is represented on the diagram. Requirements diagram contains both functional and non-functional requirements. It can also be used to depict modeling of test cases without restriction. A parent requirement is a package for the embedded or allocated or derived requirements. In that sense, deleting the parent requirement will automatically delete all the embedded ones. Requirements are elicited during the entire product lifecycle, and additional Requirements diagrams are used to represent them. Hence, the product requirements are typically laid out on a set of requirement diagrams. One of the important consequences of having requirements as model elements is that it allows the designer to specify which components in the system satisfy a given set of requirements.

- **Parametric Diagram (pd) –** is used to represent the usage of constraint blocks as constraint properties. Syntactically, the pd is similar to the ibd. In a pd, constraint properties are connected to each other through the parameters defined in the constraint block of which they represent a usage. In turn, they connect to other properties (block properties, distributed properties, etc.) in the context of their parent block. These other properties must be directly bound to parameters of the constraint properties because they can only be used as input for their parameters.
- **Block Definition Diagram (bdd) –** provides a basic structural element whose aim is to provide a discipline-agnostic building block for systems. Blocks can be used to represent all types of system components; e.g., functional, physical, human, etc. Blocks assemble to form architectures that represent how different elements in the system coexist. The SysML™ bdd is the simplest way to describe the structure of the system. It is the equivalent to the Class diagram in UML. It is used to represent the system decomposition using, for example, associations and composition relationships. The bdd is ideal for displaying the features of a block, such as its properties, and operations. SysML™ allows blocks to own special types of properties: block properties and distributed properties. Block properties impose additional constraints on classic UML properties that may include physical units or dimensions. Distributed properties let the user apply a probability distribution to the values of the property. SysML™ proposes model libraries for possible values of units, dimensions, and probability distributions. The operations (sometimes called services) represent the functional aspects of the system.
- **Internal Block Diagram (ibd) –** The ibd allows the designer to refine the structural aspect of the model. The IBD is the equivalent of the composite structure in UML. In the ibd, properties (or parts) are assembled to define how they collaborate to realize the behavior of the parent block. A part represents the usage of another block. The most important aspect of the ibd is that it allows the designer to refine the definition of the interaction between the usages of blocks by defining ports. Ports are parts available for connection from outside the owning block. Ports are categorized according to type by the interfaces or blocks that define what can be exchanged through them. Ports are connected using connectors that represent the use of an association in the ibd. Two types of ports are available in SysML™: Standard ports handle the requests and invocations of services (i.e., function calls) with other blocks, and flow ports let blocks exchange flows of information or material. For Standard ports, an interface class is used to list the services offered by the block. For flow ports, a flow specification is created to list the type of data that can flow through the port. When only a single type of object can flow through a port, then the object's type is directly assigned as the port's type.
- **SysML™ Activity Diagram (act) –** leverages and extends the activity model from UML to support continuous systems. The SysML™ Activity Diagram offers many innovations. The modeling of activities in SysML™ consists of describing behavior as a flow graph. An activity is defined as a set of actions represented as graph nodes linked by edges carrying control flow and data/item flow between actions. Object nodes represent a container for the type of data that can flow through the graph. Control nodes are used to route control and data/item flows through the activity. Activities can be related to each other to represent, for example, functional decomposition in a similar way as blocks represent structural decomposition in a bdd, or ibd. Activities can also be associated to classifiers when the latter are used as a type of object node. These diagrams can be produced using (extended) FFBDs.

A number of comprehensive open-source, commercial, and government developed tools are available that can be used to produce and organize SysML™ diagrams.

5.21 Technical Performance Measurement

Technical performance measurement (TPM) is based on a set of tools that help define a set of critical quantitative performance parameters for a system with expected performance and then tracked over time. Actual value of these critical capabilities is then measured over the lifecycle of the system and compared to the projected values to assist in decision making. TPM offers the manager an early warning to possible shortfalls in system performance and its impact on operations.

TPM is an evolutionary program management tool that builds on the two traditional parameters of Earned Value Management and cost and schedule performance indicators. A third dimension is also added – the status of technical achievement. By combining cost, schedule, and technical progress into one comprehensive management tool, program managers are able to assess the progress of their entire program. TPMs are typically established on those programs complex enough where the status of technical performance is not readily apparent. TPMs can also be valuable for risk monitoring – levels below that forecast can indicate the need for an alternate approach.

With a TPM program it is possible to continuously verify the degree of anticipated and actual achievement of technical parameters and compare with the anticipated value. TPM is also used to identify and flag deficiencies that might jeopardize meeting a critical system level requirement. Measured values that fall outside an established tolerance band will alert management to take corrective action. Relevant terms and relationships are illustrated in Figure 5-16.

By tracking the system's TPMs, the manager gains visibility into whether the delivered system will actually meet its performance specifications (requirements). Beyond that, tracking TPMs ties together a number of basic SE activities. That is, a TPM tracking program forges a relationship among systems analysis, functional and performance requirements definition, and verification and validation activities:[39]

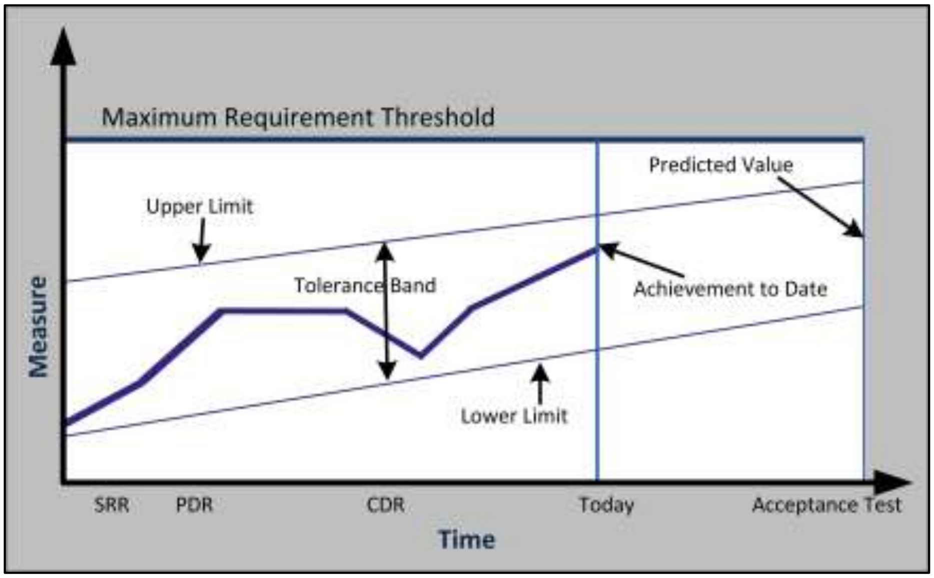


Figure 5-16 Performance measures tracked over time

- Systems analysis supports the quantification of the system's functional requirements; Systems analysis activities identify the key performance or technical attributes that determine system effectiveness

---

[39] NASA Systems Engineering Handbook SP 6105, June 1995.

- Functional and performance requirements definition activities help identify verification and validation requirements.
- Verification and validation activities result in quantitative evaluation of TPMs
- "Out-of-bounds" TPMs are signals to replan fiscal, schedule, and people resources; sometimes new systems analysis activities need to be initiated.

TPMs are identified and tracked to determine the progress of systems development. This progress tracking includes incremental measures to assess the probability of meeting the objectives as well as specific measures to determine reliability, maintainability, availability, survivability, testability, safety, electromagnetic properties, weight, balance, and manufacturability. TPMs are typically derived directly from measures of performance (MOPs) to characterize physical or functional attributes relating to the execution of the mission or function. TPMs may also be derived from MOEs to become system cost and effectiveness metrics.

Some guidance for selecting TPMs:

- Selected for their ability to provide actionable information
- Performance parameters that significantly qualify the entire system
- Parameters are directly derived from analyses, demonstrations, or test
- A direct measure of value can be derived from results of analyses or tests
- Predicted values have a basis (analyses, historical data)
- Each parameter can periodically be measured and profiled to compare with predicted values and tolerances over the project life cycle.

The most important process in TPM planning is the development of Technical Parameter Hierarchy, which requires the establishment of the “technical performance baseline”. The technical performance baseline identifies all measurable key technical elements and establishes their relative relationships and importance. The hierarchy can be representative of the program, contract, sub-contract or other subset of technical requirements. The hierarchy must comprehensively represent technical risk factors associated with the project. Typically, the highest level of the hierarchy represents system level or operational requirements with sub-system level requirements underneath these as lower level parameters. This form of TPM methodology not only serves internal tracking by the SE managers but also adds visibility of program status reporting.

The most important process in TPM planning is the development of Technical Parameter Hierarchy, which requires the establishment of the “technical performance baseline”. The technical performance baseline identifies all measurable key technical elements and establishes their relative relationships and importance. The hierarchy can be representative of the program, contract, sub-contract or other subset of technical requirements. The hierarchy must comprehensively represent technical risk factors associated with the project. Typically, the highest level of the hierarchy represents system level or operational requirements with sub-system level requirements underneath these as lower level parameters. This form of TPM methodology not only serves internal tracking by the SE managers but also adds visibility of program status reporting. The hierarchy example below is not by any means exhaustive. Much consideration must be given to select the appropriate TMPs for each program.

5.21.1 TPM Hierarchy Examples

Top - Level TPMs for Satellites and Launch Vehicles:

- Top - level technical performance measures (TPMs) for satellites include:
    - End-of-mission (EOM) dry mass
    - Injected mass (includes EOM dry mass, baseline mission plus reserve propellant, other consumables and upper stage adaptor mass)
    - Consumables at EOM
    - Power demand (relative to supply)
    - Onboard data processing memory demand
    - Onboard data processing throughput time
    - Onboard data bus capacity
    - Total pointing error
- For launch vehicles, top - level TPMs include:
    - Total vehicle mass at launch
    - Payload mass (at nominal altitude or orbit)
    - Payload volume
    - Injection accuracy
    - Launch reliability
    - In-flight reliability
    - For reusable vehicles, percent of value recovered
    - For expendable vehicles, unit production cost at the n^{th} unit
- System and sub-System Level TPMs for Satellites and Launch Vehicles
    - System Level TPMs for Satellites
        - Space Segment
        - Bus Assembly Measures
        - Thermal Control Measures
        - Power System Measures
        - Payload Assembly Measures
        - Sensor Performance Measures
        - Sensor Processor Measures
        - Hardware Measures
        - Software Measures
    - Ground Segment
        - Ground Control Station Measures
        - Support Equipment Measures
- System Level TPMs for Launch Vehicle
- Launch Segment
    - Booster (Stages I, II, III, etc.) Measures
    - Solid Rocket Motors (SRMs)

  - Liquid Motors
  - Fairing Measures
  - Guidance and Control Measures
  - Integration and Assembly Measures
  - Test and Checkout Measures
- Ground Segment
  - Telemetry, Tracking and Control Measures
  - Ground Vehicle Database Measures
  - GC3ME Measures
  - GSE Measures
  - Facilities Measures
- Technical Performance Measures that Impact Supportability
  - Maintenance Personnel
  - Maintenance Man-hours Per Hour of Operation
  - Average Skill Level Required
  - Number of Special Skills Required
  - Number of qualified vendors per component/part
  - Number of sole source drawings
  - Number of altered item drawings
- Technical Performance Measures Impact Time To Reconstitute Force
  - Cost of Reconstitution
  - Weapon System Unit Cost
  - Mean Cost to Remanufacture
  - Manufacturing Time
  - Long-Lead Time
  - Time to Manufacture/Assemble
  - Interchangeability
  - Mean Time to Remanufacture
  - Service Life

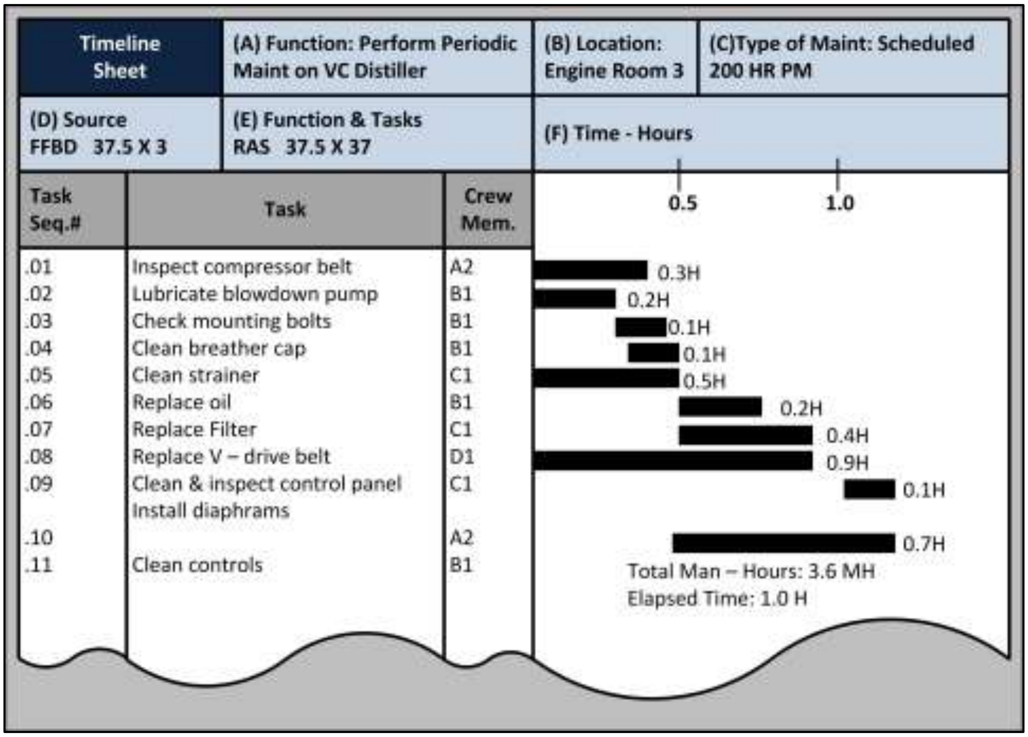
5.22 Timeline Analysis

Time-line analysis supports developing requirements for the product operation, test, and maintenance. The analysis shows:

- Time-critical paths,
- Sequences,
- Overlaps, and
- Concurrent functions.

Time-critical functions affect reaction time, downtime, or availability. Performance parameters can be derived, in part, from time-critical functions. Figure 5-17 is a sample time-line sheet for a maintenance function and illustrates that functional analysis applies to support systems as well as the prime product. Furthermore, SysML™ and UML provide timing, sequence, and activity diagrams that can be used for timeline analysis. Timeline analysis within the SysML™ or UML framework can then be easily related to other viewpoints for a more comprehensive understanding of the system design.

For simple products, most functions are constant and have a fixed relationship to their physical components. This is not the case in more complex products. Here, functions are variables with peak demands and worst-case interactions. The time-line analysis is valuable in identifying overload conditions. A matrix of function needs versus component capabilities to perform the functions can be constructed. The matrix is best left to the analysis activities after the functions have been identified.



Timeline Sheet	**(A) Function: Perform Periodic Maint on VC Distiller**	**(B) Location: Engine Room 3**	**(C)Type of Maint: Scheduled 200 HR PM**
(D) Source FFBD 37.5 X 3	**(E) Function & Tasks RAS 37.5 X 37**	**(F) Time - Hours**	

Task Seq.#	Task	Crew Mem.	Time - Hours
.01	Inspect compressor belt	A2	0.3H
.02	Lubricate blowdown pump	B1	0.2H
.03	Check mounting bolts	B1	0.1H
.04	Clean breather cap	B1	0.1H
.05	Clean strainer	C1	0.5H
.06	Replace oil	B1	0.2H
.07	Replace Filter	C1	0.4H
.08	Replace V – drive belt	D1	0.9H
.09	Clean & inspect control panel	C1	0.1H
	Install diaphrams		
.10		A2	0.7H
.11	Clean controls	B1	

Total Man – Hours: 3.6 MH
Elapsed Time: 1.0 H

Figure 5-17 Timeline Analysis Sheet shows sequence of operational and concurrent action

5.23 Trade Studies

Trades are performed throughout the concept definition, development, and design phases to select operational concepts, originating capabilities and requirements high level system architecture, systems functions and requirements, and design solutions. For space systems the focus of trade studies is to perform objective trade comparisons of all reasonable alternatives and to choose the alternative that best balances performance, cost, schedule, and risk. (We might add safety, reliability, weight, and other constraints.) Also for space systems, the trade study process is often controlled using models.

The INCOSE Systems Engineering Handbook[40] states the purpose of trade studies is to provide an objective foundation for the selection of one of two or more alternative approaches to solution of an engineering problem. The trade study may address any of a range of problems from the selection of high-level system architecture to the selection of a specific COTS processor.

Dennis Buede[41], author of Engineering Design of Systems, defines a trade study as analysis that focuses on ways to improve systems performance on some highly important objective while maintaining system's capability in other objectives. Trades studies, on the other hand, are analyses that focus on comparing a range of design options from the perspective of the objectives associated with the system's performance and cost.

The DSMC Systems Engineering Fundamentals[42] describes a trade as a formal decision making methodology used by integrated teams to make choices and resolve conflicts during the SE process.

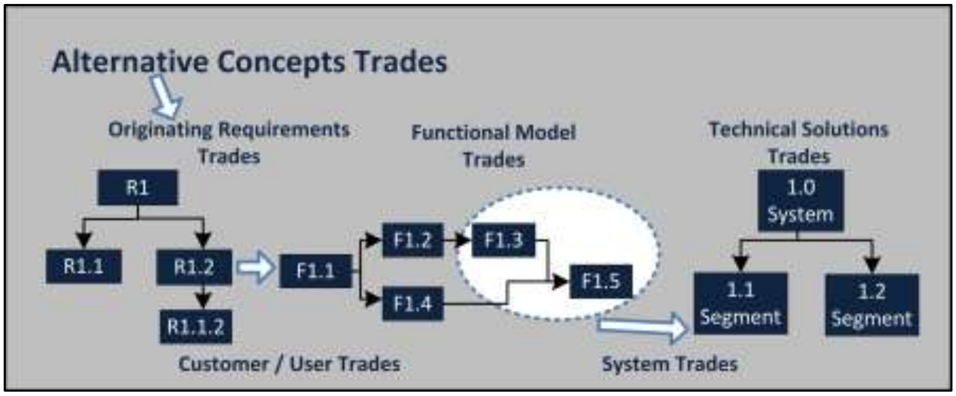


Figure 5-18 Trade studies to resolve architectures, requirements, and functional & design solutions

For the SE process, trades performed during requirements analyses initiate the recursive process to determine the optimal choices of system functions and performance requirements. As depicted in Figure 5-18, trade studies are performed within and across requirements and functions to support the functional analyses and allocation of performance requirements. Trades are also used to evaluate alternative functional architectures and to determine performance requirements for lower-level functions when higher-level performance and functional requirements cannot be readily decomposed to the lower level.

Trade studies often use various scoring, utility curve, sensitivity analyses, TPMs, or other value design methods to to select the best possible solution as discussed in Section 5.28.

Highlight Box 5-2
Typical Trade Study Outline

1. Purpose of Study – resolves technical or management issues or performs a comparative analysis

2. Scope of Study – level of detail, assumptions, and requirements

3. Trade Study Description – describe and introduce the study and its methodology, schedule, and cost

4. Analytical Approach – includes (i) candidate solutions, (ii) measures of performance, (iii) selection criteria, (iv) specialty engineering considerations, (v) form, fit, and function, (vi) sensitivity analysis, and (vii) other relevant factors

5. Trades Results – include (i) summary discussion of trade space and concepts studied, (ii) selected user/operational concept, (iii) selected system architecture, (iv) derived requirements, (v) allocated requirements, (vi) derived technical solutions, (vii) cost, schedule, and risk analysis

5.24 Unified Modeling Language (UML)

UML was introduced in Chapter 1 as a model and tool for SE, especially when a system has a strong software

---

[40]. The INCOSE Systems Engineering Handbook, July 2000.

[41]. Engineering Design Of Systems, Dennis M. Buede, Wiley, 2000

[42]. Systems Engineering Fundamentals, Defense Systems Management College(DSMC), Ft Belvoir, Jan 2001

component. SysML™, derived from UML and also discussed in this chapter, fills in the gaps to offer a meaningful tool for modeling and simulation of complex space systems.

As shown in the hierarchy diagram in Figure 5-19, UML specification[43] defines two major kinds of UML diagrams: structure diagrams and behavior diagrams. Interaction diagrams are derived from the behavior diagrams.

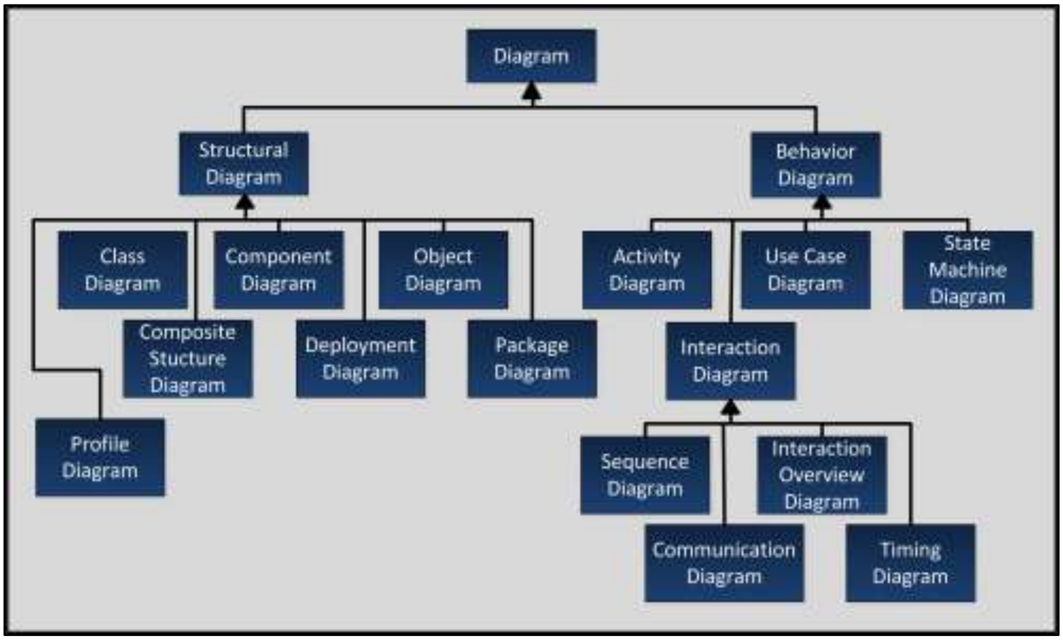


Figure 5-19 UML hierarchy diagram

5.24.1 Structure diagrams

Structure Diagrams include the Class Diagram, Object Diagram, Component Diagram, Composite Structure Diagram, Package Diagram, and Deployment Diagram. Structure diagrams show static structure of the system and its parts on different abstraction and implementation levels and how those parts are related to each other. The elements in a structure diagram represent the meaningful concepts of a system, and may include abstract, real world and implementation concepts. Structure diagrams do not represent dynamic behavior or time related activities of a system. However, they may show relationships to the behaviors of the classifiers exhibited in the structure diagrams.

- **Class diagram –** is a static structure diagram which describes structure of a system at the level of classifiers (classes, interfaces, etc.). It shows some classifiers of the system, subsystem or component, different relationships between classifiers, their attributes and operations, constraints.
- **Object diagram –** instantiates objects and data values. A static object diagram is an instance of a class diagram that shows a snapshot of the detailed state of a system at a point in time.
- **Component diagram –** shows components and dependencies between them. This type of diagrams is used for Component-Based Development (CBD) or to describe systems with Service-Oriented Architecture (SOA).

---

[43] For current UML specification and relevant information visit OMG's website

- **Composite structure diagram –** could be used to show an internal structure of a classifier or an instantiation of behavior of a collaboration. Model diagram is an auxiliary structure diagram which shows some abstraction or specific view of a system, to describe architectural, logical or behavioral aspects of the system. It could show, for example, architecture of a multi-layered application. Internal Structure diagrams show internal structure of a classifier - a decomposition of the classifier into its properties, parts and relationships.
- **Package diagram –** shows packages and relationships between the packages.
- **Deployment diagram –** shows architecture of the system as deployment (distribution) of software artifacts to deployment targets. Specification level deployment diagram (also called type level) shows some overview of deployment of artifacts to deployment targets, without referencing specific instances of artifacts or nodes.
- **Profile Diagram –** provides a generic extension mechanism for customizing UML models for particular domains and platforms. It allows definition of custom stereotypes, tag definitions, and constraints that are applied to specific model elements, such as Classes, Attributes, Operations, and Activities. A Profile is a collection of such extensions that collectively customize UML for a particular domain like aerospace, or platform like J2EE.

5.24.2 Behavior Diagrams

Behavior Diagrams include the Use Case Diagram (used by some methodologies during requirements gathering); Activity Diagram, and State Machine Diagram. Behavior diagrams show the dynamic behavior of the objects in a system, which can be described as a series of changes to the system over time.

- **Use case diagrams –** are behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors) to provide some observable and valuable results to the actors or other stakeholders of the system(s). Use Case Diagram is used in SysML™ without modifications. SysML™ also has the capability to represent test cases and to attach them to their related requirements or use cases. A test case can be an operation or a behavioral model (Interaction, State Machine, or Activity).
- **Activity diagram –** shows sequence and conditions for coordinating lower-level behaviors, rather than which classifiers own those behaviors. These are commonly called control flow and object flow models.
- **State machine diagram –** is used for modeling discrete behavior through finite state transitions. In addition to expressing the behavior of a part of the system, state machines can also be used to express the usage protocol of part of a system. These two kinds of state machines are referred to as behavioral state machines and protocol state machines. The SysML™ State Machine Diagram is used to represent the different states of the product; however, Protocol State Machines are excluded from SysML™ for simplicity.

5.24.3 Interaction diagrams

Interaction Diagrams, all derived from the more general Behavior Diagram, include the Sequence Diagram, Communication Diagram, Timing Diagram, and Interaction Overview Diagram. SysML™ Interaction Diagram: This diagram allows the designer to model a sequence of service calls between components. SysML™ leverages the UML 2.0 interaction model, but restricts its use to the Interaction Diagram only. Other forms of Interaction Diagrams (e.g., Communication Diagrams) are not used in SysML™.

- **Sequence diagram –** is the most common kind of interaction diagrams, which focuses on the message interchange between lifelines (objects).
- **Communication diagram –** (previously known as Collaboration Diagram) is a kind of interaction diagram, which focuses on the interaction between lifelines where the architecture of the internal

structure and how this corresponds with the message passing is central. The sequencing of messages is given through a sequence numbering scheme.

- **Interaction overview diagram –** defines interactions through a variant of activity diagrams in a way that promotes overview of the control flow. Interaction overview diagrams focus on the overview of the flow of control where the nodes are interactions or interaction uses. The lifelines and the messages do not appear at this overview level.
- **Timing diagrams –** are used to show interactions when a primary purpose of the diagram is to reason about time. Timing diagrams focus on conditions changing within and among Lifelines along a linear time axis.

5.25 Value Stream Analysis and Mapping

MIT's Lean Aerospace Initiative (LAI) contends (see Figure 4-1 in Section 4.2.2) that 40% effort in aerospace projects is wasted. Value stream mapping is a an important lean tool that can help identify waste, reduce process cycle times, and implement process improvement. The value stream is a collection of activities necessary to produce and deliver a product or service efficiently with little waste. Value stream analysis separates those activities that contribute to value creation from activities that create waste, and identifies opportunities for improvement. Some of the metrics that can be used to measure output value are:

- Takt or cycle time: time needed to complete a process or activity
- Recurring costs: resources required to complete a job or activity
- Non-recurring costs: Fixed resources, facilities, or tools required for an activity
- Lead time: Latency or time required to fulfill a need after it is identified
- Variation: quantitative measures of variation in the execution of jobs or activities
- Rework: incidence of defects or cost of rework
- Customer satisfaction: surveys or checklists to quantify customer satisfaction

Quantitative measurement of value is hard, and as such, metrics need to be chosen carefully based on available data. EVM (also discussed in this chapter) or process tracking software can be employed for value stream analysis.

Value stream maps can be hand drawn as pencil and paper activity. While value stream maps are not overly difficult to construct, utilizing software can help speed up the process and simplify calculations that help make up a completed map. Further enhancements often include tools that can analyze the current process and facilitate future state maps with the provision of "what if" and scenario modeling. Such tools also make it easier to engage the extended team (management, suppliers, and finance) through more accurate, clearer presentation of the current state. In any case, the most valuable part of a VSM activity is the maps themselves and whilst software can be used for documenting the findings it shouldn't detract from the process.

The essential product development steps in VSA&M are:

- Map the current or as-is product development process: Follow the product from start to finish to develop product task and flow map, collect available data, and identify possible metrics to quantify value.
- Identify waste in the process: As-is VSM is used at this stage to identify wastefulness in the process that include unnecessary waiting for people or information, superfluous inventory or information, over-processing, over-production, product defects, unnecessary movement, and unwarranted complexity and incompatibility.

- Improve the process: Study takt time to balance the activities, streamline information availability, eliminate unnecessary analyses, break-up stereotypes ("we have always done it this way"), eliminate unnecessary documents and reworking or re-formatting of documents, eliminate unnecessary reviews and approvals. Use these new insights to build an improved and less wasteful "to-be" or future state value stream map.

5.26 Value System Design

Value System Design is a technique for establishing the system requirements in a fashion that can be easily understood and measured by all who contribute to the design. It essentially takes the requirements in the user's language and translates into goals in the designer's language. Value system design looks at several areas that define what is desired of the product to include:

Objectives – Objectives include requirements but may also include goals above requirements or in areas not specifically stated in requirements. For example, you may want a faster processor because it's needed on a collateral project, or you may want to develop the capability to advance a product out of the lab and into production. Setting objectives has strong elements of the creative dimension. Objectives must be stated in terms of what is needed, not how to implement them. Presupposing solutions eliminates initiative and innovation. Objectives are often stated as maximization, minimization, or closet fit to a target. The English language with its ambiguities and slanted meanings can be a hindrance. It is important that each objective is simply stated and is measurable. Also objectives must be consistent with user requirements and lower-level objectives must be consistent with higher-level ones. Otherwise, efforts are wasted on objectives of no import. Establishing the right objectives is crucial for product success. Wrong objectives lead to wrong solutions. Using the right objectives, you have a better chance of selecting the right solution even if it is less than optimal.

Objectives Measures – Objectives measures are sometimes called Measures of Effectiveness (MOEs). A product's effectiveness determines its "worth." SE seeks the greatest possible "worth" at an acceptable cost. Measures of effectiveness characteristics include (i) Relates to performance, (ii) Simple to state, (iii) Complete, (iv) States any time dependency, (v) States any environmental conditions, (vi) Can be measured quantitatively (if required, may be measured statistically or as a probability), and (vii) Easy to measure. An example of an MOE for an automobile is fuel consumption in miles per gallon under specified environmental conditions.

Effectiveness at a system level may have several definitions. A typical definition comprises of (i) performance: the probability that the product will perform its mission, (ii) dependability: the probability that a product is available and reliable in use, and (iii) utilization: the actual use of the product versus its potential.

Measures of effectiveness have many factors. To help you identify critical contributing factors you may wish to show them graphically as a performance hierarchy tree traceable from the original user requirements, through the system objectives, to the subsystem and lower-level objectives. Be sure the measures of effectiveness have quantitative expressions. Analyze the measures of effectiveness to develop supporting measures of performance. Make the measures of performance specific, and derive lower-level measures from these. The complete hierarchical structure thus formed shows the critical technical performance measures.

Criteria and weighting – Criteria differ from constraints. Constraints are the "musts," the restrictions, the limitations that have to be met and are generally not available for trade-offs. Constraints can be used for screening to filter out alternatives, however, once screening is accomplished, constraints can no longer help determine the best alternative. Constraints establish boundary conditions within which the developer must remain while allocating performance requirements and/or synthesizing system elements and are generally pass or fail. Criteria provide a means of judging feasible alternatives. Examples might be lowest cost, most range, fastest acceleration, or closest flow rate to 10 gallons per minute. Sometimes, a measure can be both a constraint and a criterion. For example, as a constraint, the product must cost no more than $10,000, but the customer prefers the lowest cost below that point. A cost of $10,000 is the constraint; costs below $10,000 are criterion. Sources of criteria include (i) the customer, (ii) quality (iii) functions or behaviors, (iv) measures of effectiveness, (v) measures of performance, (vi) contractual costs, (vii) contractual schedules, (viii) manufacturing, (ix) product support, and (x) project and organization objectives.

When criteria are not of equal importance, weighting factors are assigned as a means of identifying relative importance. In evaluating alternatives, criteria weighting seeks a closer problem-to-solution match. Weighting can be established empirically or subjectively. The empirical method derives weights by determining how much each elementary measure contributes to a general outcome. Large numbers of measures require statistical analysis. The scenarios and environments for the studies must be chosen carefully. The sensitivity of measures of success or stated customer desires to changes in individual criteria drives the weighting of those criteria.

Subjective weighting relies on the judgment of experts. One widely used method gives raters a fixed number of points, 100 or 1000, to allocate to the criteria. The distribution of points reveals each criterion's relative importance. In another technique, experts score existing alternatives and then the criteria and weighting factors are derived by analyzing the preferred alternatives. This latter method is used more for establishing values for subsequent design efforts rather than selection candidate approaches.

The empirical techniques are sensitive to the specific conditions for which they were measured. The subjective techniques depend on the judgment of the experts. New products might not have strongly identified criteria. Scoring should always be challenged, and recursion often occurs as the program matures.

Table 5-4 is an example of a scoring chart using weighting. Cost, performance and reliability are the major factors, accounting for 80% of the total weighting. Scores in the range zero to five are assigned by criterion to each alternate and then multiplied by the weight. After the weighted scores are summed, Alternate 3 is the clear winner. Early in a program, Alternate 2 may also be carried along as insurance in case the criteria or their weighting change, e.g., Alternate 3 does not live up to expectations, or Alternate 3 depends heavily on unproven or immature technology.

Table 5-4 Criteria weighting–an example of comparison using weighted criteria

		Alternative					
		1		2		3	
Criteria	**Weight**	**Score**	**Weighted Score**	**Score**	**Weighted Score**	**Score**	**Weighted Score**
Cost	40	3	120	4	160	5	200
Performance	30	3	90	4	120	5	150
Reliability	10	2	20	3	30	3	30
Maintainability	5	1	5	4	20	3	15
Ease of Mfg	5	2	10	3	15	4	20
Ease of Use	5	5	25	4	20	4	20
Safety	3	4	12	5	15	5	15
Ease of Test	2	3	6	3	6	2	4
Total	100		288		386		454

As with any SE technique or tool, it is necessary to understand the underlying principles that contribute to Value System Design results. In the example in Table 5-4, it is prudent to analyze the sensitivity of each of the Alternates 2 and 3 to changes in requirement values. It may be that a small but acceptable change could radically change the outcome.

Utility – Utility curves are one means of checking sensitivity. Utility curves describe the relative value of a criterion for different levels of performance. They are graphs of a characteristic versus its relative numeric value. This method establishes the relative value of the factor as it increases from the minimum value of the range. The curve may show a constant value relationship (straight line), increasing value (concave curve), decreasing value (convex curve), or a stepped value. Figure 5-20 shows a set of typical utility curves. The examples shows utility ranges from 0-5. Calculating loss is one way to plot a utility. In Figure 5-20 the schedule is insensitive to time for the first six months, but missing that schedule results in a total loss. For mean time between failures (MTBF), loss decreases nearly linearly as the MTBF increases out to about 10,000 hours. Conversely, loss is fairly insensitive for mean times to repair (MTTR) less than 20 minutes, but drops sharply after that point. Battery life shows little loss of utility for all plotted values. Estimating the loss at intervals resulted in points that can be graphed. Such graphs show sensitivities in easily understandable form. A final word of caution: do not use Value System Design in isolation as the sole basis for selection. The application of another tool/technique might provide insight missed by blindly accepting the results shown. Also results should be evaluated in light of experience.

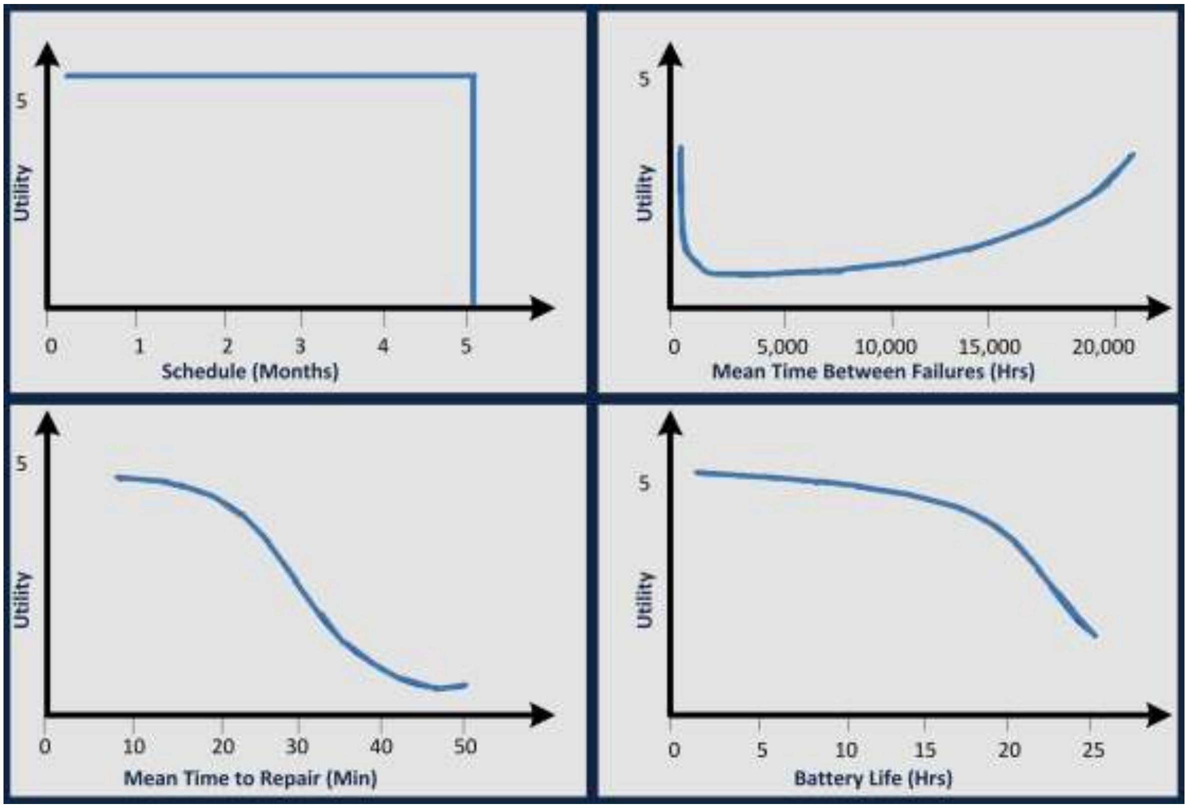


Figure 5-20 Utility curves–providing insight into criteria sensitivity

5.27 WBS and Physical Architecture Tools

A WBS is a product-oriented hierarchical tree composed of the hardware, software, services (including cross-product tasks such as SE), data, and facilities that encompass all work to be carried out under the program or contract along with a dictionary of the entries in the tree. The WBS for the entire program is called the Program or Project WBS (PWBS). The WBS for the work under the contract is called the Contract WBS (CWBS) and is prepared in accordance with the contract.

The WBS is a means of organizing system development activities based on system and product decompositions. It is a product-oriented family tree composed of hardware, software, services, data, and facilities, which result from SE efforts during the development and production of the system and its components, and which completely defines the program. The WBS is prepared from both the physical and system architectures, and identifies all necessary products and services needed for the system. This top-down structure provides a continuity of flow down for all tasks. Enough levels must be provided to properly define work packages for cost and schedule control purposes.

Because the WBS is a derivative of the physical and systems architectures, it is a direct output of the SE process. It can also be considered part of the synthesis process since it helps to define the overall system architecture. The DSMC Systems Engineering Fundamentals Book, December 2000, includes the WBS in the System Analysis and Control process as a tool to help represent and control the overall process. The WBS is not just about hardware or software but also is used to structure development activities, identify data and documents, organize integrated teams, and is used for non-technical program management purposes such as scheduling, and measurement of progress.

A program WBS is established to provide the framework for program and technical planning, cost estimating, resource allocation, performance measurement, and status reporting. The WBS defines the total system of hardware, software, services, data, and facilities, and relates these elements to each other and to the end product. The WBS is critical to the development of SEP which is required before milestone decisions for all ACAT programs. The WBS is also an integral part of preparation of the Cost Analysis Requirements Description (CARD). Furthermore, a well-developed WBS is essential to DoDI 5000.02 requirements for IMS and EVM.

A sample WBS of a launch system is shown in Figure 5-21. Program Offices usually have the responsibility to develop an overall program WBS and to initiate development of contract WBSs for each contract in accordance with common DoD practice. The program WBS is the WBS that represents the total system and, therefore, describes the system architecture. The contract WBSs are part of the program WBS and relate to deliverables and tasks on a specific contract. The Program Office with the support of SE develops the first three levels of the program WBS, and to provide contractors with guidance for lower-level WBS development. Though WBS development is a SE activity, it impacts costing, scheduling and budgeting professionals, as well as contracting officers. An integrated team representing these stakeholders is needed to support WBS development.

The first three Work Breakdown Structure Levels are organized as:

- **Level 1 –** Overall System
- **Level 2 –** Major Element (Segment)
- **Level 3 –** Subordinate Components (Prime Items)

Levels below the first three represent component decomposition down to the configuration item level. In general, the government is responsible for the development of the first three levels, and the contractor(s) for levels below three.

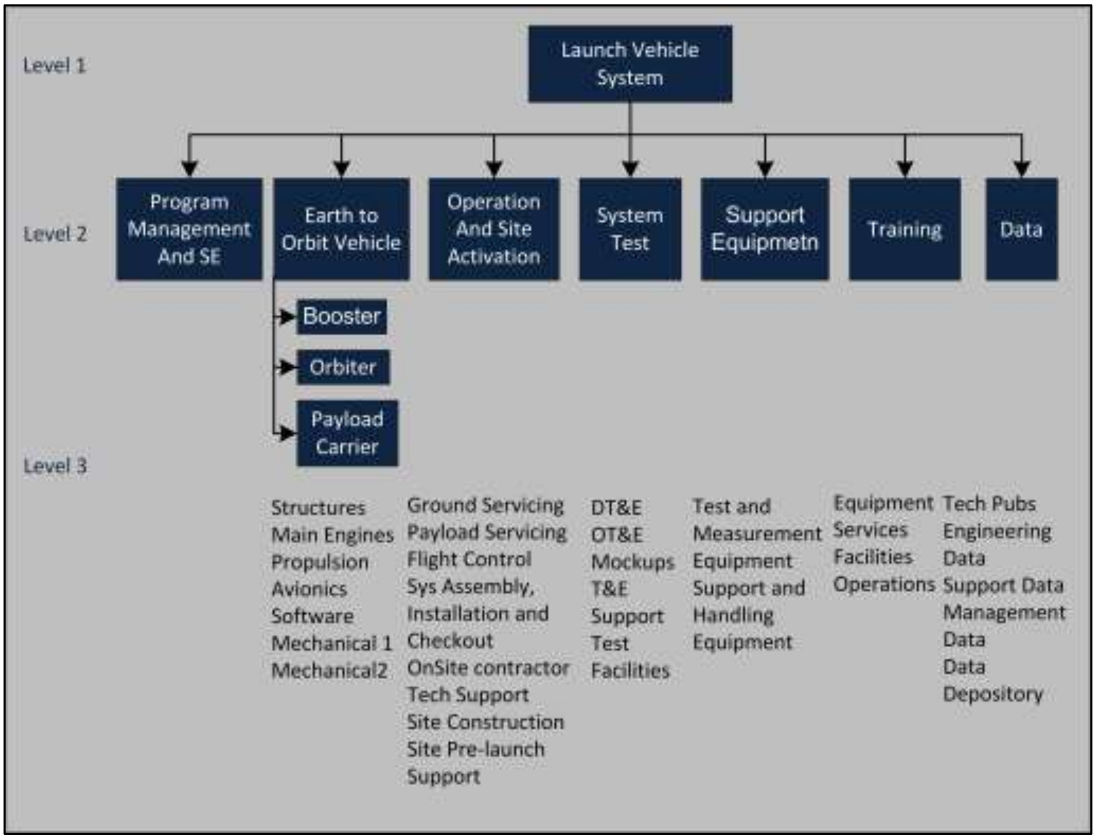


Figure 5-21 A sample WBS of a launch system

Program offices develop a program WBS tailoring the guidance provided in MIL-STD-881C. The standard offers a comprehensive framework for the PM to help "achieve a consistent application of the WBS for all programmatic needs (including performance, cost, schedule, risk, budget, and contractual)." Appendices C and F of the standard provide very detailed WBSs and definitions for missile systems and space systems, respectively.

6 Specialty Engineering Disciplines

A number of specialty engineering disciplines, supporting a system throughout its lifecycle, are indispensable and integral to the SE process. This chapter briefly introduces the specialty engineering disciplines (SED) that are crucial to the success of SE for complex systems like those in the SMC portfolio of space related programs. Based on system and program requirements, the SPOs implement approved and documented processes to perform specialty engineering and support SMC's goals for OSS&E and Mission Assurance. SMC has developed a companion volume to this book that provides a standard approach and framework to SED.

The SED framework includes the essential activities, tasks, and products that shape the body of knowledge for each SED. The SMC Standard SED framework is designed to characteristically capture the following attributes:

- Provide SED contributions to all major SMC acquisition activities while accounting for acquisition phase dependencies.
- Integrate with the overarching SE activities and adjunct Specialty Engineering activities.
- Integrate with the program and project management activities.
- Comply with technical mandates, regulations, and objectives.
- Provide a high degree of usability to leverage for SPO's detailed Specialty Engineering planning, process development and SED's execution.
- Provide a low risk and cost effective path toward mission success.

SED's analytical solutions, contributions, and controls must be highly leveraged through the system analysis and control activity. It is essential to recognize that a SE organization in a SPO must plan and execute essential engineering and management efforts within the context and in full support of the overarching SE function. It is the responsibility of each SE organization to ensure that their contributions are timely, adequate, consistent, documented, and approved, and comply with the relevant public law, DoD and other mandates, and SPO requirements. Furthermore, SE organizations must coordinate and effectively communicate with and support other SEDs that are closely related to their discipline within and, often, external to the SPO to get an appreciation of the enterprise or SoS perspective.

For details on how a SED framework is implemented refer to SMC's "Systems Engineering Handbook, Volume II: Specialty Engineering Disciplines," 2011. The following subsections briefly describe the overarching SE and the SED's technical domains and their importance to complex space systems and programs.

6.1 Overarching Systems Engineering

SE is also considered a SED since this discipline includes specific engineering functions that include requirements analyses, interface analyses, functional analyses, technical solutions trades, systems studies, and system element allocations, as well as the integration and verification and validation planning and execution.

The responsibility to orchestrate the engineering functions and manage technical information typically resides within the SE organization. In performing the management and control function, the SE effectively integrates all engineering functions through the full system life cycle. The SE ensures technical information advances through systematic control, collaboration and sharing across the organization. The approach to concurrently perform these engineering functions and manage information is the subject of this book.

The SPO SE organization has the responsibility to perform the overarching engineering management and control functions. To summarize SE, as a SED,

- provides engineering activities over a system or product life cycle
- aligns activities to an evolving technical baseline
- aligns both the activities and technical baseline with required technical review gates
- provides an overarching engineering management and control function as defined within the Analysis & Control process

6.2 Test and Evaluation Engineering

Test and Evaluation (T&E) is an ever-present activity that is essential to successful execution of a program over the lifecycle of a system. Consistent with customer needs, T&E organization produces the system lifecycle T&E strategy, obtains the necessary resources to conduct the testing, and works with other organizations within the overarching SE to ensure that testability is factored into the design. It helps verify that the system design meets requirements and that the design characteristics and reliability is adequate for production, deployment, and operation. Testing is performed to assess maturity of new technology, characterize design, integration, verification, reliability, qualification, LRIP, FRP, depot, pre-launch, DT&E, and OT&E. For large and complex systems at SMC, T&E organization strategizes, plans, coordinates, and manages testing and evaluation over the entire lifecycle of the program. Some of the testing activities include:

- Requirements analysis, decomposition, and allocation for test
- Design for test
- Test to validate models
- Design of tests to mimic intended use environment (“test as you fly”)
- Strategies to minimize testing need in production and field upgrades

Throughout the program life-cycle, T&E Engineering ensures evolving technologies and design solutions meet system and design specifications and that the final design meets operational requirements and constraints. T&E Engineering contributes to this process by supporting concept and architecture development and analyses; modeling and simulation efforts; and technology development; design trades, V&V, and sustainability analyses. T&E Engineering supports the requirements analyses and allocation process, to assure that the requirements are well stated, are verifiable, and that associated verification methods are unambiguous. Further, T&E Engineering develops and derives verification requirements that include test procedures, analyses, demonstrations, and inspection. The scope of T&E Engineering activities with SE includes evaluation of verification and validation of interfaces, functions, and integration and test at the component, segment and system levels. T&E Engineering ensures specification test strategies, plans, and methodologies developed by SE are adequate.

The T&E Engineering organization for the SPO manages and integrates T&E activities, resources, and information within statutory and regulatory guidelines using sound engineering principles. T&E Engineering ensures the baseline test and analysis requirements are appropriately established, resources are available, and the events are incorporated in the contractors Integrated Master Schedule (IMS) and government master schedules. T&E Engineering strives for the identification and elimination of inherent or latent defects, process (e.g. workmanship), and procedural deficiencies to ensure a high level of confidence in the progression of the system development efforts to meet the intended reliability growth targets, performance requirements, and Total Ownership Cost (TOC) targets. T&E Engineering collaborates with the Program Office SE organization and the acquisition community, and provides the evaluation results including deficiency and risk discovery and remedies.

With enterprise-wide connectivity, net-centric operations, and ubiquitous sharing of information T&E organization must also involve itself in testing for information assurance in cyberspace.

6.3 Software Engineering

Software engineering focuses on all aspects of software over the lifecycle of the system to include software development, design, production, verification and validation, operation and maintenance. It is the application of a systemic, quantifiable and disciplined approach to analyzing, designing, assessing, implementing, testing, maintaining and developing software. Software is a well-defined and established SMC SED, and is an integral element of all space and missile systems today.

With increasing emphasis on SoS-level interoperability, reusability, open vendor-neutral architecture, net-centric operations, and the need to provide information assurance to maintain warfighter advantage, the SW Engineering organization's role has become crucial to the success of the program and the enterprise.

Software developed for space applications needs the same rigorous engineering principles as are applied to hardware that goes into space. One of the methods for ensuring this rigor is to conduct Software Development Capability Evaluations for all parties that will be producing software. These evaluations ensure the developers have proper, rigorous and well documented processes for:

- Managing and Allocating Requirements
- Proper Documentation Procedures
- Software Design
- Error Handling Procedures
- Design Reviews
- Configuration/Change Management
- Test Procedures and Testing
- Integration
- Defect Management
- Maintainability
- Delivery

Since SW errors can and have resulted in catastrophic mission failure, it is imperative to follow a rigorous development process. Common certifications to ensure rigor in software development include CMMI® (Capability Maturity Model Integration) and ISO 9000 certification. These certifications show the developer has processes in place to properly control and document the software development process; thereby, helping to safeguard against unwarranted defects in delivered software. The SPO's SW organization ensures that a rigorous well-documented process is followed in all activities related to SW development, debugging, production, and sustainment.

SMC has developed an extensive body of knowledge to guide acquisition of software which has increasingly become a major factor in the success of its programs. This guidance, as stated in Chapter 1, is documented in (i) SMCI 63-103, Software Acquisition Process Improvement Instruction, (ii) SMCI 63-104, Software Acquisition Instruction, (iii) SMCI 63-108, Software Acquisition Management Plan (SWAMP), (iv) SMC-S-012, Software Development for Space Systems, (v) SMC-S-21, Technical Reviews and Audits for Systems, Equipment and Computer Software, Appendix C.

6.4 Integrated Logistics Support

DoDD 5000.01 requires Program Managers to: "develop and implement performance-based logistics strategies that optimize total system availability while minimizing cost and logistics footprint." Further, within the Defense Acquisition System, DoDD 5000.01 requires that: "Planning for Operation and Support and the estimation of total ownership costs shall begin as early as possible. Supportability, a key component of performance, shall be considered throughout the system life cycle."

The Program Manager (PM), as the life-cycle manager, is responsible for accomplishing program objectives across the life cycle, including the Operations & Support (O&S) phase. Employing performance-based life-cycle product support tied to sustainment metrics is the overarching DoD concept for providing materiel readiness to the user. The PM, Product Support Manager (PSM), and Life-Cycle Logistician can influence the design and provide effective, timely product support capability to achieve the system's materiel readiness and sustain operational capability. This can be effected by placing the emphasis on integrating life-cycle management principles, using performance-based life-cycle product support strategies, combining SE processes resulting in materiel readiness at optimal life-cycle cost (LCC) through reduction of frequency, duration, and related costs of availability degrading events, reducing the system’s manpower and logistics footprint.

The practice of Integrated Logistics Support (ILS) ensures that proper skills, equipment and resources are available for use when needed. ILS Operations are essentially a contingency action. That is to say, if things were perfect at the time of space system delivery, the deployed operational space system would be self-sustaining and would not require a support infrastructure to bring it on line and maintain optimal operating conditions.

ILS is a multifaceted activity starting with early SE and running through the complete system life cycle of operational deployment to ultimate decommissioning (including disposal, where appropriate). ILS can be broken into four distinct areas: Logistics System Engineering; Logistics Products Acquisition (including T&E); hands-on Logistics Operations; and ILS Program Management. A well-structured ILS activity seamlessly integrates these to provide an effective and economical space system sustainment from cradle to grave. By its nature, ILS is an iterative process wherein actions performed upon the start of a program (i.e. during the Materiel Solution Analysis Phase) are continued and refined throughout the entire system life cycle.

Logistics and Support or ILS contains ten elements that are mini-disciplines in their own right. These elements are:

- **Maintenance planning –** the determination of what maintenance operations are required and the organizational level at which they will be performed.
- **Manpower and personnel –** the numbers of personnel and kinds of training required at each level to support the maintenance planning.
- **Supply support –** provisioning and the development of data to support provisioning.
- **Support equipment –** planning, design and development of equipment to test, handle and service the system in the field.

- **Technical data –** planning and development of manuals, drawings, and related documents required to operate and maintain the system equipment at all planned maintenance levels.
- **Training and training support –** planning development and execution of training required to implement the maintenance planning and of all the devices, mock-ups, and documentation necessary to conduct training.
- **Computer resource support –** planning and support of efforts required to maintain and upgrade fielded system software/hardware.
- **Facilities –** plan and implement the modification or upgrade of existing facilities, or the development of new facilities to support the system.
- **Packaging, handling, storage & transportation –** planning the modification or upgrade of existing containers, equipment, or facilities, or the development of new ones to enclose, handle, warehouse or move complete systems or their components.
- **Design interface –** the sum of all efforts to ensure transfer of the latest design information to those performing ILS analyses and related work, and to ensure that the results of ILS operations properly influence system design. Often these efforts result in establishment of a central database of design and support data that can be accessed electronically by all those involved in the development and use of the data

The cost of ILS over the system life cycle is an important concern. This problem of affordability is compounded by the fact that only an adequate up-front investment provides assurance of a “just right” logistics support posture at and during system deployment. This distinction between budgetary requirements vs. down-stream forecasts must be met, and resolved, through a comprehensive LCC analysis. Failing to properly perform the ILS function can lead to schedule delays and cost overruns.

ILS involvement in early design decisions greatly reduces support costs and facilitates some of the recent reliance on Commercial Off-the Shelf (COTS) items, Non-Development Items (NDI), and joint usage. ILS personnel are involved from the earliest requirements analyses through development, production, deployment and continuing operations. A properly constructed ILS program offers better system reliability, availability, maintainability, testability, and safety.

6.5 Design Engineering

Design Engineering function for a SPO is incorporated in all design engineering activities throughout the acquisition life cycle process of a system to achieve higher innovation through technology insertion, balanced solutions, reduced system development risks, apply reuse strategies, and reduce cycle time and system costs. For each of the life cycle phases, the SMC design engineer participates to meet the objectives in planning, acquisition, and engineering activities for effective contract execution. Typical design engineering organization guidelines include:

- Minimize number of parts to reduce cost of fabrication and assembly
- Foolproof and unambiguous assembly process
- Standardize handling and assembly operations
- Design testability into the product and consider built-in tests
- Avoid tight tolerances for robust manufacturing, testing, and operation
- Design for convenient and intuitive user interface
- Design for ease of upgrades and service
- Utilize common parts and materials for easy availability and lower cost

The design engineering discipline provides the means by which space system requirements are converted into a detailed design, along with the documentation necessary to manufacture and test hardware. The discipline encompasses all the engineering functions necessary to design a hardware product and verify its performance prior to product manufacturing. Key design engineering functions for space include:

- Electrical power generation, storage, conditioning, and distribution
- Command and data handling, which typically integrates communications between ground and space with spacecraft commanding, and telemetry and payload data downloading
- Spacecraft attitude/altitude determination and control
- Spacecraft thermal and EMI/EMC control
- Vehicle structure and electromechanical devices such as gimbals, separation rings and deployment mechanisms
- Propulsion subsystem for orbit insertion

Design engineering converts SE concepts into a series of products that are then assembled into an integrated spacecraft. These products are either complete subsystems, such as electrical power or vehicle structure, or they are individual parts, assemblies, or components, such as reaction wheels, solar arrays, or transponders. Design engineering organization interprets requirements to satisfy mission performance within externally imposed vehicle weight, volume, and electrical power budgets for hardware. When system life cycle cost is considered, it is desirable to minimize the cost of the launch system.

The engineering skills that generally make up the design engineering process at the subsystem level include analysis, requirements synthesis and technical insight. Engineering skills at the equipment level include analysis, design layout, drawing, bread-boarding and brass-boarding, and performance testing.

In addition to the above, engineering management is necessary to control the overall design process. Designing for space applications requires unique approaches that are very different from those of other applications. The design engineering discipline must consider operational environments that are nothing like that of terrestrial environments. High radiation and microgravity environments, the stress and vibrations of launch, are only a few examples of the uniqueness of space applications design considerations.

6.6 Manufacturing and Producibility

Manufacturing is a conversion process, transforming raw material into a finished product. The process consists of bringing together resources; materials, manpower, tooling, equipment, technology and facilities in a structured, organized manner to produce a system design that meets expected capability needs. DoDI 5000.02 requires that manufacturing feasibility, processes, and risk be assessed on all acquisition programs beginning during the MSA Phase and continuing until the Full-Rate Production Decision Review.

Many discrete sub-processes are involved, leading to the most effective, economical means of fabrication, assembly, installation and checkout, inspection and test, and final acceptance of the space system end product. These sub-processes can be placed into logical groupings relating to the management and organization of the function, the planning and design for producing the product, and daily activities involved in the execution of the plan. These groupings are for the purpose of evaluating the process and are not identified with rigid organizational alignments that may vary considerably among contractors.

Manufacturing, like other disciplines, is an integral part of the product development process. In order to assure a consistently repeatable, high quality product, the detail design must be translated into a series of operations that add value to the product and transform the raw materials into a usable end item. The processes and discipline necessary to fabricate, assemble, integrate and test a product are a critical part of the product development cycle and necessary for delivering a cost-effective product to meet the customer's requirements in a timely manner.

The manufacturing function interfaces with SE, design engineering, quality, configuration control and procurement throughout the product development process. The SE process involves the manufacturing function early in the design phase when producibility is paramount in conducting trade studies of the preliminary design concepts. As system functions are defined in the design evolution, the detailed designs are further evaluated with design engineering to identify manufacturing alternatives that will enhance the producibility of the design and insure consistent repeatability in the hardware manufacturing process. Material and process specifications as well as delivery schedules are provided to the procurement function to control purchased items and materials. Accurate and complete production records are maintained to document the as-built configuration.

Methods and processes that enhance the producibility of a product may also have beneficial impact on testing, reliability, and support. On the other hand, certain means of functional division, interconnection, or assembly may improve producibility but adversely affect testability or reliability, or add to the problems of maintenance, servicing, provisioning, or operation. Achieving a balanced space system design requires that the other disciplines in the Integrated Product Team (IPT) be recognized as important contributors to the finalization of manufacturing decisions.

6.7 Quality Assurance

Quality Assurance is a program-wide activity that impacts the working of SE and other SEDs to help produce the right system over its entire lifecycle. The QA organization ensures that all contractually imposed specifications, standards, processes, and other design requirements are met. It also assists in establishing internal conformance levels and criterion of acceptability for processes and products.

The QA organization for the SPO implements necessary process control, statistical sampling techniques, and assures that design specifications are within the manufacturing process capabilities to meet customer expectation on quality. The search for quality is a process of continuous improvement. The most common QA processes applied today in the industry for QA are codified in Six Sigma and Lean.

Six Sigma strategy is to identify and root out all sources of variation, looking to achieve six-sigma standard deviation (3.4 ppm defects). It is a data driven methodology that is characterized by:

- Customer requirement definition that drive process improvement
- Identify or define metrics to measure variation
- Analyze measured data to discover problems in process
- Develop and implement solutions to problems to minimize variation
- Codify, control, and sustain improvements

Lean is a people-oriented quality management approach. It attempts to continuously enhance the value of a product, service, or process by eliminating waste. Figure 6-1shows the five step process based on principles of lean. Value is defined by the customer requirements and expectations. Value stream mapping decomposes all activities required to transform inputs to outputs in an attempt to identify and eliminate waste. As wasteful activities are eliminated, the value creation process is optimized. Establishing pull refers to the customer's ability to see and effect changes if necessary in the value creation process over the entire supply chain at any time. Lean recognizes that there is always room for improvement. Establishing a visible and interactive process for value creation allows for all stakeholders to participate in waste elimination in pursuit of perfection.

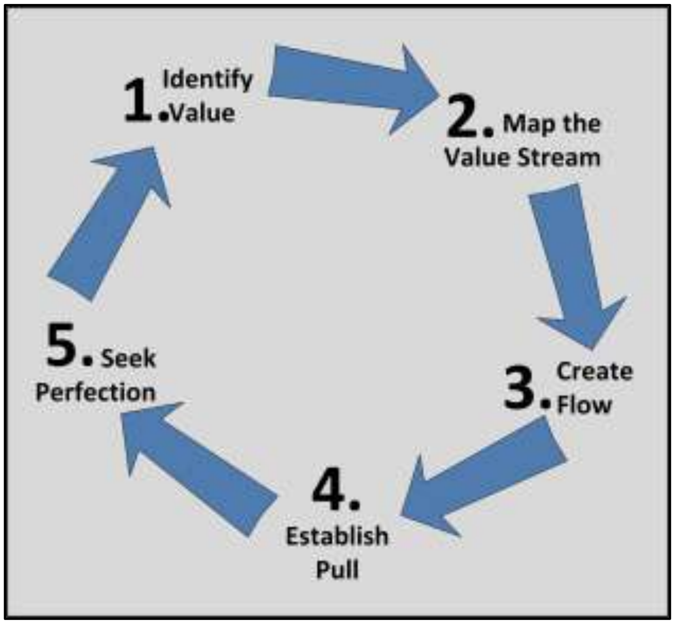


Figure 6-1 Lean process flow

QA manages quality processes such as the internal engineering change release system, calibration system, inspection systems and other quality related processes. It acts as an in-process and final check of workmanship, test and overall production functions. Furthermore, QA organization employs statistical quality control and related tools and technologies to assess process variability and production defects. The QA organization provides technical advisement on product and process improvements, supports root cause of anomalies, non-conformances, or test failure investigations, and makes recommendations on dispositions of non-conformances.

The QA organization understands legal and contractual ramifications of design and planning options and is therefore valuable as a counsel to steer away from future problems inherent in proposed implementation approaches. Quality is also helpful in establishing test programs to verify that the design meets user requirements. Once design-proofing tests are complete, QA oversees planning and design in the production system. Quality may also identify slight changes in the design of individual candidate approaches that could make the process cost-effective.

As software has attained a greater importance in modern systems, so too has the need for Software Quality Engineering and Control. Project Engineers must be mindful of this requirement and assure that Software Quality is involved appropriately in the program.

Because they can be an important aid in avoiding pitfalls and future problems, Quality must be involved in the SE process during the space acquisition program from its inception through final disposal. Project Engineers should promote a good working relationship with Quality personnel and listen well to their suggestions. If Quality concerns cause problems, it is not due just to the intractability of Quality, but more often a need to reevaluate some assumptions and requirements. It may even be a sign that the Project Engineer should confer with the user/customer to ascertain that they are willing to assume the costs involved in reaching certain goals.

While emphasis is placed on the engineering and test functions to provide the leadership necessary to satisfy product performance, configuration, and environmental requirements, the integrated efforts of the entire program are necessary to provide a product that meets these requirements. Within this broader perspective, quality becomes an attribute distributed throughout all program functions. The QA function is the set of assurance processes that verify and validate that the product development process is complete, in compliance with requirements, and meets customer expectations. These assurance processes embrace nearly the complete product acquisition life cycle. In the design phase of a program, QA planning is predominant. It involves quality in design, quality of design and, quality of conformance as the product matures through development and production.

Product Quality is the result of providing a control structure in which to create process discipline necessary to realize high reliability systems. Discipline is the building material necessary to ensure consistently predictable, repeatable, and successful processes. Product or process quality, in the broadest sense, is achieved through planning and control over the range of functions and processes needed to develop, manufacture and operate a space system. QA as a function is made up of quality management, quality engineering and inspection processes that play critical roles in executing this overall control structure.

The processes that generally make up the QA function include quality management, procurement quality, manufacturing, assembly, test, and software development. A program's ability to meet its cost, schedule, and performance goals may be impacted by these processes in either a positive or negative fashion. A disciplined approach to quality is essential to avoid failure. This in turn may affect cost, schedule, and in the extreme, performance if waivers are necessary. Conversely, if a disciplined process is maintained, product results predictably meet expectations.

6.8 Reliability and Maintainability and Availability

Reliability & Maintainability (R&M) and associated Availability are related specialty engineering activities that help achieve, as much as possible, a trouble-free operation of systems.

Reliability is directed toward assuring that a given design attains the longest possible continued operation and operating life. Maintainability is directed toward achieving the reliability inherent in a design through servicing and maintenance, and efficiently restoring the system to operation should failures occur. Availability, the fraction of time a system remains operational, is a function of reliability and redundancy in critical functions. Higher reliability with redundancy provides systems with higher availability.

All devices and systems have a calculable reliability, depending on the material and operational aspects of the product. While the definition of reliability varies depending on the product, it is generally accepted as "the ability of a component or system to perform its required function under given conditions for a specified interval of time" or, in simpler terms, "a component or system's resistance to failure." This is generally expressed as a statistical probability or frequency of a product's failure in a given amount of time. These reliability indicators include Mean Time Between Failures (MTBF) and Mean Time To Repair (MTTR). In order to decrease the failure or hazard rate of a product, special analysis and testing must be completed. While many industries incorporate these practices into design engineering, logistics engineering, and SE, they are, in fact, a part of the field of reliability engineering.

As is often the case in complex systems, like SMC's space portfolio, the finished product is a combination of individual subsystems and their components, each one with its own reliability. The various reliabilities are factored into the reliability of the product or system as a whole. When one component fails, the system either loses its ability to function entirely or continues to function with some degradation in performance or functions with a significantly higher risk of catastrophic failure or unforeseen consequences. This does not exclude the probability that the overall system will fail despite the fact that all the components are operational. Since reliability is not solely dependent on a product's resistance to failure, but also its ability to complete its intended function, other factors must be included when calculating a product or system's reliability or availability.

Space systems are faced with the extreme heat of the sun and the near-absolute cold of open space. The vacuum of space, harsh electromagnetic radiation, high-speed protons from the sun, space debris, and free fall environment in orbit present unique hardware and firmware survival issues that can easily disable SVs from performing as intended. Space debris can reach speeds around 17,000 mph. Most space missions are designed to last from 7 to15 years in this environment.

Not only is the environment of space extreme compared to that of earth, in-orbit maintenance is hard, if at all possible, and expensive. In this situation, the line between maintainability and reliability blurs. Even minor satellite failure can become catastrophic. As such, satellites possess smart systems and are designed with redundancy and other failure resistant mechanisms.

Because of such dramatic consequences, often space systems reliability involves a great deal more testing and prototyping. Reliability engineers in such projects often require highly specialized facilities and equipment to simulate the operational environments of space as closely as possible. Since the costs of failure and remodeling are so high, special attention and testing is required to guard against premature failures.

What are deemed as acceptable levels of reliability and costs of failure are largely decided by the program office. The focus of the RAM SED is to improve Reliability, Availability, Maintainability, and Testability. Reliability predictions are then used to justify the proposed parameters and accurately create a reliability model using fault trees and block diagrams to represent the relationship between components of a system and display the correlation between the perceived part-based failure rate and the experimental failure rate. This allows engineers to better understand the failure rate of the unit. There are two analytical methods used to formulate the reliability model:

- **Physics of failure –** This process involves analyzing possible points of weakness in the design of the unit or device. This includes analyzing elements such as chemical corrosion and fracture mechanics. Essentially, this process involves predicting physical weaknesses in the design or material of the product.
- **Parts stress modeling –** parts stress modeling focuses on individual component or device by categorizing and calculating the stress each piece undergoes during operation, and making sure that each piece is capable of surviving to fulfill its function.

For most complex systems like SMC's space programs analytical modes of failure are hard to develop. As such, empirical data for similar devices is collected and analyzed statistically to augment physical methods to predict reliability. For example, integrated circuit boards built with similar materials, processes, and workmanship are likely to have similar failure modes given similar operational environments. Data collected over the history of use of such similar products is extensively used in the aerospace industry to help predict parts, component, and subsystem reliability.

In addition to actually testing the unit as a whole, reliability engineers are charged with deciding which tests are necessary and the required resources to complete the tests efficiently and cost effectively. Since multiple or long tests are expensive while others are impractical and unnecessary, reliability engineers devise an effective test strategy that is both realistic and affordable. Depending on necessity, engineers can test the system, subsystems or individual components in order to gauge the reliability of the product. The length or number of tests performed on the product is determined by the desired statistical confidence level of the failure rate. If the product does not need a particularly high confidence level or only needs to complete its function for a given amount of time, relatively short testing will be done on the product. In these cases, different methods are employed, such as accelerated life testing or simulations.

Factors Affecting Reliability:

- **Environmental –** The more immediate and preventable threat is the stress caused by the operating environment of the product. If a company produces a submarine and forgets to make the craft waterproof, the reliability is going to be pathetically low. More reasonable examples of this factor are chemical corrosion, extreme temperatures, high acoustic and vibrational loads, and extreme pressures. Most environmental hazards can be prevented in the development stages and can be easily remedied by altering the material and making it more durable.

- **Stress-related –** Depending on the life-cycle of the product in question and the function of the individual components, each part undergoes a certain amount of stress. There is no such thing as a product that is infinitely reliable; all units are subject to wear and tear over time. However, should an engineer wish to increase the reliability of a product, it is key to identify which pieces are most prone to failure, and therefore, are sources of unreliability.

Various methods are used by engineers to bolster the reliability of the product. Fail-safe devices, fault tolerant systems, and repair systems are all functions that increase the maintainability of a product, which affects reliability to an extent, though they are only used in situations where it is considered cost-effective. More practical methods include:

- **Redundancy –** This method involves developing an alternate or substitute path to succeed in the event of failure of a given component. This method is favored because it requires relatively little testing, and greatly increases the reliability of a product by allowing it to withstand failure. Redundancy is employed extensively to improve reliability of space systems where maintenance missions are too expensive.
- **Component derating –** In this solution, the engineers analyze the environment of operation and use components and materials with significantly higher tolerances than necessary to increase the chances of success.

While reliability engineering shares many aspects with safety and maintainability engineering, it differs in that reliability engineering is often concerned with the implied costs of failure and maintenance. These goals may overlap at times, but in many situations improving safety is not the same as improving reliability. For example, adding fail-safe devices and components may increase safety, but they could lower reliability as they result in additional maintenance and additional costs.

Software is prone to different kinds of defects and attacks than hardware, and a different kind of reliability is required. Software issues are generally not the result of environmental or stress threats, but are inherent in the coding of the programs. Thus software reliability does not concern itself with the failure rate, but the number of faults present in the lines of code. Often the failure frequency can be expressed as the number of mistakes or defects per thousand lines of code. The only other source of unreliability in software engineering is unexpected results from the coding of the program. In this case, a specific combination of inputs may result in system errors or crashes. Given the sheer number of possible combinations, these few inputs are difficult to find, and thus software is tested extensively.

Quality engineering, like reliability, demands rigorous testing of the product or system in question, forcing it under severe stress to expose unseen weaknesses and collect data. Both disciplines are driven to create improvements in the system itself and the processes involved in the system's creation. In fact, many quality engineers consider reliability, safety, and maintainability among the parameters of a product when improving its quality. Again, the driving difference between the two practices is the latter's concern with cost and benefit of improvements.

6.9 Spectrum Management

Spectrum Management (SM) engineering provides for the planning, engineering, administration and coordination of joint use of a range of frequencies of electromagnetic (EM) radiation by subsystems and equipment that radiate or receive EM energy. EM or radio spectrum, ranging in frequencies from 3 KHz to 300 GHz, is a natural resource within national and international boundaries that the DoD must share with the civil and commercial entities for wireless communication. Advances in modern technologies in recent years and a shift in joint warfighting strategies have demonstrated a proliferation of potential conflicts resulting from the increased use of the EM spectrum by both government and non-government users. As a result, the requirement for proper analyses and management of the use of the spectrum has risen dramatically in response to the increasing demand for its application.

The SMC Program Office SM Engineer is the designated authority responsible for establishing and executing the SM program. The SM Engineer is also responsible for planning, administering and ensuring essential SM engineering and management efforts are integrated with the various acquisition, management, and SE activities. In addition, the SM ensures adherence to effective and compliant contributions with respect to the various policies, DoD mandates, instructions, and SMC acquisition program and technical objectives, while implementing the program strategies and plans within the SM Engineer’s realm of responsibilities.

6.10 Concept Development

Concept development begins with the JCIDS process as an ICD is developed from capability need and gap assessment, joint operating concepts, joint functional concepts, and joint integrating concepts by the combat commanders. Representatives from impacted DoD communities examine multiple concepts to optimize the way the DoD provides the intended capabilities. A system concept development effort is then initiated once a capability shortfall or an emerging or evolving change to a military threat is identified and it is determined that a new or revised system is required to meet the challenge.

This SED delineates SPO tasks and products in support of the development of the operational and system concepts. A proper concept development effort facilitates subsequent program phases that define, produce and deliver materiel solutions within an identified trade-space in support of capability needs analyses.

Developing a concept for a new or improved system requires the application and rigor of the SE process that responds to a new or evolving operational needs or deficiencies. While a top down flow of activities appears in a typical concept development cycle, in reality the concept development is more often than not an iterative process with multiple iterations and influences from diverse stakeholders. Concept developers typically work as a team, often led by a systems engineer, where each member brings his or her unique expertise as operator, sustainer, technologist, engineer, program manager, or a member of the acquisition community. A Concept Developer plans and performs the essential engineering conceptual development and management efforts to ensure that the resulting program is timely, adequate, consistent, and compliant with the military need.

6.11 Architecture Engineering

Architecture Engineer assists the SE team capture system functional, physical, logical, performance, and behavioral requirements by developing various system artifacts. By creating dynamic system models, abstractions of a particular domain concept, and/or models, the Architecture Engineer assists the Systems Engineers to define system functions, parameterize requirements to perform the functions, and provides functional and physical decomposition of design trades and interface definition and decisions. Systems modeling and analyses is a time-proven approach that reduces requirements creep, reduces costly engineering changes and keeps control of the development schedule by identifying and mitigating technical risks early. In addition, architecture products are essential to describe operational and system concepts, operational environments such as production, integration and sustainment, and mission scenarios. Enterprise-level net-centric operations, interoperability, reuse, and information assurance mandates necessitate a SPO's upfront investment in the development of architecture products to meet DoD-wide pervasive requirements in a timely manner.

The Architecture Engineer for the SPO plans and executes the essential architecture development and management efforts in an integrated and effective manner. The architecture team also supports engineering efforts, acquisition activities, and management activities and decision making.

6.12 System Safety Engineering

Systems Safety Engineering is a well-defined and established discipline and it is one of the engineering disciplines inherent in the multi-disciplined Environment that includes Safety and Occupational Health (ESOH). The Safety Engineer establishes, plans, and executes the system safety engineering and management efforts in an integrated and effective manner.

Various authors and handbooks define system safety as the application of engineering and management principles, criteria, and techniques to achieve acceptable mishap risk, within the constraints of operational effectiveness and suitability, time, and cost, throughout all phases of a systems life cycle. Thus, as stated in the MIL-STD-882D (NOTE: SMC recommends using MIL-STD-882C), the Department of Defense is committed to protecting all personnel—private, or military—from any injuries—fatal or non-fatal—or damage to public or private property that can be caused by occupational illness or materials or facilities or National Security Systems while executing the DoD missions of national defense. The DoD is committed to complying with all laws and regulations to the maximum extent practical. The system safety approach to managing these risks is paramount to ensuring the DoD objectives as they relate to NSS efforts. The DoD system safety approach includes but is not limited to, mishap risk management consistent with mission requirements as implemented in all development and operational phases of the weapon system.

The system safety approach conforms to the acquisition procedures in the DoD 5000 series and provides a consistent means of evaluating identified system risks. Risks must be identified, evaluated, and mitigated to a level acceptable (as defined by the system user) to the appropriate milestone decision authority, and compliant with federal laws and regulations, Executive Orders, treaties, and agreements. Program trade studies associated with mitigating mishap risk shall consider total system life cycle cost in any decision. Residual system safety risk associated with an individual system must be reported to and accepted by the appropriate authority as defined in the DoD 5000 series. The DoD program office teams should review MIL-STD-882D (NOTE: SMC recommends using MIL-STD-882C) and include tailored versions to meet their program's system safety risk mitigation objectives.

The SPO documents system safety approach in the System Engineering Plan (SEP) and the contractor documents their approach in the SEMP. If warranted the SPO may direct the contractor to develop a specific system safety plan for nuclear systems, high explosives, flight worthiness certification, or experimental systems. The plan identifies all possible known hazards and the associated risks along with mitigation plans to reduce the risks to an acceptable level if it cannot be eliminated. Specifically, the safety plan:

- is consistent with all applicable policies, directives, and instructions, and overall program acquisition strategy and objectives.
- defines and allocates system safety requirements to include (i) safety design requirements and associated safety design criteria, (ii) identification and analysis of potential hazards, (iii) identification,

analysis, and optimization of safety risks in design, materials, testing and production of end item, and (iv) minimizes safety related rework by inclusion of features during definition and development of the system,

- isolates hazardous substances, components, and operations to improve personnel and materials safety
- eliminates catastrophic risks
- minimizes critical hazards

6.13 Acquisition Systems Protection & International Program Security

Program protection planning is integral to the overall acquisition strategy and is performed at the front end of the program. The program identifies Critical Program Information (CPI) that applies to components, engineering, design, manufacturing processes, or technologies that, if compromised, could cause significant degradation in mission effectiveness, shorten the expected combat-effective life of the system, or reduce technological advantage. The Program Protection Engineering organization identifies the resources needed to accomplish the evaluation and initiates protection, if required, at the earliest possible opportunity.

The required extent of program protection plan (PPP) depends on the nature, significance, and criticality of CPI identified for the program and the overall mission. If no CPI is identified, only a program letter stating the fact is required, which is then approved by the decision authority. If it is determined that the program contains CPI, a PPP is required during the technology development phase and made available to the decision authority at Milestone B. For cooperative programs with foreign participation, a technology assessment and control plan (TA/CP) is required IAW DoDI 5200.39, DoD 5220.22M, DoDD 5530.3, and DoDI S-5230.28, as an annex to the PPP. Furthermore, a Delegation of Disclosure Authority Letter (DDL) IAW DoDD 5230.11, enclosure 4, is also produced as an annexure to the PPP.

Depending on the nature and significance of the CPI other documents, usually annexed to the PPP, are required that include (i) Counterintelligence (CI) support plan, (ii) System Security Management Plan (SSMP) IAW MIL-HDBK-1785, (iii) Security Classification Guide (SCG) IAW DoD 5200.1-R and DoD 5200.1-H, (iv) OPSEC plan IAW DoDD 5205.02 and DoDD 5205.02-M, and (v) classified anti-temper (AT) plan. The program protection engineering organization plans and performs essential engineering tasks to comply with DoD mandates and SPO requirements.

6.14 Survivability Engineering

A sound survivability engineering organization is required during all phases of acquisition to ensure that space systems can survive and operate under natural and hostile battlefield conditions including nuclear, chemical, biological, conventional, blast and fragmentation, radiological, electromagnetic, and natural environments. Survivability engineering and management, accomplished early in the acquisition phase, influences the selection of the preferred concept, technologies, the eventual design, and identifies additional support resources required to maintain system readiness. Survivability engineering also plans and provides design inputs to help enable rapid restoration of the system, subsystem, component, or equipment when compromised to improve sustainability of the operations. Both manmade and natural causes must be factored into survivable system development, design, and operation.

SMC SPO Survivability Engineering organization supports the full range of concept development, requirements development, engineering analyses and trades, system design and system survivability verification and validation. The SPO survivability organization also pursues survivability related technology advancement leveraging research facilities, organizations, and projects. For example, the survivability program leverages radiation-hard electronics technologies and industrial base information through the DoD

Radiation Hardened Oversight Council (RHOC) as well as commercial sector manufacturers, government labs, and other government agencies. Survivability is the capability of a system to operate without degraded performance if exposed to adverse natural and/or hostile environments. System survivability extends to include the personnel and their interactions essential to operate the system, usually relevant to ground segments. Survivability and force protection Key Performance Parameters (KPPs) generally apply to SMC ground systems and missions that rely on human interaction to protect and safeguard the warfighter. System survivability is also extended to a system's supporting infrastructure (facilities, basing, subsystems, etc.) and other interfacing systems. AFSPC requires all new space acquisitions to address space protection requirements (including survivability) as KPPs or Key System Attributes (KSAs), where appropriate.

These requirements are to be based on system specific validated threats and vulnerabilities. Comprehensive analysis of threat, impact of loss, countermeasures, and cost are required to determine how the KPPs or KSAs are correlated to the protection of the space, link, ground infrastructure, or cyber system components. The Personnel Survivability and Force Protection are mandatory KPPs generally applicable to ground systems. JCIDS Manual, enclosure B, states "[survivability] includes attributes such as speed, maneuverability, detectability, and countermeasures that reduce a system's likelihood of being engaged by hostile fire, as well as attributes such as armor and redundancy of critical components that reduce the system's vulnerability if it is hit by hostile fire." This aspect of survivability is addressed by the Human Systems Integration (HSI) SED.

For SMC systems acquisition, survivability has been defined as the capability of a space system to avoid or withstand hostile natural and man-made environments without suffering an abortive impairment of its ability to accomplish its designated mission. Natural environmental challenges include acceleration, vibration, acoustics, depressurization, temperature extremes, radiated RF emissions, separation shock, electric charge buildup, material outgassing, orbiting debris, micrometeorites, ionospheric scintillation and ionizing radiation. Natural radiation sources include solar flares, the Earth's magnetosphere and cosmic rays. Man-made threats include electromagnetic jamming, laser blinding, anti-satellite weapons, directed energy weapons and nuclear detonations.

AF Space Command and US Strategic Command place high importance on protection of space systems. The battlefield is directly supported by space systems that provide critical capabilities such as early attack warning, protected communications, Intelligence, Surveillance, and Reconnaissance (ISR), GPS navigation, or weather prediction and reporting. This means space systems must be able to operate in extreme space environments and survive an array of manmade hostile attacks.

Survivability Engineering must carefully catalog and evaluate all survivability threats. Then the possible solutions, costs and other engineering factors must be balanced to produce the optimum survivability solution. Failing to put some of these processes in place risks not finding an optimum solution – either the space vehicle will be over designed for threats that are not likely or the space vehicle will be ill prepared to handle a real threat. Common consequences of not accounting for natural threats are the reduced lifetime of the space vehicle due to accumulation of contaminants on surfaces, radiation damaged electronics, and extreme temperature cycling. Common consequences of not accounting for manmade threats are denial of service due to attacks on communications links or damage to electronics, sensors, or the space vehicle itself due to some form of attack. Failing to survive an attack can mean a space system will not be available for use in the battlefield.

Ensuring survivability on the ground can be equally demanding. Ground systems can be subjected to a long list of natural environmental challenges such as thunderstorms, wind storms, ice storms, snow, attack by wildlife, fungus, fog, blowing sand and dirt, lightning, and seismic events, as well as, man-made threats such as nuclear or conventional, biological and chemical weapons. While these systems must be able to survive these challenges, solutions are not typically as constrained as they are for space systems.

6.15 Human Systems Integration

Within the Department of Defense's acquisition lifecycle framework, SE and program management activities, HSI is a vital component in the DoD's total system approach. HSI is the engineering application of the knowledge of human capabilities and limitations with respect to system or equipment design, development, operations, and sustainment. Its objective is to maximize efficiencies, effectiveness, and safe system performance, while minimizing cost, manpower, skills and training resources. HSI is a comprehensive management and technical strategy to ensure human performance factors are continuously addressed throughout the system life cycle.

When HSI is a significant factor, the SPO establishes and designates an HSI Engineer responsible for managing and executing the HSI program. The appointed HSI Engineer's responsibilities include planning, supervising and ensuring essential HSI and management efforts are integrated with the various acquisitions, management, and engineering processes. The HSI Engineer ensures effectiveness and compliancy to the assorted policies, DoD mandates, instructions, and SMC acquisition program and technical objectives, as they pertain to the implementation of the HSI program strategies and plans.

Personnel Subsystems addresses the factors affecting the man-machine interface. Considerations include Human Engineering and the associated field of Ergonomics, man-in-the-loop requirements, decision processes and automated situation reporting, and an understanding of the intelligence, experience and training of the expected operators. The Project Engineer must include such analysis in candidate space system selections and development. If you require an operator who is less than four feet tall, has only one arm and requires no resources to take care of bodily functions, your chances of widespread acceptance of the system are slim to none. To ensure system objectives are met and personnel safety is considered, HSI must be integrated into all phases of SE: design, manufacture, test, and support. HSI expertise should be integrated in practically every IPT.

HSI is the systematic use of knowledge to achieve compatibility in the design of interactive systems of people, machines, and environments to ensure their effectiveness, safety, and ease of performance. Human Factors requirements help develop effective human-machine interfaces and minimize system characteristics that require extensive cognitive, physical, or sensory skills; require excessive training or workload for intensive tasks.

For space systems, human factors requirements are typically applicable to:

- **Manufacturing and maintenance procedures –** Space systems typically optimize the weight and size constraints over manufacturability or maintainability factors; however, some consideration needs to be given to human factors to ensure the systems can be built and maintained safely and reliably.
- **Launch procedures –** Launching could be a very dangerous function. Launch systems and their supporting equipment are very large, involve highly flammable, poisonous and/or cold materials and require absolute accuracy. Human Factors engineering can contribute to safety and accuracy by making human operations clearer and simpler.
- **User equipment operations –** Some systems (such as GPS) have extensive user equipment. These systems may have hundreds or thousands of pieces of equipment operated by a similar number of users. Human Factors can be used to great effect to simplify the user's interaction and ensure reliable and safe operation of the user equipment.

- **Ground control system operations –** The primary human interface to the space vehicle is via the ground control system. Human Factors engineering can be used with great success to simplify the user's interactions and ensure the safe and reliable operation of the space vehicles.

Human Factors engineering often works closely with:

- **Systems engineering –** trade studies are often done to determine if some operation should be performed manually or automated.
- **Reliability –** the more complex a human-machine interface is, the more difficult it is to operate it reliably. In some cases, it is better to completely automate the interface or use automation to at least reduce the complexity of using the interface.
- **Safety –** confusing user interfaces and operations can cause incorrect operation and lead to critical errors.
- **Training –** human factors engineering has a great impact on training. The more complex a human-machine interface is, the more difficult it is to safely train personnel to use it.
- **Software engineering –** more and more human-machine interactions involve computer based consoles and software programming.

Failure to include HSI early on in the development process risks producing a system that is so difficult to maintain or operate that mission success is jeopardized. Trying to add Human Factors engineering late into the process can result in costly and time consuming rework jeopardizing budgets and schedules. Failing to ever include Human Factors engineering can result in a system that is dangerous and unworkable. Such a system rarely meets its original goals.

6.16 Mass Properties Engineering

Mass Properties is an engineering discipline that is concerned with estimating mass, center of gravity, and inertia values of space vehicles to include the upper stage, injection stages, satellites, satellite payloads, reentry vehicles, launch vehicles, and ballistic vehicles. Mass Properties Engineering organization provides for the control, determination and documentation of the mass properties and mass properties limits of space vehicles and their subsystems and components. The mass properties of an item include the item's weight (or mass), center of gravity (or center of mass), mass moments of inertia, and mass products of inertia in 3-space.

SMC SPO Mass Properties (MP) Engineer ensures effective prediction of the space vehicle mass properties parameters to support performance analyses, stability and control analyses, structural dynamics, load analyses, and other analyses. The MP Engineer establishes and implements a mass properties control program with the objective of meeting the space vehicle mass properties requirements for weight, center of gravity, mass moments of inertia, and mass products of inertia as they apply. The control of weight growth is a continuous activity from system concept development through the last item of production. However, a more restrictive definition usually refers to the technology development and EMD phases when most growth occurs. Costly fixes and possible schedule delays are avoided when weight prediction and control is applied while defining alternative enabling concepts or technologies and determining the preferred solutions. During detailed design, a major effort is required to keep the designers' attention focused on weight efficiency.

The importance of mass is a fundamental and crucial aspect of the evolution of a spacecraft program. Due to the high cost of delivering every pound of a payload to orbit, effective prediction of the space vehicle mass properties is the pivotal foundation of any space vehicle design. Mass properties management begins at the earliest program stages when an effective mass properties management program is initiated to ensure the predicted weight is not exceeded. Mass properties management processes provide for the determination, documentation and control of the mass properties of space vehicles and their subsystems and components.

Valuable tools in weight management are mass budgets for each component, a mass contingency and ballast. The mass contingency is simply an amount of mass kept in reserve for later application to specific components that can’t meet their original mass targets. Ballast is actual mass (often sand or concrete) that is launched as a part of the space vehicle to help achieve (i) overall mass objective, (ii) center of gravity, or (iii) products of inertial for rotational balance.

Mass properties management must be done in conjunction with many other engineering disciplines. The following list shows some of the most important engineering disciplines that are affected by mass properties:

- **Attitude Determination & Control (ADC) –** Mass properties affects overall sizing of ADC components, balance and pointing accuracy
- **Structures and mechanical design –** Mass properties affects the strength requirements of the spacecraft structure and the overall balance
- **Dynamics group Finite Element Model (FEM) –** Mass properties data is utilized in FEM to determine vehicle dynamic response.
- **Propulsion –** Mass properties determine the amount of fuel needed for maneuvering and attitude control and the moment arm of the SV thrusters.
- **Survivability –** The mass properties group works with the survivability group for optimum shielding and hardening methods.
- **Parts, Materials and Processes (PMP) –** Mass properties works with the mass properties group to select materials to help reduce weight.

6.17 EMI/EMC Engineering

Each system must be designed to operate in natural or man-made EM environment. It must be compatible to all defined EM environments to which it is intended to be exposed. EM environments can be natural or man-made, unintentional or hostile to include radiation environment in space, EM interference (EMI) from within or outside the system, electrostatic discharge, and EM pulse. The system must operate in various operating modes and mission phases, while working harmoniously in concert with other systems. EM radiation can impact performance and health of other electronic subsystems and the personnel that operate or interact with the system.

EM compatibility (EMC) involves the control and reduction of EMI. Space and ground systems must be able to operate and remain free of overstress and anomalies caused by either intentional or extraneous EM energy emanating within or outside the system from man-made or natural sources. EMC refers to the capability of electrical and electronic systems, equipment, and devices to operate in their intended electromagnetic environment within a defined margin of safety, and at design levels of performance, without suffering or causing unacceptable degradation as a result of EMI. EMI is concerned with electromagnetic disturbances that interrupt, obstruct, degrade or limit the effective performance of electronic or electrical systems, transmission channels, equipment or devices.

EMI/EMC Engineering is intimately involved with the testing of all electro-mechanical components to include power cables and harnesses, antennas, and electronics. Emitted radiation and susceptibility to EMI/EMC is measured and documented. Component performance tests are also conducted in expected operating EMI/EMC environments.

EMI/EMC Engineering is responsible for developing and implementing a thorough and comprehensive Electromagnet Control Plan based on early and continuous evaluation of the evolving system concepts and providing feedback to the design functions and IPT trade study activities. Responsibilities include (i) developing the appropriate design criteria based on performance requirements and environmental exposure, (ii) defining test requirements to develop, qualify and verify conformance to the EMI/EMC functional requirements, (iii) identifying and mitigating program risks, (iv) identifying resources required, (v) preparing integrated schedules and cost goals to accomplish the required tasks, and (vi) monitoring performance to assure parameters critical to program goals and systems operational effectiveness are satisfied throughout the systems life cycle.

6.18 Parts Materials, and Processes Engineering

PMP engineering is responsible for the application, selection, qualification, procurement, documentation and disposition of all parts, materials and processes required to implement the system design. This includes all flight, qualification, proto-qualification, and deliverable ground segment hardware.

PMP engineering participates in all phases of the program life cycle. The level of PMP engineering participation depends on the type of and application of the system being acquired. Typically PMP engineering involves three engineering disciplines: Parts Engineering, Materials and Processes Engineering, and Contamination Control Engineering. A typical PMP organization is shown in Figure 6-2:

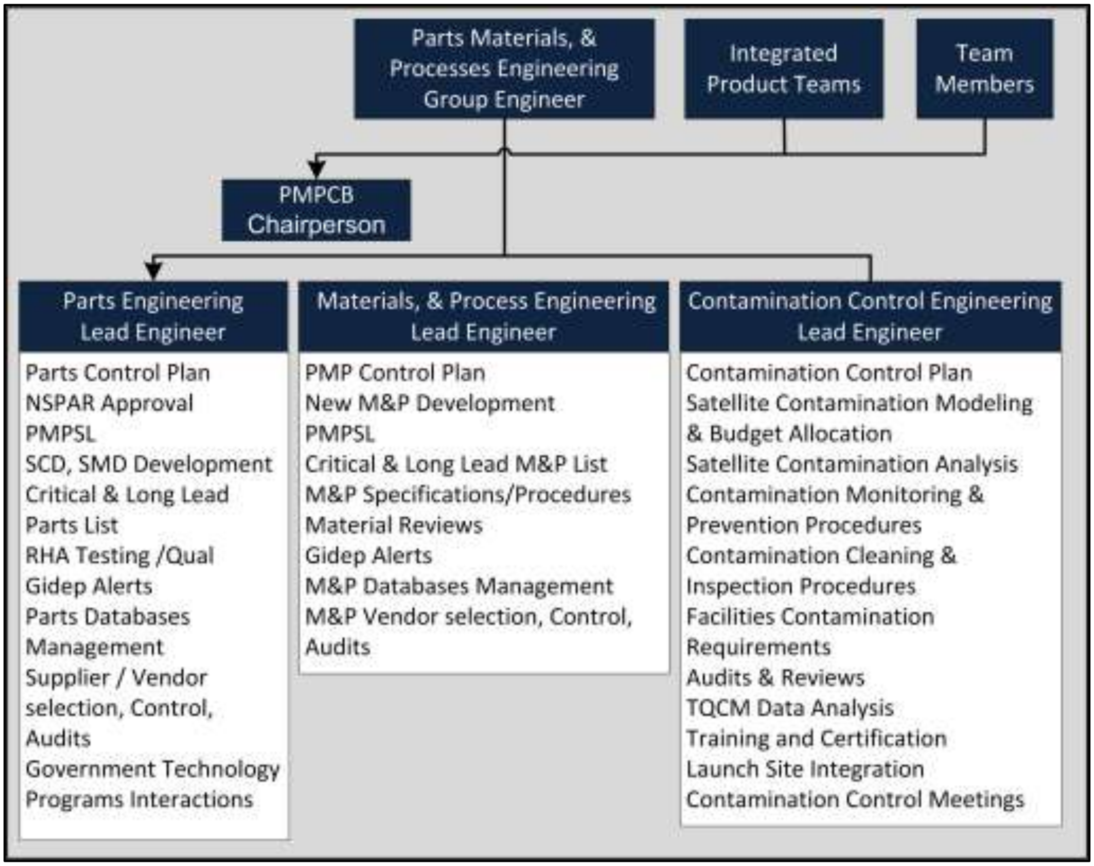


Figure 6-2 PMP Typical Functions, Roles and Responsibilities

At SMC, PMP Engineering activities include implementation of the PMP Selection and Control program. A parts, materials, and processes selection and control program is vital to:

- Ensure integrated management and balanced technical decisions regarding selection, application, acquisition, control, and standardization of parts, materials, and processes
- Improve acquisition and qualification of piece parts, materials, and critical processes that meet system requirements
- Implement the reliability program at the PMP level to reduce PMP failures at all levels of integration, assembly, test, and operations
- Reduce program life-cycle costs

The PMP engineering organization is responsible for implementing and operating the Parts, Materials and Processes Control Board (PMPCB). The purpose of the PMPCB is to (i) resolve PMP selection, qualification and procurement issues, (ii) review, maintain and track changes to the As-Designed Parts, Materials and Processes List, (iii) disposition discrepant or suspect PMP and (iv) assess, identify, mitigate, track and report on PMP risks for the entire program. The representative from the program office may be a member of the PMPCB and an active participant.

The PMPCB coordinates all PMP across all SPO organizations (both internal to the contractor and their subcontractors, suppliers and vendors). The PMPCB is designed to provide both horizontal and vertical lines of communications by allowing the participants to identify concerns, issues and problem areas, quickly determining their impact and informing others within the program organization including the program office. Also, lessons-learned for one organization can be applied to other organizations to reduce overall development time and reduce acquisition costs. With new technologies and high product obsolescence as well as emphasis on process validation rather than product test and qualification, it is very important that these past experiences and lessons-learned are taken in consideration.

PMP’s implementation and integration within the other specialty disciplines varies consistent with contractor’s organizational structure and each program phase. The actual integration and requirements decomposition process for each physical element within each program phase forms the contractor's proposed systemic approach to implementation of Mission-need PMP Program Requirements.

An effective PMP Program should define two levels of implementation and performance. The first level constitutes the contractor’s internal PMP Process activities. The second level constitutes the contractor’s proposed control and flow-down of PMP Requirements, to their outside suppliers or subcontractors and activities, to ensure uniform PMP Program implementation.

For space programs, an Integrated Program Team or the Parts, Materials, and Processes Board (PMPCB) are traditionally established as the vehicle for PMP process integration and interface with all necessary disciplines and control of outside vendors and subcontractors throughout all program phases.

Furthermore, space industry maintains a comprehensive list of PMP and their level of space qualification as codified by the Technology Readiness Levels (TRL). These TRLs are described in Appendix D. With the help of the industry, SMC continues to maintain a database of space industrial base as described in Appendix E.

6.19 Information Assurance Engineering

DoD defines Information Assurance (IA) as "measures that protect and defend information and information systems by ensuring their availability, integrity, authentication, confidentiality, and non-repudiation. This includes providing for the restoration of information systems by incorporating protection, detection, and reaction capabilities (CNSSI No. 4009)." All SMC programs classified as National Security Space Systems (NSS) are required to comply with IA mandates derived from Clinger-Cohen Act and the DoD 8500 series.

NSS that include space vehicles (SV) and ground satellite operations control (SOC) are required to comply with Net-Ready Key Performance Parameters (NR-KPP) that encompass DoD-wide interoperability and supportability with crucial IA) to protect the information advantage in an internetworked environment. This secure internetworking, combined with changes in technology, organization, processes, and people allows new and robust forms of organizational behavior.

As illustrated in Figure 6-3, IA organization must protect against an adversary's ability to get in, stay in, and act[44] – shrinking any of these areas reduces the level of impact on a program's network infrastructure and information. The trick is to maintain a strong IA posture in a complex and constantly changing environment, where people, technology, and computer network defense operations are constantly in a state of flux.

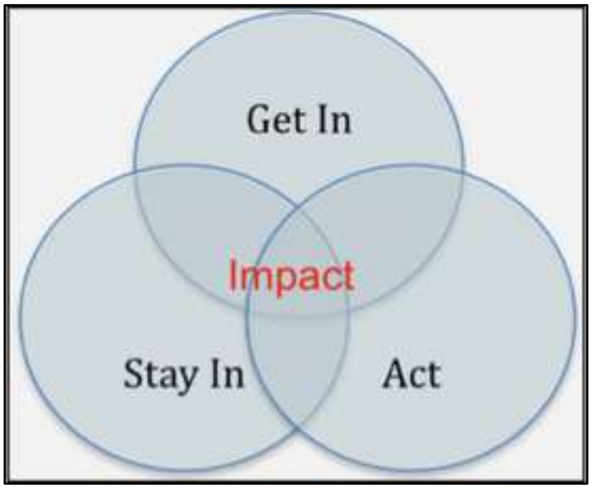


Figure 6-3 Cyber-attack model (from Applied Physics Laboratory, Johns Hopkins University, Maryland)

Identifying cyber-attacks on a system or network is just the beginning of the critical task of actually fixing the vulnerability found through patching, updating, or reconfiguring. The mitigation process is beset by the false positives introduced by security tools, growing number of vulnerabilities, labor intensive manual auditing, and the exposure-time between discovery and mitigation. The various categories of cyber-attacks include

- Distributed Denial of Service (DDoS), Jamming, availability
- Spoofing, hijacking, link/data integrity
- Viruses, worms
- Authentication, selective availability
- Insider attacks

A SPO's IA organization performs engineering over the entire lifecycle of the program to protect, monitor, analyze, detect, and defend against unauthorized activity within its information systems and computer networks. IA organization attempts to fulfill the need to access information with robust and assured confidentiality, availability, integrity, and authentication in a policy-driven, internetworked, agile, always-on, publish-subscribe, net-centric environment.

---

[44] Cyber-attack Model from National Information Assurance Engagement Center, Applied Physics Laboratory, Johns Hopkins University, Maryland

IA organization develops computer network attack/defense scenarios using context-free grammar, attack-graphs, topologies, or taxonomies. The field as yet is not mature and lacks standard tools and techniques to identify, describe, and classify cyber-attacks that are accepted industry-wide. While some automated tools have been developed, the security modeling, simulation, and analysis continues to be tedious manual work for the most part. Furthermore, a unique part of the space systems is their inherent need for space-ground wireless communications for Tracking, Telemetry, and Command (TT&C). IA input early in the development and design phases of the program can help mitigate loss or corruption of information against this vulnerability by introducing cost-effective technologies like use of open standards for end-to-end SOC to SV encryption to minimize hostile entry points. Other standard techniques to guard against network attacks include (i) automated security configuration management, (ii) IA status alerts, (iii) host based security system (HBSS), (iv) adware/spyware detection and eradication, (v) user defined operational picture for situational awareness, (vi) insider threat detection tools, (vii) intrusion protection, (viii) data encryption, (ix) user cyber-attack awareness training, (x) hardware tokens, and (xi) web content filtering.

6.20 Net-Centric Engineering

Network-centric operations offer a competitive edge to well-informed but geographically-dispersed units or entities by pervasive networking to enable enterprise-wide sharing of information. This new SED focuses on five key areas to realize a net-centric information sharing vision: (i) data and services, (ii) secured availability, (iii) computing infrastructure readiness, (iv) communications readiness, and (v) network operations agility.

Net-centric engineering is responsible for the development and design of SPO solutions for interoperable enterprise-wide communications capability. Net-centric engineering helps derive necessary requirements, select DISR and other open standards, provide technology solutions, and procure COTS products or guide development of new software to achieve net-centric operations. Net-centric engineering assures that SPO services and systems are secure, reliable, interoperable, and able to communicate across a universal information infrastructure based on Internet Protocol (IP) and related non-proprietary and vendor-neutral standards. This internetworking, combined with changes in technology, organization, processes, and people allows new and robust forms of organizational behavior.

All SMC programs classified as NSS are required to comply with CJCSI 6212.01 that mandates net-ready key performance parameters for interoperability for new programs and upgrades. Net-centric Engineering vision is a major and enabling part of the enterprise-wide interoperability and supportability requirements. SMC in conjunction with SPAWAR and other organizations have developed implementable guidance that is available through the Net-Centric Enterprise Solutions for Interoperability (NESI). This information is available to the SPO net-centric engineering which can be tailored as necessary to develop acquisition artifacts including net-centric data/services strategy, ASP, and ISP.

6.21 Environmental Engineering

Environmental engineering is a well-defined and established SMC discipline. Environmental Engineers' activities include implementation of ESOH mandates and best practices. The intent is to identify potential environmental, safety, and operational health problems as early as possible in the product lifecycle to provide greater opportunities to eliminate hazards. As design decisions are made and the development efforts transition to production and fielding, ESOH related design improvements may be orders of magnitude more expensive. SPO's Environmental Engineering organization's objectives include:

- Establish environmental, safety, and operational health requirements based on public law, Government policy and mandates, operational constraints, and SMC practices.

- Propose technical solutions and evaluate the inherent ESOH implications of proposed technical solutions to influence technical decisions to meet the environmental, safety, and operational health requirements.

The Environmental Engineering plans and executes the essential environmental engineering and management efforts in an integrated and effective manner.

SMC takes a firm and compliant position with its legal obligations concerning Environmental, Safety and Health (ES&H) issues. It has developed and implemented policies for application, and documented instructions, to ensure all new proposed and supplemented versions of space and missile weapons systems have been reviewed and subjected to technical scrutiny for potential environmental impacts.

Environmental Engineering leads the Environmental Impact Analysis Process (EIAP) which utilizes a systematic, interdisciplinary approach to environmental analysis that is common to good SE. The EIAP process identifies the SPO as the proponent for all environmental actions related to, proposed, new, and/or revised space systems. The EIAP analyzes potential air quality impacts including installation of compatible use zones, all new facilities, water resource endangerments, safety and occupational health, hazardous materials and waste, biological resources, cultural resources, geology of soils, socioeconomic issues including environmental justice. Analyses may also include non-ionizing/ionizing radiation, de-orbiting debris, noise and sound. All Acquisition Environmental Documentation must be coordinated through members of the SMC Environmental Protection Committee for their review, comments and approval.

PESHE Guides, Charters, Checklists, Risk Analyses, NEPA requirements and the final PESHE product are all ‘living documents’ that are regularly reviewed and updated as needed throughout the life time of the program by the SPO.

6.22 Prognostics & Health Management (PHM) Engineering

Historically, space and airborne systems have used built-in-test (BIT) capability to provide:

- Fault detection
- System (or equipment) response to the fault
- Fault event warning and/or logging to aid in troubleshooting

However, BIT designs typically provide after-the-fact passive man-in-the-loop fault diagnostics and management. For launch, space, and missile systems, unique constraints (remote systems, minimal event response time, autonomous safety, harsh environments) drive the need for more sophisticated and autonomous PHM.

As the system design is engineered, failure precursors, which indicate changes in a measured variable that can be associated with impending failure, are systematically identified. An active PHM design solution includes automated monitoring of the failure precursors, prognostics, and fault correction. PHM provides the capability to make intelligent, informed, and appropriate decisions relating to system faults within and across systems during system development, integration and test, and operations and sustainment. A solid PHM program will also provide cost savings over the system life-cycle. Key attributes of PHM include real time or near real time health status availability; proactive advisory generation based on health state; autonomic logistics (reduced human interaction); no or minimal false alarms; and autonomous fault management to preclude safety mishaps, performance degradation, and catastrophic failures. The applications of PHM include:

- **Spacecraft –** Spacecraft PHM capabilities include autonomous health and operations monitoring and control; power and attitude control monitoring with automated systems reset and restart; transmitter and receiver and communication link tests; automatic reset features to restart remote computers.
- **Launch/Missile Systems –** Pre-launch failure detection, notification, and response for abort determination, command destruct /self-destruct, stage event monitoring and diagnostics; communication systems link tests
- **Other Electronic Devices –** autonomous health and operations monitoring and control, BIT during manufacturing

7 Verification and Validation

Verification and validation activities starting with the early SE and running concurrently over the entire lifecycle of the system are crucial to the success of the program. Concept engineering teams assembled to explore and refine concepts include V&V and test and evaluation engineers to help make decisions to initiate and proceed with a program. Figure 7-1 shows iterative yin-yang of V&V loops within the overall SE lifecycle process.

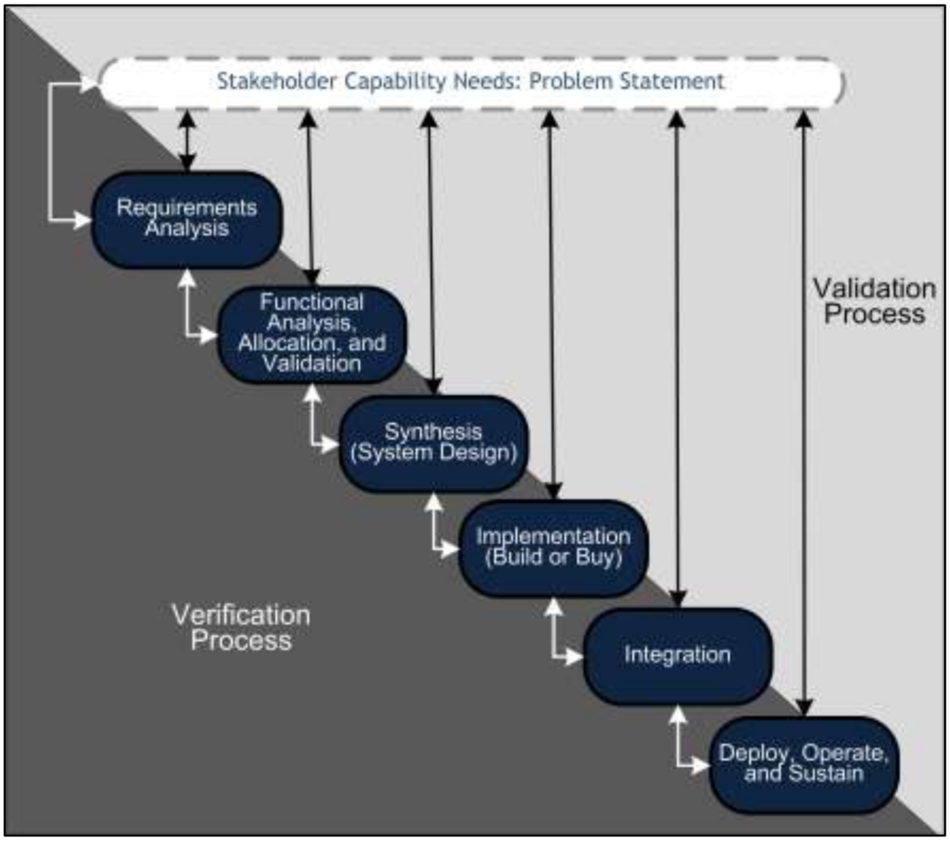


Figure 7-1 V&V in an iterative SE lifecycle environment

In SE, verification establishes the fact that a system, subsystem, or a component meets its specification and allocated baseline requirements as derived from and traceable to the stakeholder needs. Verification assures that the system is built right. A good practice is to develop verfication criteria as the the requirements are written and analyzed. This criteria is further refined with the requirements as the program moves toward maturity and eventual deployment.

Validation is intended to confirm that the right system is built. Validation shows that the stakeholder needs were accurately captured, designed, and implemented – that the delivered system opertes and fulfills real customer needs. Validation criteria, scenarios, and procedures are gathered and refined as customer needs are converted to verifiable requirements. These validation methods are used to forecast and eventually demostrate that the system produced performs as expected by the customer.

7.1 V&V Methods and Techniques

V&V results obtained over the development, design, production, and integration processes provide incremental assurance that the product will pass the customer's criteria. Eventually these results provide proof that the product performs as specified, and provide an indication of how well the product satisfies operational needs. Major categories for V&V activities include the following methods:

- **Inspection –** Inspection consists of direct examination of system, system constituents, or artifacts that may include documentation presenting the results of prior lower level verifications, drawings, vendor specifications, software version descriptions, documents, or computer program code. A direct physical attribute such as dimensions, weight, physical characteristics, color, or markings may also be examined. Inspections determine conformance to requirements by the visual examination of drawings, data, or the item itself using standard quality control methods, without the use of special laboratory procedures or equipment.
- **Analysis –** Analysis is the evaluation of data by generally accepted analytical techniques to determine that the item will meet specified requirements. Analytical techniques may include SE analysis, statistics, and qualitative analysis, analog modeling, similarity, and computer and hardware simulation. Analysis may also include assessing the results of lower level qualifications activity. When it is not required or cost-prohibitive exhaustive tests or demonstrations to show compliance, analysis can be used to extend results of limited test data to show full compliance. For example, if the integrated system or subsystem is too large for the thermal-vacuum chamber, results of from thermal-vacuum testing of components may be analyzed for acceptance. Thus, analysis is employed when test or demonstration techniques cannot adequately or cost-effectively address all the conditions under which the system must perform or the system cannot be shown to meet the requirement without analysis.
- **Test –** Test is a method in which technical means, such as the use of special equipment, instrumentation, simulation techniques, or the application of established principles and procedures, are used for the evaluation of the system or system components to determine compliance with requirements. Test consists of operation of all or part of the system under a limited set of controlled conditions to determine that quantitative design or performance requirements have been met. It includes the collection and subsequent examination of quantitative data to make that determination of compliance. Tests may rely on the use of elaborate instrumentation and special test equipment to measure the parameter(s) that characterize the requirement. These tests can be performed at any level of assembly within the system assembly hierarchy. Test is selected as the primary method only when test activities produce results that are necessary and sufficient to show compliance. The analysis of data derived from tests is an integral part of the test program.
- **Demonstration –** Demonstration consists of operation of all or part of the system under a limited set of controlled conditions, or the qualitative determination of the properties of a test article, to determine that qualitative design or performance requirements have been met. Demonstration relies on observing and recording functional operation not requiring the use of elaborate instrumentation, special test equipment, or quantitative evaluation of data. Elaborate instrumentation is any instrumentation beyond the inherent capabilities of the system to record and display information. Demonstration data may be generated by test events at any level of assembly. Demonstration generally verifies system characteristics such as human engineering features, services, access features, and transportability. Demonstration requirements are normally documented within a test plan for a specific test event, operations plan, or test procedures. Demonstration data or results may be obtained during program test events or dedicated demonstration activities. Demonstrations are used to show successful completion of an action, either by the system/component or upon the system/component, and may be associated with some aspects of dependability requirements that include reliability, maintainability, safety, human systems integration. Specialty engineering disciplines and related V&V requirements and documentation needs are discussed in detail in the SEDs handbook.

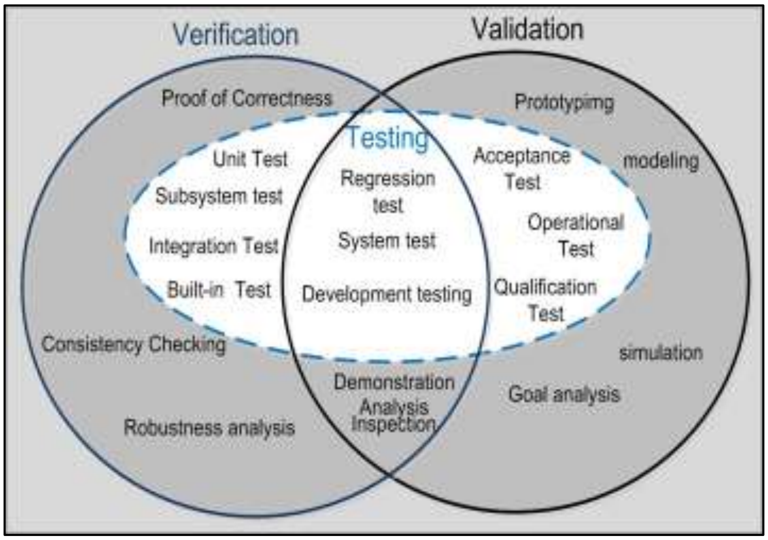


Figure 7-2 V&V and related testing

To ensure a satisfactory conclusion to the V&V process, it is necessary to plan early in the development life of the program. V&V plans must be established to provide adequate direction for system engineers to complete the process. As an example, the Advanced EHF program built requirements V&V plans prior to the signing of the EMD contract. These plans described in detail how each individual requirement was to be assured. Information in the plan included: the requirement and its identification number (traceable through a database tool to higher or lower level requirements); any other requirements which may be verified together; verification approach (i.e., analysis, test); which test series would be used to verify or what analysis tools would be used; for analyses, was information required from a particular test to support the analysis; assumptions; inputs; outputs or expected results; and test sets required. Eventually, when V&V is completed for each requirement the individual V&V plans include links to analytical results or test data that satisfy the V&V of the requirement. This is a very good, well thought out approach to ensuring requirements are met.

As shown in Figure 7-2, both verification and validation strongly rely on various types of testing as described below. However, verification testing is geared toward showing that the baseline requirements for form, fit, and function are met. In this regard verification also employs demonstration, inspection and analysis that may include:

- **Consistency checking –** to show how well the implementation correspond to the model
- **Proof of correctness –** use of mathematical procedures to verify correct system behavior, often intractable and expensive
- **Robustness analysis –** usually employed early in the program to evaluate performance of technical or architectural solutions to show repeatable performance characteristics within requirements

Other than standard V&V tools and techniques as shown in Figure 7-2, validation typically depends on modeling, simulation, prototyping, and goal analysis to show that customer needs are realizable, and if so, how well given the maturity of applicable technology, its cost, and the desired schedule.

Table 7-1 lists some of the considerations involved in V&V control. Those associated with Verification are fairly well integrated into engineering practices, since they have been in general use and are often contractually required. The Validation controls are less well understood and implemented. Their major thrust is to document results, to integrate the results into all design decisions, and provide traceability from the designs to the related analyses. This process ensures that anyone making future changes is aware of all the factors that shaped how a particular design evolved, and can avoid possible counter-productive decisions. Recently relational database tools have been developed which assist in this process. Making such databases available to all cognizant functions though an electronic network enhances the probability of arriving at an optimum design. SE is often the instigator and curator of the database/network combination.

Table 7-1 Validation and verification control considerations

Verification	Validation
Document preparation properly supervised and approved.	Analyses properly identified and defined prior to start.
Documents are under configuration control.	Analysis results documented and cataloged for traceability.
Non-conformance identified and analyzed.	Analysis results disseminated to design/ specialty disciplines.
Measuring/test equipment calibrated to traceable standard.	Design decisions traceable to associated analyses.

7.2 Test and evaluation

Testing increases confidence in meeting customer requirements and is part of overall risk reduction. The complete test program for launch vehicles, upper-stage vehicles, and space vehicles encompasses development, qualification, acceptance, system, prelaunch validation, and post-launch validation tests. Developmental tests are conducted to obtain data on the operational characteristics of the test subject for use in design decisions, and are a primary part of Validation. Qualification or acceptance tests are conducted to show proof that particular designs or particular units meet design specifications and are the purview of Verification.

Test methods, environments, and measured parameters shall be selected to permit the collection of empirical design or performance data for correlation or trending throughout the test program. See SMC Standard, SMC-S-016, "Test Requirements for Launch, Upper-Stage, and Space Vehicles," 2008, or MILHDBK-340, Volume II for further guidance.

A satisfactory test program requires the completion of specific test objectives in a specified sequence. The test program encompasses the testing of progressively more complex assemblies of hardware and computer software. Design suitability should be demonstrated in the earlier development tests prior to formal qualification testing. All qualification testing for an item should be completed, and consequential design improvements incorporated, prior to the initiation of flight hardware acceptance testing. In general, hardware items subjected to qualification may be eligible for flight, provided suitable analyses, refurbishment and verification are completed. The test plan for verification follows the pyramid test philosophy, i.e., requirements and hardware/software functions are verified at the lowest level possible where test perceptivity and environmental stress is usually the greatest.

7.2.1 Test Categories

Some of the major test categories and their objectives are described in this section. Note that this is not an exhaustive list and that, for the most part, there is no specific naming convention for tests. As such test names listed here may have other equivalent names elsewhere. Sometimes tests are referred to the specialty engineering function or requirement that are verified or validated. For example, environmental or safety tests could be performed on units, subsystem, or the entire system to show compliance.

7.2.1.1 Unit test

Unit test is performed on the smallest piece of testable hardware or software. Each unit is tested separately before integrating them into modules to test the interfaces between modules. Unit testing has proven its value in that a large percentage of defects are identified during its use. One of the most valuable benefits of unit tests is that they give confidence that components work in isolation as expected.

7.2.1.2 Development test

Tests conducted on representative articles to characterize engineering parameters, gather data and validate the design approach. Development tests may be performed on a dedicated engineering model to demonstrate the design adequacy and quality of workmanship.

7.2.1.3 Integration test

Tested units are combined into subsystems, and subsystems are joined into the system and tested at each stage in a hierarchical bottom-up approach. Units and their interfaces are tested for faults against each other that compose a subsystem. Similarly, subsystem tests aid in fault detection and isolation by testing specific functions within a subsystem to determine if they perform as required. In many cases, these tests check the interface between the subsystem under test and associated subsystems. Systems are also tested for their interfaces against other systems. For example, SV is integrated with the LV for launch. Also, if the system under test is part of a SoS with specified CONOPS, all of its interfaces involved in interoperability are tested for accurate behavior.

7.2.1.4 Regression test

Regression tests are performed on existing devices or, especially software, when they are somehow modified for improvement. The intent is to ensure that the change did not introduce new faults within or defects in performance of other parts of the system through unintended changes to the device interface under test. This is usually accomplished by running all existing device tests and possibly new tests designed especially to probe possible faults in the changes or improvements.

7.2.1.5 System test

System testing of hardware or software is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. It is performed on the entire system to show that it functions as expected and meets the stakeholder capability needs as documented in the specifications. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of hardware or logic.

7.2.1.6 Built-in test and prognostics and health management

Built-in test (BIT) is a mechanism that permits a device to test itself. These tests, typically performed at power-up or during normal operation, provide a level of confidence that a module or component is operating correctly. These tests are used to (i) increase reliability, (ii) reduce repair cycle time, and (iii) cost of testing during manufacture. While built-in testing is typically passive, prognostics and health management philosophy offers an active approach to on-system testing to monitor, predict, and avoid failure before it happens. Extensive failure and fault detection data is used to develop model scenarios and prediction algorithms which are then incorporated into the system diagnostic design process. Systems when operating collect in situ data for prognostics and health management and possible unattended remedial action for greater dependability.

7.2.1.7 Qualification test

Tests conducted to demonstrate satisfaction of design requirements including margin and product robustness for designs that have no demonstrated history. A full qualification validates the planned acceptance program, in-process stress screens, and retest environmental stresses resulting from failure and rework. Qualification hardware that is selected for use as flight hardware is evaluated and refurbished to show that the integrity of the hardware is preserved and that adequate margin remains to survive the rigors of launch and provide useful life on orbit.

7.2.1.8 Proto-Flight Test

Proto-flight testing is performed on flight hardware for which there is no previous qualification heritage. Proto-flight testing accomplishes the combined purposes of design qualification and flight acceptance. For other than dynamic testing, proto-flight test levels and durations are identical to the qualification levels and durations. Proto-flight dynamics test levels are identical to qualification test levels but for flight acceptance durations.

7.2.1.9 Acceptance test

Vehicle, subsystem, and unit tests conducted to demonstrate that flight hardware is free of workmanship defects, meets specified performance requirements, and is acceptable for delivery.

7.2.1.10 Operational test

Operational test are validation tests performed following launch to verify specification performance, interface compatibility, calibration, and the ability to meet mission requirements.

7.3 Design for testing

Design for testing (DFT) philosophy, especially for electronic circuitry, is an essential ingredient for good V&V. DFT is part of design specifically employed to ensure that a device, part, or unit is testable. In general DFT is achieved by employing extra hardware or software. Specific on-device built-in fault detection, prognostics, and health testing is incorporated at strategic locations to predict or identify failure or to retrieve and observe meaningful test data. Benefits of DFT include

- enhanced ability to view and correct device faults, often leading to unattended operation
- a reduction in time and cost to develop, build, and perform tests,
- support for a test hierarchy from parts or chips or software code snippets to complete systems,
- improved opportunity for concurrent engineering, and
- allows quick convergence of design to exhibit repeatable and robust behavior
- a reduced lifecycle cost.

The DFT techniques include:

- Ad-hoc methods that rely on good design practice to include test points, initialization, partitioning, and redundancy
- Structured methods involve addition of extra logic and signals dedicated to test. The device can possibly have two modes of operation – normal and test. For example, this includes built-in self-tests as employed by the GPS receivers.

7.4 Documenting test and evaluation

Test and evaluation is integrated with the rest of the SE effort. Documented decisions for test and evaluation are recorded in the Test and Evaluation Master Plan (TEMP). The testing program in the TEMP is consistent with the overall SE program management plan. The test program in the TEMP provides the technical performance measurements required for review, audits, and risk management. Other documents integrated with the TEMP include:

- Configuration management plan.
- Functional analysis documents.
- Requirements Allocation Sheets (RASs) and Design Constraint Sheets (DCSs).
- Test Requirements sheets.
- Specifications.

Test and evaluation is not limited to the primary product. The facilities and support system need to be considered by risk reduction efforts as well. For example, supportability can and is measured to optimize lifecycle cost.

7.5 Testing for space and launch environments

As introduced in section 1.6 unforgiving environment for both launch and in space where SVs operate demands careful testing. The following subsections describe some of the rigorous testing required for launch and space operations qualification. For a more detailed and thorough discussion of testing for space see SMC-S-016, "Test Requirements for Launch, Upper-Stage, and Space Vehicles," 2008.

7.5.1 Vibration testing

Operating engines during powered flight and high acceleration lift through the atmosphere induce vibration that the SV must survive to reach orbit intact. Sinusoidal and other vibration data for specific launch vehicles like Delta IV or Atlas V is available from their respective LV user's guides along with other relevant test details including amplitude, frequency, and duration. Typically, maximum flight level sinusoidal environment is maintained for acceptance testing, but is increased by 3dB for payload qualification and proto-flight testing.

7.5.2 Acoustic testing

The maximum acoustic environment by the SV occurs during lift off and transonic flight for a duration of about 10 seconds. Qualification and proto-flight testing is parformed at 3dB higher than the expected level, lasting 120 and 60 seconds, respectively. Acceptance testing is performed at the expected acoustic environment level for a duration of 60 seconds.

7.5.3 Shock testing

Pyrotechnic shock levels are hard to simulate in laboratory. Most direct method is to deploy chosen LV's spacecraft separation system in flight configuration with functional ordnance devices. Payload and proto-flight qualification testing is performed by activating the system twice, and acceptance testing is similarly performed by activating the SV separation system once.

7.5.4 Thermal testing

SVs experience a wide range of temperatures from liftoff to eventual orbit where they function for 15 or more years of their useful life. Thermal cycling as they spin and move in and out of sunlight generates stress and fatigue in materials leading to structural damage, especially if the coefficients of expansion are not well-matched. Electronic parts are especially prone to failure under extreme thermal cycling that results in cracking, delamination, and bond and solder-joint failure. Thermal tests are used assure hardware dependability in its expected thermal environment in space. Thermal test specifications depend on test objective, but are typically used at the unit or device level to screen for defective parts. Parts, subsystems, and systems where appropriate are tested under operating conditions at extreme hot and cold temperatures and after many hot-cold cycles, and their performance is monitored and compared to the required specifications.

Thermal tests are typically conducted in a chamber that can also be used to simulate the vacuum of space. Like real space, this helps eliminate convective averaging of temperatures under test.

7.5.5 Vacuum testing

SVs operate in the vacuum of space. Under normal temperature and pressure on earth, materials absorb or adsorb gases. Outgassing in vacuum or space can change material properties, often making them brittle that makes it harder for parts and assemblies to survive thermal cycling. Tests are performed in Thermal-vacuum chambers to characterize outgassing to observe failure modes under thermal cycling, screen defective parts and assemblies, and collect data on dependability.

7.5.6 EMI/EMC

Ensure product does not generate EM energy that may interfere with other spacecraft components or with launch vehicle or range safety signals. Verify that the product is not susceptible to the range and/or launch EM environment. Detect emitted signals, especially at the harmonics of the clock frequencies. Check for normal operation while injecting signals or power losses.

7.6 Independent V&V

Independent V&V (IV&V), especially for high risk programs, is advisable. A set of contractor(s) unrelated to the developer are invited in to assess the program and its health. IV&V effort, like the V&V, typically start early in the program. Independent sets of expert eyes, not too close to the day-to-day functioning of the program are more likely to find errors and omissions. IV&V organization typically has:

- managerial independence with a separate responsibility from the developing contractor and can choose where to focus V&V effort,
- financial independence with specific funding and schedule, and
- technical independence in choice of personnel, tools, and techniques to avoid bias.

The philosophy, as borne out by experience in the commissioning of complex systems, is that errors found earlier are cheaper to fix and verify. It leads to clearer specifications that represent the customer needs more accurately for easier validation. It also forces the developer to apply best practices and processes in product development thus reducing risk of failure.

7.7 Reducing integration and test time

In this era of cost competition and short schedules, reducing integration and test time has major benefits. Paying attention to what requirements must be tested, and accommodating the need for future testing to the fullest practical extent will lower costs and shorten schedules. Equally important is ascertaining the level at which you will verify requirements. Attention here will avoid the use of convoluted testing arrangements or the need to tear down the product to make certain measurements. Considerations for reducing integration and test time include:

- Clear identification of the system level for each requirement to be evaluated.
- Interface definition.
- Peer walkthroughs.
- Models and simulations.
- Robust design to component parameter variation, manufacturing process
- Robust inputs, targets outputs.
- Commonality, standardization.

- Simplicity.
- Testability.
- Reliability.
- Maintainability.
- Test equipment and facilities availability.
- Independence of components.
- Hardware emulator for untested software; tested software for untested hardware.
- Modular, bottom-up testing.
- Understanding of the critical path.
- Test plan and test procedures ready.

8 Afterword

This booklet is not intended to provide a comprehensive view on SE. It does provide background and reasonable starting point for those who are encountering SE for the first time, or a reprise of the current thinking for those who have been away from it for a while. The booklet also helps in forming the right questions to pursue additional knowledge. It is expected that for specific SE implementation problems and needs additional information will be sought from other authoritative books, websites for professional organizations like the INCOSE, and practicing professionals. The suggested additional readings in the bibliography is a good place to start gathering more information.

Appendix A–Glossary

(Sources used in the preparation are in parentheses following each definition)

Accomplishment: See “significant accomplishment.”

Accomplishment criteria: See “significant accomplishment criteria.”

Acquisition program: Within the DoD, an approved and funded activity that defines the skill and manpower levels for the people, develops and produces the products, and develops the processes that make up a system.

Affordable: An acquisition program for which the life-cycle cost of is in consonance with the long-range investment and force structure plans of the Department of Defense or individual DoD Components.

Allocated baseline: The initially documented, validated, and approved design-to requirements and all changes thereto approved in accordance with the contract. The allocated baseline includes (a) the physical hierarchy, (b) the design-to requirements for each product in the hierarchy, and (c) separable documentation identifying all design-to requirements for each component or computer software unit and each integrated grouping of components.

Allocation: (1) All or part of a requirement for a higher level system element that has been designated to be satisfied by a lower tier element or item. (2) The process of decomposing the requirements for a system among the elements or items of the system. (3) The results of (2).

Analysis: (1) The performance and assessment of calculations (including modeling and simulation) to evaluate requirements or design approaches or compare alternatives. (2) The verification method of determining performance (a) by examination of the baseline, (b) by performing calculations based on the baseline and assessing the results, (c) by extrapolating or interpolating empirical data of collected using physical items prepared according to the baseline, or (d) by a combination of all of the above.

Analysis of Alternatives (AoA): An important step usually required early in the work leading up to an acquisition program. The evaluation of the operational effectiveness, operational suitability and estimated costs of alternative systems to meet a mission capability. The analysis assesses the advantages and disadvantages of alternatives being considered to satisfy capabilities, including the sensitivity of each alternative to possible changes in key assumptions or variables.

Analysis of Materiel Approaches (AMA): Part of the JCIDS analysis process. When the analysis of doctrine, organization, training, materiel, leadership and education, personnel and facilities (DOTMLPF) capabilities and deficiencies indicates that a materiel approach may be needed, the AMA will determine the best materiel approach or combination of approaches to provide the desired capability or capabilities, especially for joint capability or capabilities. It will not usually consider which specific “systems” or “system components” are the best. For example, the AMA may compare the capability provided by a space platform with that by provided by an unmanned aerial vehicle (UAV) but will not usually assess the best alternatives among space platforms or UAVs. That best specific system will usually emerge from an analysis of alternatives (AoA) after the ICD is approved and be the basis for the CDD.

Approved: The formal acceptance of an item, data, or document by the management level required by the contract or contract plan. If the level is the Government, the Government has notified the Contractor that it is acceptable through a contractual letter.

Architecture: See system architecture.

Article: An individual copy of item.

As-built configuration: A production-representative article built or fabricated in accordance with the design release or product configuration baseline.

Attribute: A quality, property, or characteristic of results of the SE process.

Audit: An independent examination of the results of work to assess compliance with a specification, standard, or contract, or other criteria.

Balance: The act of assessing and comparing capabilities to be provided, cost, schedule, risk, and evolvability for alternative requirements, requirements allocations, functional architectures, and/or designs to include identifying the capabilities or constraints that drive or otherwise cause high sensitivity to cost, schedule, or risk.

Balanced: A set of system requirements, requirements allocations, functional architecture, and/or design for which the capabilities to be provided, cost, schedule, risk, and evolvability have been assessed and found to be acceptable in the context of the program that is to satisfy the requirements.

Baseline: noun–Document(s) or database(s) that record a set of requirements and/or product solutions and that can be changed only by formal, documented procedures.

Brass-board: A highly functional prototype that demonstrates the functionality of a particular component without the weight, packaging, power and reliability constraints of the final product.

Build-to requirements: Drawings, manufacturing or assembly instructions, process specifications and instructions and/or any other data required to manufacture an item.

Capability: The ability to execute a specified course of action. It is defined by an operational user and expressed in broad operational terms in the format of an initial capabilities document or a doctrine, organization, training, materiel, leadership and education, personnel and facilities (DOTMLPF) change recommendation. In the case of material proposals, the definition will progressively evolve to materiel performance attributes identified in the CDD and the CPD to guide an acquisition program.

Capability Development Document (CDD): A document that captures the information necessary to develop one or more acquisition programs, normally using an evolutionary acquisition strategy. The CDD outlines an affordable increment of militarily useful, logistically supportable and technically mature capability.

Capability Production Document (CPD): A document that addresses the production elements specific to a single increment of an acquisition program.

Change: A modification of an approved requirement, baseline, or product as documented in a decision data base, specification, or any other configuration management documentation and approved in accordance with the contract.

Change control: The engineering management function of (a) limiting change to a baseline or product to that which has been (i) assessed for impacts to capabilities, cost, schedule, risk, and growth potential and (ii) approved by documented procedures in accordance with the contract and (b) assuring implementation of all changes so assessed and approved to the products of the program.

Change proposal: A proposed change to the currently approved configuration baseline for a configuration item and the documentation by which the change is described, justified, and, if required by the contract, submitted to the Government for approval or disapproval.

Clinger-Cohen Act (CCA): The Clinger-Cohen Act (i.e., the Information Technology [IT] Management Reform Act of 1996 [ITMRA]), which took effect August 8, 1996, abolished the Brooks Act (it repealed Section 111 of the Federal Property and Administrative Services Act of 1949 (40 U.S.C. 759)). The Brooks

Act made the General Services Administration (GSA) the central authority for procurement of automatic data processing (ADP) resources. The Federal Information Resources Management Regulation (FIRMR) was issued to implement the Brooks act and established a process that required Federal agencies to obtain a Delegation of Procurement Authority (DPA) from GSA to acquire ADP, initially, and telecommunications (TC) resources. Passage of the Clinger-Cohen Act is causing a major paradigm shift in the process for acquiring and managing IT. The task of understanding the objectives of Clinger-Cohen and establishing a program or process to manage IT in a Federal agency is a major undertaking. Under Clinger-Cohen, National Security Systems (NSS) is considered part of IT.

Commercial-off-the-shelf (COTS): An item that is available in the commercial marketplace that does not require unique Government modifications or maintenance over its life-cycle to meet the requirements.

C**ompatibility:** The capability of two or more items to exist or function in the same system or environment without mutual interference.

Component: An item that is viewed as a separate entity for purposes of design, manufacturing, software coding, testing, maintenance, contracting, reprocurement, record keeping, or configuration management. A configuration item is a component, but all components are not necessarily configuration items, i.e., they may be controlled by other than formal configuration management procedures. Hardware components may be further divided into additional components; software components may be further divided into additional components and/or software units.

Computer software: The complete set or any item of the set of computer programs or instructions in the physical hierarchy and the associated documentation.

Concept: A rudimentary or unfinished design, used for preliminary assessments of system effectiveness, cost, schedule, or risk.

Configuration: The functional and physical characteristics of an item as documented in a baseline and ultimately achieved in a product or process.

Configuration baseline: The configuration document(s) or database(s) that record the initially approved set of requirements and/or product solutions and all approved changes thereto and that is changed only by formal, documented procedures.

Configuration control: Formal change control for configuration items.

Configuration item: An item that satisfies a documented set of requirements and is designated for separate configuration management to include any item required for logistic support or designated for separate procurement.

Configuration management: For configuration items, (1) the identification and documentation of the configuration, (2) the control of changes to the items or their documentation, (3) configuration status accounting, and (4) the auditing to confirm that conformance to all requirements has been verified.

Configuration status accounting: For configuration items, the recording and reporting of (1) the approved configuration baseline and identification numbers, (2) the status of proposed changes, deviations, and waivers, (3) the implementation status of approved changes, and (4) the configuration of all units of the configuration item owned by the Government.

Constraint: A technical requirement imposed other than directly by the definition of the needed capability. Constraints can be imposed by an interface with another system, by the natural or threat environment, by public law or regulation, by the program budget (also called a cost constraint), or other factors.

Control: The engineering management function of ensuring that plans are having the intended effect and that work is being completed according to the plans. Controlling is one of the basic functions of engineering management -- the others are planning, organizing, staffing, directing, and monitoring.

Cost engineering: The art of analyzing and estimating the cost of a design solution and relating those costs to the requirements.

Critical Design Review (CDR): (1) During Engineering and Manufacturing Development (EMD) or similar phase, the review by the Contractor and the Government of (1) the status of any changes to the functional baseline and architecture and allocated baseline since they were established, (2) the design baseline for each configuration item including the completeness and compatibility of interfaces between the items and between the items and other systems, facilities, and personnel, (3) the basis for each element in the design baseline in terms requirements and objective, comprehensive, quantitative design trades, (4) the balance between performance, cost, schedule, and risk for each element in the selected design baseline, (5) the two-way traceability from the source of the functional baseline to the design baseline and back, and (6) the verification that the design baseline can meet the contract requirements. The data available for CDR should document or demonstrate these six items and reside in the decision data base. (2) During the Program Definition and Risk Reduction (DEM/VAL) or similar phase, a review conducted on each prototype (1) to evaluate the progress, technical adequacy, and risk resolution of the detailed design and (2) to determine its alignment with the evolving functional architecture and allocated baseline including compatibility of the physical and functional interfaces among the item and other items, systems, facilities, and personnel.

Data accession/internal data list: An evolving list, prepared and maintained by the Contractor, of data acquired or prepared under the contract and accessible by the Government either by access to a management information system or by PCO direction.

Decision database: The linked and readily retrievable collection of data (including inputs and intermediate and final results) that provide the audit trail of decisions and their rationale from initially stated needs and requirements, the system threat assessment, other program documents, and DoD policy, AF practice, and public law to the current description of the system requirements and the products, processes, facilities, and personnel requirements that collectively satisfy the requirements. It includes, as they evolve, (1) the functional baseline, the functional architecture, the physical hierarchy, and the allocated, design, and product baselines; (2) life-cycle verification, manufacturing, support, deployment, training, operations, and disposal data, procedures, and plans (including but not limited to test plans and procedures, drawings, manufacturing instructions, logistics support plans, common [Government-inventory] support equipment requirements, spares requirements, training programs [or training program requirements for training programs not developed under the contract], technical manuals, and required Government personnel skill and manpower levels applicable to both OT&E and the operations phase); (3) the embedded software; (4) remaining risks and corresponding risk monitoring (including TPMs and metrics) and mitigation steps; (5) cost estimates and their bases; (6) data, models, and analytic techniques used to verify that an evolving solution can meet its requirements; (7) the verification results that verify compliance of designs or delivered products with the contract requirements; (8) the approval authority and rationale for any changes to the data; and (9) any other decision support data developed under the contract linked to its basis in the rest of the data base. It provides for the efficient traceability through the architectures, baselines, and the physical hierarchy from any element up to the Government sources of the functional baseline or down to the lowest elements of the allocated, design, and product baselines; from any element to the corresponding requirement reference; from any requirement to the corresponding verification method and verification plans, procedures, and data; from any component in the physical hierarchy to its design-to and build-to requirements, product description, and supportability data; and from any element to its change history.

Demonstration: The verification method of determining performance by exercising or operating the item in which instrumentation or special test equipment is not required beyond that inherent to the item and all data required for verification is obtained by observing operation of the item.

Deployment function: Tasks to be performed to take the elements of a system or system upgrade from the completion of development, training, manufacturing, and verification to a state of operational readiness.

Derating: re-specification of electronic piece parts for electrical stresses and radiation environments encountered.

Derived requirements: Requirements not explicitly stated in the operational requirements and which are inferred from the nature of the proposed solution, the environment, policy, law, best engineering practice, or some combination of the above.

Design: verb: Architecting and selecting products (including processes) and corresponding personnel manpower, skill levels, and specialized training that satisfy all requirements and describing them so that the products can be manufactured or coded, verified, deployed, operated, supported, and disposed of and so that the personnel can be selected and trained. Noun: The result of designing.

Design baseline, design release baseline: The initially documented, validated, and approved design for a product and all subsequent changes thereto approved in accordance with the contract. Includes the documented requirements for material ordering ("buy-to" requirements), hardware fabrication and manufacturing process setup and operation for developmental hardware ("build-to" requirements), software coding ("code-to" requirements), integration ("integrate-to" requirements), verification, training, deployment, operations, support, and disposal ("verify-to, train-to, deploy-to, operate-to, support-to, and dispose-to" requirements) and personnel skill and manpower levels that collectively satisfy the requirements baseline. The design release baseline usually includes separable documentation for each hardware and software component. For programs that will transition to production, the design baseline forms an initial or preliminary product configuration baseline. The complete product configuration baseline will usually be formalized near the end of development or early in production. If the Event Critical Design Review (CDR) or the equivalent is held, the design release baseline is usually formalized as part of the Event close-out.

Design constraints: Requirements that form boundaries within which other requirements must be allocated and items must be designed. The constraints may be externally imposed or result from decisions internal to the program or contract. Design constraints include interface, environmental, physical mass and dimensional, reliability, maintainability, human factors, logistics support, personnel resource (skill levels and manpower) and training, standardization, design and construction practices, and fiscal (cost) requirements.

Design to Cost (DTC): noun: An acquisition management technique in which cost design constraints are derived and allocated to the items to be designed. adj.: Derived by applying the DTC technique.

Development function: Tasks to be performed to take a system or system upgrades from the statement of the operational requirement to readiness for verification, manufacturing, training, deployment, operations, support, and disposal.

Developmental Test & Evaluation (DT&E): Test and evaluation activities to (1) support technology selection, requirements analysis and allocation, and design and (2) verify compliance with the contract requirements.

Deviation: A specific written authorization, granted prior to the manufacture of an item, to depart from one or more particular requirements of an items approved configuration baseline for a specific number of units or a specified period of time.

Disposal function: Tasks to be performed to ensure that the disposition of products and by-products that are no longer needed or no longer useful complies with applicable security classification guidance and environmental laws and regulations. The function addresses the short and long term impact to the environment and health hazards to humans and animals as well as recycling, material recovery, salvage for re-utilization, demilitarization, and disposal of by-products all other functions, i.e., across the life cycle.

Documented: Recorded on paper or in electronic or other media in accordance with the contract.

Effectiveness: See "system effectiveness."

Eight primary system functions: The essential tasks that must be accomplished so that a system will satisfy the operational needs, DoD policy, and the law over the life cycle. Any defense acquisition program must complete eight primary functions: development, manufacturing, verification, deployment, operations, support, training, and disposal.

Elastomeric: having elastic properties

Element: In a system, baseline, or architecture, any product, any representation of a product, any requirement or allocation of a requirement, or any logical or abstract representation or decomposition thereof (such as a function, sub-function, object, or data structure).

Environment: The natural and induced conditions experienced by a system including its people and products (including its processes) during operational use, stand-by, maintenance, transportation, and storage. The natural conditions include space (exo-atmospheric), atmospheric (weather, climate), ocean, terrain, and vegetation. Induced conditions includes manufacturing (process conditions, clean room, storage), test, transportation, storage, normal operations (thermal, shock, vibration, electromagnetic, the range of power inputs), maintenance, combat (dust, smoke, nuclear-chemical-biological), and the threat (existing and potential threat systems to include electronic warfare and communications interception).

Environmental constraints or requirements: The expected worst case impact of the environment on the system or item as well as the system or items allowed impact on the environment.

Equipment: Hardware, hardware and software, or an assembly of hardware or hardware and software.

Event: A point in a program or contract defined by significant accomplishments and accomplishment criteria (or metrics) in the IMP. The goal for the calendar date to complete an event is documented in the IMS.

Evolutionary Acquisition: Is an acquisition strategy that defines, develops, produces or acquires, and fields an initial hardware or software increment (or block) of operational capability. It is based on technologies demonstrated in relevant environments, time-phased requirements, and demonstrated manufacturing or software deployment capabilities. These capabilities can be provided in a shorter period of time, followed by subsequent increments of capability over time that accommodate improved technology and allow for full and adaptable systems over time.

Evolutionary Development: There are generally two types of evolutionary development, exploratory development and throw-away prototyping. Exploratory development starts with requirements that are well defined and add new features when customers propose new requirements. Throw-away prototyping establishes the objective of understanding a customer's requirements (i.e., they often don't know what they want, hence poor requirements to start) and uses such means as prototyping to focus on poorly understood requirements, redefining requirements as you progress.

External interface: A design constraint imposed on a system by another system or facility.

Firewire: A specific computer device communication protocol developed by Apple. Also, known at IEEE 1394.

Formal: An act that follows a documented procedure and that is approved by the signature of an authorized individual recorded in a readily retrieved archive.

Function: A task to be performed to achieve a required outcome or satisfy an operational need.

Functional analysis and allocation: The determination of the top level functions that are needed to accomplish the eight primary system functions over the life of the system, their relationship, and their decomposition to sub-functions to the point that each sub-function or set of sub-functions can be related to one and only one physical element in the allocated baseline, the allocation of the top-level requirements and constraints in the requirements baseline to determine how well each function and sub-function must be performed, and the capture of the aggregate in a functional architecture.

Functional architecture: The product of functional analysis and allocation; including hierarchical arrangement of functions, their decomposition into sub functions, the associated time-lines, and the allocation of the requirements and constraints in the requirements baseline to the functions and sub-functions. Note: A specific form of a logical solution representation as used in ANSI/EIA-632-1998.

Functional baseline: See requirements baseline.

Functional Configuration Audit (FCA): For each configuration item, the formal examination of its functional characteristics to verify that it has achieved the requirements in its allocated baseline. For a system, the formal examination of its functional characteristics to verify that it has achieved the requirements in the functional baseline.

Functional requirement: A task that must be accomplished to provide a needed operational capability (or satisfy an operational need or requirement). The top-level functional requirements are the eight primary system functions stated and linked as they apply to the operational need or requirements.

Hardware: Items made of a material substance but excluding computer software and technical data packages.

Hirel: high reliability

Initial Capabilities Document (ICD): Documents the need for a materiel approach to a specific capability gap derived from an initial analysis of materiel approaches executed by the operational user and, as required, an independent analysis of materiel alternatives. It defines the capability gap in terms of the functional area, the relevant range of military operations, desired effects and time. The ICD summarizes the results of the DOTMLPF analysis and describes why non-materiel changes alone have been judged inadequate in fully providing the capability.

Initial Operational Test and Evaluation (IOT&E): See "Operational Test and Evaluation (OT&E)."

Inspection: The verification method of determining performance by examining (1) engineering documentation produced during development or modification or (2) the item itself using visual means or simple measurements not requiring precision measurement equipment.

Integrated Logistics Support (ILS): A disciplined, unified, and iterative approach to the management and technical activities necessary to (1) integrate support considerations into system and component design; (2) develop support requirements that are consistently related to readiness objectives, to design, and to each other; (3) acquire the required support; and (4) provide the required support during the operational phase at minimum cost.

Integrated Master Plan (IMP): A description, usually contractual, of the applicable documents, significant accomplishments, accomplishment criteria, events, and critical processes necessary to satisfy all contract requirements. The completion of each significant accomplishment is determined by measurable accomplishment criteria. The significant accomplishments have a logical relationship to each other and, in subsets, lead up to events. Each event is, in turn, complete when the significant accomplishments leading up to it are complete. The critical processes are described by narratives that include Objectives, Governing Documentation, and an Approach. The IMP includes an indexing scheme (sometimes called a single numbering system) that links each significant accomplishment to the associated CWBS element, event, significant accomplishment criteria, and tasks presented in the Integrated Master Schedule (IMS). The data in the IMP defines the necessary accomplishments for each event both for each IPT and for the contract as a whole. See also Integrated Task and Management Plan (ITAMP).

Integrated Master Schedule (IMS): The schedule showing the time relationship between significant accomplishments, events, and the detailed tasks (or work packages) required to complete the contract. The IMS uses (and extends if necessary) the same indexing (or single numbering system) as used in the Integrated Master Plan (IMP).

Integrated Product and Process Development (IPPD): A management technique that simultaneously integrates all essential acquisition activities through the use of multi-disciplinary Integrated Product or Process Teams (IPTs).

Integrated Process Team (IPT): Team composed of specialists from all appropriate functional disciplines working together (1) to develop and operate processes that affordably meet all program requirements and (2) to enable decision makers to make the right decisions at the right time. For Acquisition Category I and II (ACAT I and II) space programs, the IPT is chaired by a senior individual in the office of the Air Force Mission Area Director for Space (SAF/AQS).

Integrated Product Team (IPT): Team composed of specialists from all applicable functional disciplines working together (1) to deliver products and processes that affordably meet all requirements at acceptable risk and (2) to enable decision makers to make the right decisions at the right time by timely achievement of the significant accomplishments in the Integrated Master Plan (IMP).

Integrated Task and Management Plan (ITAMP): A single document that combines and fulfills the purposes of the Statement of Work (SOW) and the Integrated Master Plan (IMP). The Task Section of the ITAMP replaces the SOW and the other sections are identical to the IMP.

Integration: The merger or combining of two or more parts, computer software units, components, or other items into a still higher level item to ensure that the functional requirements and design constraints for the higher level item are satisfied.

Interface: The boundary, often conceptual, between two or more functions, systems, or items or between a system and a facility at which interface requirements are set.

Interface constraint: See interface requirement.

Interface control: The process of identifying, documenting, and controlling all interface requirements on a system or the elements of a system.

Interface Control Document (ICD), Interface Control Drawing: Drawing or other documentation that depicts interface designs or elements of interface designs that satisfy interface requirements.

Interface Control Working Group (ICWG): A group with representation from all sides of an interface that seeks agreement on mutually compatible interface requirements and controls the documentation of the resulting interface agreements. ICWGs that address external interfaces will usually be chaired by the Government. ICWGs that address internal interfaces, if separate, may be chaired by the Contractor.

Interface requirement: The functional and physical design constraints imposed on each other by two or more functions, items, or systems or between a system and a facility. Functional interfaces include signal, electrical, electromagnetic, and software. Physical interfaces include keep-out volumes and mating surfaces and connections.

Interface requirements specification (IRS), interface specification: A repository for interface requirements that details the functional and physical connection between systems or system elements or between systems and facilities.

Internal interface: The functional and physical design constraints imposed on an item resulting from the designs selected for other items in the same system. (Also, see interface requirement and external interface.)

Interoperability: The ability of systems, units, or forces to provide services to or accept services from other systems, units, or forces and to use the services so exchanged to operate effectively together.

Item: Any product (where products include processes and facilities).

Life cycle: The scope of a system or upgrade evolution beginning with the determination of a mission need or identification of a system deficiency through all subsequent phases through disposal of the system.

Life Cycle Cost (LCC): The total cost to the Government of acquisition and ownership of the system over its useful life. It includes the cost of development, production, operations & support, and disposal.

Logistics Support Analysis (LSA): Engineering efforts, as part of the SE process, to assist in: causing support considerations to influence design; defining support requirements that are related optimally to design and to each other; acquiring the required support; and providing the required support during the operational phase at minimum cost.

Manufacturing function: Tasks to be performed to convert materials and parts into a product ready for verification, training, and/or deployment.

Metric: A measure used to indicate progress or achievement.

Microradian: An angular measure that is one millionth of a radian. Approximately .000057 degrees.

Milestone: (1) A point in a program or contract at which some team member or leader is held accountable and at which progress toward completion of the program or contract is measured. Also, see event. (2) Major decision points that separate the phases of defense acquisition programs. Phases include, for example, engineering and manufacturing development and full-rate production.

Milestone Decision Authority (MDA): The individual designated in accordance with DoD 5000.02 to approve entry of a defense acquisition program into the next phase.

Mission Need Statement (MNS): A statement of the need for a material solution to perform an assigned mission or to correct a deficiency in existing capability to perform the mission.

Modification: The act of changing a system or component after delivery to improve some characteristic, to adapt it to function in a changed environment, or to respond to a change in the law. Also, see upgrade.

National Security System (NSS): Any telecommunications or information system operated by the U.S. Government, the function, operation, or use of which:

- Involves intelligence activities;
- Involves cryptologic activities related to national security;
- Involves command and control of military forces;
- Involves equipment that is an integral part of a weapon or weapons system; or,
- Subject to the limitation below, is critical to the direct fulfillment of military or intelligence missions. This does not include a system that is to be used for routine administrative and

business applications (including payroll, finance, logistics, and personnel management applications).

Non-Developmental Item (NDI): Any item that is (1) available in the commercial marketplace or (2) previously developed and in use by a department or agency of the United States, a State or local Government, or a foreign Government with which the United States has a mutual defense cooperation agreement and that does not require unique upgrades or maintenance over its life-cycle to meet the current requirements. In some cases NDI may be extended to include items that (a) have been developed but are not yet available in the commercial marketplace or in use by a Government entity or (b) require only minor modification or upgrade. In other cases, items meeting these latter criteria are termed Near-NDI or N-NDI.

Objectives: Operationally significant desired levels of performance or functionality above the requirement that are goals for the program or contract but not a requirement.

Operational effectiveness: The overall degree of mission accomplishment of a system when used by representative personnel in the environment planned or expected (e.g., natural, electronic, threat etc.) for operational employment of the system considering organization, doctrine, tactics, survivability, vulnerability, and threat (including countermeasures, initial nuclear weapons effects, nuclear, biological, and chemical contamination (NBCC) threats).

Operational requirements: Requirements generated by the Operator/Users, normally in terms of system capabilities or characteristics required to accomplish mission tasks, and documented in a Mission Needs Statement (MNS) that evolves into an Operational Requirements Document (ORD) and associated Requirements Correlation Matrix (RCM).

Operational Requirements Document (ORD): Usually prepared during Phase 0, Concept Exploration, the ORD will be based on the most promising alternative determined during the Phase 0 studies. The ORD documents how the system will be operated, deployed, employed, and supported by describing system-specific characteristics, capabilities, and other related operational variables. The ORD will be updated for Milestones II and III. The CSAF approves all Air Force and Air Force-led ORDs.

Operational Test & Evaluation (OT&E): Independent test and evaluation to determine the effectiveness and suitability of the weapons, equipment, or munitions for use in combat by typical military users; and the evaluation of the results of such tests. Can be either Initial (IOT&E) or Follow-on (FOT&E). IOT&E is conducted on production or production representative articles, to support a decision to proceed such as beyond low-rate initial production. It is conducted to provide a valid estimate of expected system operational effectiveness and operational suitability. FOT&E is conducted during and after the production period to refine the estimates made during IOT&E, to evaluate changes, and to reevaluate the system to ensure that it continues to meet operational needs and retains its effectiveness in a new environment or against a new threat.

Operations function: Tasks to be performed subsequent to verification and deployment to accomplish defined missions in either the expected peacetime or wartime environments excluding training, support, and disposal.

Performance: A measure of how well a system or item functions in the expected environments.

Performance requirement: The extent to which a mission or function must be executed, i.e., a functional requirement that is stated in terms of quantity or quality such as range, coverage, timeliness, or readiness.

Physical architecture: The physical hierarchy and the functional requirements and design constraints for each element in the hierarchy. It can be viewed as an intermediate step between the functional architecture and the physical hierarchy, on the one hand, and the allocated baseline, on the other hand. It is not directly addressed in this CPAT.

Physical Configuration Audit (PCA): For each configuration item (CI), the formal comparison of a production-representative article with its design baseline to establish or verify the product baseline. For the system, the formal comparison of a production-representative system with its functional and design baseline as well as any processes that apply at the system level and the formal examination to confirm that the PCA was completed for each CI, that the decision data base represents the system, that deficiencies discovered during testing (DT&E and IOT&E) have been resolved and changes approved, and that all approved changes have been implemented.

Physical hierarchy, product physical hierarchy: The hierarchical arrangement of products, processes, personnel skill levels, and manpower levels that satisfy the functional baseline. The top entry in the hierarchy is the system. The hierarchy extends to include all components and computer software units necessary to satisfy the functional baseline whether deliverable or not. It includes the prime operational hardware and software, Contractor-supplied support equipment, Government-inventory support equipment, technical manuals, training programs for both Government and Contractor personnel, Government personnel skill and manpower levels, spare parts requirements, and factory support equipment and tooling which collectively result in the system that satisfies the functional baseline.

Physical requirement: A physical characteristic, attribute, or distinguishing feature that a system or item must possess.

Plan: Documented approach, resources, and schedule necessary to complete a task.

Planned value: The predicted value of a technical parameter at the planned time of measurement based on the planned profile.

POH Primer: Project Officers Handbook Primer. A web based application that has primer material to aid a project officer in doing his/her job.

Preliminary Design Review (PDR): During Engineering and Manufacturing Development (EMD), the review by the Contractor and the Government of (1) any changes to the functional baseline since it was established, (2) the functional architecture, (3) the physical hierarchy, (4) the allocated baseline for each configuration item including the completeness and compatibility of interfaces between the items and between the items and other systems, facilities, and personnel, (5) the basis and the balance between performance, cost, schedule, and risk for each element in the architectures and each requirement in the baseline, (6) the two-way traceability from the source of the functional baseline to the allocated baseline and back, and (7) the verification that the allocated baseline can meet the system requirements. The primary PDR data is the Decision Data Base documenting or demonstrating these seven items.

During the Program Definition and Risk Reduction (DEM/VAL) or similar phase, a review conducted on each prototype to evaluate the progress, technical adequacy, and risk resolution of the selected design approach; to determine its alignment with the evolving functional baseline and architecture and allocated baseline including compatibility of the physical and functional interfaces among the item and other items, facilities, and personnel.

Primary functions, primary system functions: See the entry, “eight primary system functions.”

Procedure: A documented description of a sequence of actions to be taken to perform a given task.

Process: A set of steps or activities that bring about a result and the criteria for progressing from step to step or activity to activity.

Product: What is delivered to the customer (e.g., hardware, software, test reports, RFPs, data...), as well as processes (e.g., system engineering, design, manufacturing, test, logistics, acquisition security...) which make the product possible.

Product baseline: Build-to requirements for each physical element to be manufactured; software code for each software element that has been separately designed or tested; and buy-to requirements for each other physical element, part, or material to be procured from a subcontractor or vendor.

Product baseline completion: For each configuration item (CI), the contract status in which a production-representative article and any associated processes have been formally demonstrated to satisfy the corresponding design baseline to establish or verify the product baseline for the CI. For the system, the contract status in which (1) a production-representative system and any processes that apply a the system level have been formally demonstrated to satisfy the system functional and design baseline, (2) it has been formally confirmed that (a) the Product Baseline is complete for each CI, (b) that the decision data base represents the system, (c) that deficiencies discovered during test and evaluation (DT&E and IOT&E) have been resolved and changes approved, and (d) that all approved changes have been implemented.

Product physical hierarchy: See physical hierarchy in this Annex.

Program technical requirements and constraints: Verifiable requirements and objectives restated or derived by the acquisition community from the program operational requirements, the program threat assessment, applicable DoD and DoD-Component practices and policies, and program decisions to achieve all program requirements and objectives. Technical requirements include all program functional and performance requirements, design constraints, and, ultimately, personnel tasks, numbers and skills of personnel, quantities of equipment, spares, repair parts, and consumables. Government program technical requirements are usually initially documented in a Systems Requirements Document (SRD) or similar record and evolved by the Government or the prime Contractor into the System Specification. Technical requirements for the elements of the system are allocated from the Government program technical requirements to the components of the system and documented consistent with the management and contracting structure and support plans.

Requirements: Characteristics, attributes, or distinguishing features that a system or system element must have within a stated environment or set of conditions in order to meet an operational need and comply with applicable policy and practices. Also, see operational requirements and program technical requirements.

Requirements analysis: The determination of the system specific functional and performance requirements and design constraints based on analyses of the operational need, requirements, objectives (or goals), and measures of effectiveness; missions; projected utilization environments; DoD policies and practices; and the law.

Requirements baseline: The initially documented, validated, and approved system-level (top-level) functional and performance requirements and design constraints, their allocation or assignment to the next level, and all changes thereto approved in accordance with the contract. Typically initially approved at the System Design Review (SDR) or similar event. Also called the functional baseline.

Risk: A measure of the uncertainty of attaining a goal, objective, or requirement and the consequences of not attaining it. The uncertainty is the result of one or more undesirable events that could occur during the system life cycle for which insufficient resources and time are programmed to overcome them. The consequences are inability to satisfy the operational military need and exceeding the programmed budget and directed schedule.

Risk management: A documented process for the prospective (looking ahead) and recurring identification of what can go wrong, assigning a level of risk (e.g., High, Moderate, Low) to each risk, and planning and implementing mitigation steps for each commensurate with the level of risk. Also, see the Risk Management CPAT.

Schedule, schedule requirements: Progress characteristics imposed on the completion of program phases, on contract events and deliveries, and operation and support parameters such as time between failures and repair time.

SEMP: The SEMP describes the Contractor's SE process activities to be accomplished during the contract, detailing the contractor's processes and procedures for completing the SE effort.

SEP: The SEP defines the methods by which system requirements, technical staffing, and technical management are to be implemented on a program, addressing the government efforts and the integration of contractor technical efforts. The SEP is the top-level management focal point for the integration of all SSP engineering activities.

Significant accomplishment: A specified step or result that indicates a level of progress toward completing an event and, in turn, meeting the objectives and requirements of the contract.

Simulation: The process of conducting experiments with a model (an abstraction or simplification) of an item and/or part or all of its operating environment for the purpose of assessing its behavior under selected conditions or of evaluating various strategies for its operation within the limits imposed by developmental or operational criteria. Simulation may include the use of analog or digital devices, laboratory models, or "test bed" sites. Simulations are usually programmed for solution on a computer; however, in the broadest sense, military exercises and war games are also simulations.

Software, software product: See computer software.

Specification: A description of the essential technical requirements for items (hardware and software), materials, and processes that includes verification criteria for determining whether the requirements are met.

Specification tree: The hierarchical depiction of all the specifications needed to formally control the development, procurement, manufacture, integration, verification, and/or re-procurement during any part of the life cycle.

Spiral Development: Is an iterative process represented as a spiral rather than a sequence of activities with backtracking for developing a defined set of capabilities within one increment. This process provides the opportunity for interaction between the user, tester, and developer. In this process, the requirements are refined through experimentation and risk management. Risk are explicitly assessed and resolved throughout the process. There is continuous feedback, and the user is provided the best possible capability within the increment. Each increment may include a number of spirals; each loop in the spiral represents a phase in the process. No fixed phases such as specification or design loops in the spiral are chosen depending on what is required.

Subsystem: A grouping of items satisfying a logical group of functions within a system.

Support equipment: All equipment (mobile or fixed) required to support the operation and maintenance of a materiel system. This includes associated multi-use end items, ground-handling and maintenance equipment, tools, meteorology and calibration equipment, test equipment, and automatic test equipment. It includes the acquisition of logistics support for the support and test equipment itself.

Supportability: The degree to which planned logistics support (including system design; test, measurement, and diagnostic equipment; spares and repair parts; technical data; support and facilities; transportation requirements; training; manpower; and software support) allow meeting system availability and wartime usage requirements.

Survivability: The capability of a system to avoid or withstand natural and man-made hostile environments without suffering an abortive impairment of its ability to accomplish its designated mission.

System: An integrated composite of people, products, and processes that satisfy an operational requirement or objective. An acquisition program defines the skill and manpower levels for the people, develops and produces the products, and develops the processes.

System architecture: 1. A structure or organization that shows the elements and their relationship for a set of requirements or a system concept or both. 2. A high-level property or attribute of a system such as openness or interoperability. 3. A standard for achieving 2.

System effectiveness: Quantified or otherwise objective measure(s) (such as communications throughput, surveillance sensitivity, or navigation accuracy) that relates the system concept or design to the system technical functional and performance requirements and constraints.

System element: See element.

Systems engineering: As a process, an interdisciplinary effort to recursively and iteratively (1) support the evolution of, first, the operational need, and then later, the operational requirements and objectives, (2) translate the requirements and objectives into, first, a functional baseline, second, an allocated baseline, third, a design baseline, and, finally, a product baseline, (3) to maintain those baselines over the life cycle of the system, and (4) verify initially that the requirements can be met by the evolving baselines and ultimately that the requirements have been met.

As a team or organizational entity, a group that is directly responsible for certain activities in the process and for facilitating or monitoring others as a staff function to a program or product manager. Note: All of the technical organizations involved in a program or contract have a role in the system engineering process so there is much more than what the system engineering team or office does. Also, see Section 1.1.

System Functional Review (SFR): A review defined in the draft MIL-STD-499B, usually held after the SRR, before the PDR, and instead of the SDR, by the Contractor and the Government to confirm that (1) the planned risk reduction efforts have been completed and the results reflected in the proposed functional baseline and preliminary functional architecture and allocated baseline, (2) the proposed requirements (functional) baseline is accurate and comprehensive (though perhaps with TBDs, TBRs, and TBSs), (3) the preliminary functional architecture and allocated baseline reflect the proposed functional baseline and is balanced with respect to performance, cost, schedule, and risk, (4) the decision data base supports two-way traceability from the source of the functional baseline to the preliminary allocated baseline and from any element to the rationale for that element and shows the rationale and approval authority for all changes, (5) the verification that the evolving allocated baseline can satisfy the functional baseline, (6) the preliminary physical hierarchy, the planned (or approved) PWBS, and the proposed CWBS are all consistent, (7) the life cycle cost for the evolving design is consistent with the program affordability constraints, and (8) the remaining risks have been identified and can be handled in the context of the planned next phase. The primary SFR data is the Decision Data Base documenting or demonstrating these eight items.

System of Systems (SoS): A set or arrangement of interdependent systems that are related or connected to provide a given capability. The loss of any part of the system will degrade the performance or capabilities of the whole. An example of an SoS could be interdependent information systems. While individual systems within the SoS may be developed to satisfy the peculiar needs of a given user group (like a specific Service or agency), the information they share is so important that the loss of a single system may deprive other systems of the data needed to achieve even minimal capabilities.

System Requirements Review (SRR): A review, usually held near the end of the Program Definition and Risk Reduction or similar phase (Phase I), by the Contractor and the Government to confirm that (1) the planned risk reduction efforts are making adequate progress and reflect the technologies envisioned to implement the preferred system concept(s), (2) the operational requirements and objectives have been accurately and comprehensively translated into technical requirements and are reflected in the preliminary functional baseline, (3) the preliminary functional baseline and the plans to complete it account for the eight primary functions and all design constraints on the system design, (4) the preliminary physical hierarchy is consistent with the preliminary functional baseline, (5) life cycle cost projections remain consistent with the program affordability constraints, (6) the decision data base supports two-way traceability from the source of the functional baseline to the functional baseline and from any element to the rationale for that element and shows the rationale and approval authority for all changes, and (8) the significant accomplishments and accomplishment criteria have been planned for the next wave of technical activity on the contract. The primary SRR data is the Decision Data Base documenting or demonstrating these eight items.

System technical requirements: Characteristics, attributes, or distinguishing features, stated in terms of verifiable functional and performance requirements and design constraints, that a system or system element must have within a defined environment or set of conditions, including the threat, in order to provide a needed operational capability and comply with applicable decisions by the milestone decision authority, policy, practices, and law. The system technical requirements are documented in the requirements baseline. Technical requirements for the elements of the system are allocated from the requirements baseline.

System Threat Assessment Report (STAR): Describes the threat to be countered and the projected threat environment. The threat information should reference DIA or Service Technical Intelligence Center approved documents.

System Verification Review (SVR): A review, usually held near the end of Phase II, EMD, by the Contractor and the Government to confirm that (1) the system has been verified to satisfy the functional, allocated, and design baselines including an assessment of the assumptions and methods used in verification by analysis, (2) that the decision data base has been maintained and represents the system, (3) that deficiencies discovered during testing (DT&E and IOT&E) have been resolved and changes approved, (4) that all approved changes have been designed and verified, (5) the life cycle cost projections remain consistent with the program affordability constraints, (6) planning is complete and procedures, resources, and other requisite systems or facilities are available to initiate production, verification, training, deployment, operations, support, and disposal, and (7) the remaining risks have been identified and can be handled in the context of the planned next phase. The primary SFR data is the Decision Data Base documenting or demonstrating these eight items.

Tailoring: The process by which sections, paragraphs, and sentences of specifications, standards, and other requirements or tasking documents are evaluated to determine the extent to which they are applicable to a specific acquisition contract and then modified to balance performance, cost, schedule, and risk.

Task: A unit of work that is sufficiently well defined so that, within the context of related tasks, readiness criteria, completion criteria, cost, and schedule can all be determined.

Team: A group of people that collectively have the necessary knowledge, skills, and resources and are assigned the Responsibility and Authority and are held Accountable (RAA) to perform a task or function.

Technical Data Package (TDP): The evolving data needed for implementing the acquisition strategy, engineering, production, verification, deployment, training, operations, logistics support, and disposal for an item. It defines the configuration and procedures to ensure that the item meets requirements. It consists of performance requirements and the associated development and product specifications, standards, quality assurance provisions, drawings, associated lists, process instructions, packaging details, training program, and technical manuals. The technical data package is a part of the decision data base.

Technical Performance Measure (TPM): A parameter that is related to progress toward meeting the program or contract functional requirements or goals and is assessed periodically and at certain events to estimate the degree to which the final value will meet the anticipated or required level.

Test: The verification method of determining performance by exercising or operating the system or item using instrumentation or special test equipment that is not an integral part of the item being verified. Any analysis of the data recorded in the test and that is needed to verify compliance (such as the application of instrument calibration data) does not require interpretation or interpolation/extrapolation of the test data.

Test plan: Documented approach, resources, and schedule to verify compliance of a system or one of its elements by test.

Test report: Documentation of compliance with the test plan and the compliance or non-compliance of the items under test.

Threat: (1) Countries or groups that are considered to have a potential adverse impact on the national security of the United States. (2) Weapon systems that must be defeated by U.S. systems in battle and the environment in which those systems operate. Note: Threat information, to include the target data base, shall be validated by the Defense Intelligence Agency (DIA) for acquisition programs subject to review by the Defense Acquisition Board (DAB).

Time-line analysis: The analysis of the time sequencing of the elements of the functional architecture and the operation of the elements of a design response to define any resulting time or sequencing requirements.

Traceability: The ability to relate an element of the functional baseline, functional architecture, physical hierarchy, allocated baseline, design baseline, and product baseline (or their representation in the decision data base) to any other element to which it has a master-subordinate (or parent-child) relationship.

Trade-off study: An objective comparison with respect to performance, cost, schedule, risk, and all other reasonable criteria of all realistic alternative requirements; architectures; baselines; or design, verification, manufacturing, deployment, training, operations, support, or disposal approaches.

Training function: Tasks to be performed to achieve and maintain knowledge and skill levels necessary to perform the operations, support, and disposal functions efficiently and effectively over the system life cycle.

Unit: A subdivision of time, fabrication or production quantity, or some other system or program parameter. For software, a subdivision of a component.

Unit Production Cost (UPC): The cost of a single, specified unit (such as first or average) under a defined set of production ground rules (such as schedule and quantity).

Upgrade: A change from previously delivered items because of obsolescence of a part; a change in the military need or threat; an operational, supportability, or training deficiency is identified; the system life must be extended; a change in the law occurs; or an unsafe condition is detected. Also, see modification.

Users: The personnel who operate, maintain, support, or dispose of an item delivered to the Government inventory or those who train such personnel.

Variation: The difference between the planned value of a technical parameter and the current assessed value.

Verifiable: Product compliance with a requirement can be verified at the level of the system structure at which it is stated by a finite and objective process.

Verification: The task of determining whether a system or item meets the requirements established for it.

Verification function: Tasks to be performed to evaluate the compliance of the evolving system (people, product, and processes) with the program or contract requirements. Includes analysis, demonstration, test, inspection, and special methods. The function includes technology assessments and demonstrations and all test and evaluation such as Development Test and Evaluation (DT&E) and Operational Test and Evaluation (OT&E). Also includes the evaluation of program or contract risks and monitoring the risks.

Verification method: A way to verify that a solution meets a requirement. The usual verification methods are test, demonstration, inspection, and analysis. Other, special methods are also sometimes applied. The verification method for each requirement should be included in the baseline containing the requirement.

Waiver: A written authorization to accept an item which, subsequent to the start of manufacture, is found to depart from specified requirements but nevertheless is considered suitable for use “as is” or after repair by an approved method.

Warfighter: An individual who directly fights a war.

Work Breakdown Structure (WBS): A product-oriented hierarchical tree composed of the hardware, software, services (including cross-product tasks such as SE), data, and facilities that encompass all work to be carried out under the program or contract along with a dictionary of the entries in the tree. The WBS for the entire program is called the Program or Project WBS (PWBS). The WBS for the work under the contract is called the Contract WBS (CWBS) and is prepared in accordance with the contract.

Appendix B–Acronyms

Note:The following is an alphabetical list of acronyms. (Many terms are defined in Appendix A.)

ACAT	Acquisition Category
ACT	Activity Diagram
ACWP	Actual Cost of Work Performed
ADC	Attitude Determination & Control
ADD	Architecture Description Document
ADM	Acquisition Decision Memorandum
AEHF	Advanced Extremely High Frequency
AF	Air Force
AFI	Air Force Instruction
AFMC	Air Force Materiel Command
AFMCI	Air Force Materiel Command Instruction
AFPD	Air Force policy Directive
AFRL	Air Force Research Laboratory
AFSCN	Air Force Satellite Control Network
AFSPC	Air Force Space Command
ANSI	American National Standards Institute
AoA	Analysis of Alternatives
ARM	Active Risk Manager
ASME	American Society of Mechanical Engineers
ASP	Acquisition Strategy Panel
ATD	Advanced Technology Demonstration
AV	All Viewpoint
BAC	Budget at Completion
BCWP	Budgeted Cost for Work Performed
BCWS	Budgeted Cost for Work Scheduled
BDD	Block Definition Diagram
BIT	Built-In Test
BOE	Basis of Estimate
CA	Control Account
CAIV	Cost As Independent Variable
CAM	Control Account Manager
CARD	Cost Analysis Requirements Description
CBA	Capabilities Based Approach
CBD	Component-Based Development
CCA	Clinger-Cohen Act
CCB	Configuration Control Board
CCTD	Concept Characterization and Technical Description
CDD	Capability Development Document
CDR	Critical Design Review
CDRL	Contract Data Requirements List
CEBoK	Cost Estimating Body of Knowledge
CER	Concept Explorations and Refinement
CI	Configuration Item or Counterintelligence
CIO	Chief Information Officer
CJCS	Chairman of the Joint Chiefs of Staff
CJCSI	Chairman of the Joint Chiefs of Staff Instruction
CM	Configuration Management
CMMI	Capability Maturity Model - Integrated
CNSSI	Committee on National Security Systems Instruction
COA	Continuing OSS&E Assessment
COCOMO ™ II	Constructive Cost Model II
COI	Community of Interest
CONOPS	Concept of Operations
COSYSMO®	Constructive SE Cost Model
COTS	Commercial off the Shelf
CPAT	Critical Process Assessment Tool
CPD	Capability Production Document
CPI	Critical Program Information or Cost Performance Indices

CPM	Capabilities Portfolio Management
CRS	Computer Resource Support
CSDR	Cost and Software Data Reporting
CSOW	Contract Statement of Work
CTE	Critical Technology Elements
CV	Cost Variance or Capability Viewpoint
CWBS	Contract Work Breakdown Structure
DAC	Designated Acquisition Commander
DAG	Defense Acquisition Guidebook
DAS	Defense Acquisition System
DCR	DOTMLPF Change Recommendation
DCS	Design Constraint Sheet
DDL	Delegation of Disclosure Authority Letter
DDos	Distributed Denial of Service
DepSecDef	Deputy Secretary Defense
DFD	Data Flow Diagram
DFT	Design for Testing
DG	Defense Guidance
DI	Design Interface
DIA	Defense Intelligence Agency
DISA	Defense Information Systems Agency
DISN	Defense Information Systems Network
DISR	DoD Information Technology Standards Registry
DIV	Data and Information Viewpoint
DM	Data Management
DM2	DoDAF Meta-Model
DMAIC	Define, Measure, Analyze, Improve, Control
DMSMS	Diminishing Manufacturing Sources and Material Shortages
DMSO	Defense Modeling & Simulation Office
DoD	Department of Defense
DoDAF	DoD Architecture Framework
DoDD	DoD Directive
DoDI	DoD Instruction
DOTMLPF	Doctrine, Organization, Training, Materiel, Leadership and Education, Personnel and Facilities
DP	Development Planning
DPG	Defense Planning Guidance
DSMC	Defense Systems Management College
DSN	Deep Space Network
DT&E	Development Test and Evaluation
DTS	Direct to Sailors
EA	Enterprise Architecture
EAC	Estimate at Completion
EHF	Extremely High Frequency
EIA	Electronics Industries Association
EIAP	Environmental Impact Analysis Process
EM	Electromagnetic
EMC	Electromagnetic Compatibility
EMD	Engineering and Manufacturing Development (Phase II)
EMI	Electromagnetic Interference
EOM	End-of-Mission
ES&H	Environment Safety and Health
ESOH	Environmental Safety and Occupational Health
EVM	Earned Value Management
EVMS	Earned value management system
FCA	Functional Configuration Audit
FEM	Finite Element Model
FFBD	Functional Flow Block Diagram
FFP	Firm Fixed Price
FISMA	Federal Information Security Management Act
FMEA	Failure-Mode Effects Analysis
FMECA	Failure Modes, Effects, and Criticality Analysis
FOC	Full Operational Capability
FOT&E	Follow-On Operational Test and Evaluation
FRACAS	Failure reporting, analysis and corrective action system
FRP	Full Rate Production
FRR	Flight Readiness Review
FTA	Fault Tree Analysis
GAO	Government Accountability Office
GBS	Global Broadcast System

GHz	Gigahertz
GIG	Global Information Grid
GPS	Global Positioning System
HBSS	Host Based Security System
HDBK	Handbook
HLA	High Level Architecture
HPF	Hazardous Processing Facility
HSI	Human Systems Integration
html	Hyper Text Markup Language
HW	Hardware
IA	Information Assurance
IAS	Information Assurance Strategy
IAW	In accordance with
IB	Issue Books
IBD	Internal Block Diagram
IBM	International Business Machines
IBR	Integrated Baseline Review
ICAM	Integrated Computer Aided Manufacturing
ICD	Initial Capability Document or Interface Control Document
IEC	International Electrotechnical Commission
IEEE	Institute for Electrical and Electronics Engineers
IICD	Internal ICD
ILS	Integrated Logistics Support
IM	Interface Management
IMP	Integrated Master Plan
IMS	Integrated Master Schedule
INCOSE	International Council on Systems Engineering
IOC	Initial Operational Capability
IOT&E	Initial Operational Test and Evaluation
IP	Internet Protocol
IPPD	Integrated Product and Process Development
IPT	Integrated Product Team
IPv6	Internet Protocol version 6
IS	Information Systems
ISDN	Inter-Services Digital Network
ISO	International Organization for Standardization
ISP	Information Support Plan
ISR	Intelligence, Surveillance, and Reconnaissance
IT	Information Technology
ITAMP	Integrated Task and Management (or Master) Plan (ITAMP)
ITU	International Telecommunications Union
IV&V	Independent V&V
JCA	Joint Capability Areas
JCIDS	Joint Capabilities Integration and Development System
JCTD	Joint Capability Technology Demonstration
JEON	Joint Emergent Operational Need
JLRSA	Joint Long Range Strategic Appraisal
JPAM	Joint Program Assessment Memorandum
JROC	Joint Requirements Oversight Council
JSPD	Joint Strategic Planning Document
JTIC	Joint Interoperability Test Command
JTRS	Joint Tactical Radio System
JUON	Joint Urgent Operational Need
KDP	Key Decision Point
KHz	Kilohertz
KPP	key performance parameter
KSA	Key System Attributes
LAI	Lean Aerospace Initiative
LCC	Life Cycle Cost
LCSP	Life-Cycle Sustainment Plan
LOE	Level Of Effort
LRIP	Low-Rate Initial Production
LSI	Large Scale Integration
LV	Launch Vehicle
M&S	Modeling and Simulations
MAJCOM	Major Command
MBSE	Model-Based SE
MDA	Mission Data Archive or Milestone Decision Authority
MDD	Materiel Development Decision
MID	Management Initiative Decision
MILHBDK	Military Handbook
MILSATCOM	Military Satellite Communications
MIL-STD	Military Standard

MIT	Massachusetts Institute of Technology
MNS	Mission Need Statement
MOE	Measures of Effectiveness
MOP	Measures of Performance
MOSA	Modular Open System Approach
MP	Mitigation Plan or Mass Properties
MR	Management Reserve
MRR	Manufacturing Readiness Review
MSA	Materiel Solution Analysis
MS A	Milestone A
MS B	Milestone B
MS C	Milestone A
MSS	Mobile Satellite Service
MTBF	Mean Time Between Failure
MTTF	Mean time to failure
MTTR	Mean Time To Repair
MTTRF	Mean Time To Restore Function
MUOS	Mobile User Objective System
N2	NxN Diagram
NASA	National Aeronautics and Space Administration
NBCC	Nuclear, Biological, and Chemical Contamination
NCDS	Net-Centric Data Strategy
NDI	Non-Developmental Item
NEPA	National Environmental Policy Act
NESI	Net-Centric Enterprise Solutions for Interoperability
NetOps	Network Operations
NIPERNET	Non-classified Internet Protocol Router Network
NIST	National Institute of Standards and Technology
NMI	NASA Management Instruction
NMS	National Military Strategy
NOAA	National Oceanic and Atmospheric Administration
NORAD	North American Aerospace Defense Command
NPOESS	National Polar-orbiting Operational Environmental Satellite System
NR-KPP	Net Ready Key Performance Parameter
NRO	National Reconnaissance Office
NSPAR	Non-Standard Part Application Request
NSS	National Security System, National Security Space
NSSAP	National Security Space Acquisition Process (Program)??
NuDet	Nuclear Detonation
O&S	Operation and Sustainment or Operations and Support
OA	Operational Architecture (as in OA View)
OAA	OSS&E Assurance Assessment
OMB	Office of Management and Budget
OMG	Object Management Group
OOA/D	Object-Oriented Analysis/Development
ORD	Operational Requirements Document
ORM	Operational Risk Management
OSA	Open Standards Architecture
OSD	Office of the Secretary of Defense
OSJTF	Open Systems Joint Task Force
OSS&E	Operational Safety, Suitability & Effectiveness
OSSA	Open-Standard System Architecture
OT&E	Operational Test and Evaluation (IOT&E and/or FOT&E)
OTA	Operational Test Authority
OV	Operational Viewpoint
P&D	Production and Deployment
PAF	Payload Attach Fitting
PBD	Program Budget Decisions
PBR	Program Budget Review
PC	Personal Computer
PCA	Physical Configuration Audit
PCI	Personal Computer Interconnect
PCO	Procuring Contracting Officer
PD	Parametric Diagram
PDM	Program Decision Memoranda
PDR	Preliminary Design Review
PEO	Program Executive Officer

PESHE	Programmatic Environment Safety and Occupational Health Evaluation
PFR	Post Flight Review
PHM	Prognostics & Health Management
PM	Program Manager
PMP	Parts, Materials, and Processes
PMBoK	Project Management Body of Knowledge
PMPCB	Parts, Materials and Processes Control Board
PMPSL	Parts, Materials and Processes Selection List
PMR	Procurement Management Review
POCC	Payload Operations Control Center
POH	Project Officer's Handbook
POM	Program Objective Memoranda
PPBE	Planning, Programming, and Budgeting Execution process
PPBS	Planning, Programming, and Budgeting System
PPF	Payload Processing Facility
PPM	Parts per million
PPP	Program Protection Plan
PRR	Production Readiness Review
PSC	Preferred System Concept
PSM	Product Support Manager or Program Supportability Management
PV	Project Viewpoint
PWBS	Program or Project Work Breakdown Structure (WBS)
QA	Quality Assurance
QFD	Quality Function Deployment
R&M	Reliability and Maintainability
RAA	Responsibility, Authority, and Accountability
RMA	Reliability, Maintainability, Availability
RAM	Reliability, Availability, Maintainability
RAS	Requirements Analysis/Allocation Sheet
RCA	Root-Cause Analysis
RCM	Requirements Correlation Matrix
RD	Requirements Diagram
RDT&E	Research Development Test & Evaluation
RF	Radio Frequency
RFP	Request for Proposal
RHA	Radiation Hardness Assurance
RHOC	Radiation Hardened Oversight Council
RHP	Risk Handling Plan
RMB	Risk Management Board
RMP	Risk Management Plan
S&T	Science and Technology
SAR	System Requirements and Architecture Review
SATCOM	Satellite Communications
SBA	Small Business Administration
SBIRS	Space Based Infrared System
SDP	Software Development Plan
SDR	System Design Review
SE	Systems Engineering or Support Equipment
SED	Specialty Engineering Disciplines
SEI	Software Engineering Institute
SEIT	System Engineering & Integration Team
SEMP	Systems Engineering Management Plan (Contractors)
SEP	Systems Engineering Plan (Government)
SEPM	System Engineering and Program Management
SFR	System Functional Review
SHF	Super high frequency
SIPRNET	Secret [formerly Secure] Internet Protocol Router Network
SLOC	Source lines of code
SM	Spectrum Management
SMC	Space and Missile Systems Center
SMCI	Space and Missile Command Instruction
SOA	Service-Oriented Architecture
SOC	Space Operations Centers or Satellite Operations Control
SoS	System of Systems
SOW	Statement of Work
SPAWAR	Space and Naval Warfare Systems Command

SPI	Schedule Performance Indices
SPO	System Program Office
SPS	Standard Positioning System
SRD	System Requirements Document
SRM	Solid Rocket Motors
SRR	System Requirements Review
SSMP	System Security Management Plan
STA	System Threat Assessment
STAR	System Threat Assessment Report
Std	Standard
StdV	Standards Viewpoint
SV	Systems Viewpoint or Schedule Variance
SvcV	Services Viewpoint
SVR	System Verification Review
SW	Software
SWAMP	Software Acquisition Management Plan
SWAPI	Software Acquisition Process Improvement
SysML	System Modeling Language
T&E	Test & Evaluation
TA/CP	Technology Assessment/Control Plan
TBD	To Be Determined (see definition in Annex 1)
TBR	To Be Resolved (see definition in Annex 1)
TBS	To Be Supplied (see definition in Annex 1)
TCPI	To Complete Performance Index
TD	Technical Data or Technology Development
TDP	Technical Data Package
TDRSS	Tracking and Data Relay Satellite System
TDS	Technology Development Strategy
TEMP	Test and Evaluation Master Plan
TM	Technical Manual
TO	Technical Order
TOC	Total Ownership Cost
TPM	Technical Performance Measures/Measurements
TQCM	Total Quality Control Management
TRA	Technology Readiness Assessment
TRD	Technical Requirements Document
TRL	Technology Readiness Level
TRR	Test Readiness Review
TT&C	Telemetry Tracking and Command
UAV	Unmanned Aerial Vehicle
UHF	Ultra-high frequency
UML	Unified Modeling Language
UON	Urgent Operational Need
UPC	Unit Production Cost (See also DTC, DTUPC)
URL	Uniform Resource Locator
USAF	United States Air Force
USCM	Unmanned Space Vehicle Cost Model
USD (AT&L)	Under Secretary of Defense (Acquisition, Technology, and Logistics)
USecAF	Under Secretary of the Air Force
V&V	Verification and Validation
VAC	Variance at Completion
VCJCS	Vice Chairmen of the Joint Chiefs of Staff
VSA&M	Value Stream Analysis and Mapping
VSM	Value Stream Mapping
W3C	World-Wide Web Consortium
WBS	Work Breakdown Structure (see also CWBS and PWBS)
WP	Work Package
XOR	Exclusive OR

Appendix C – Techniques of Functional Analysis

C.1 Functional Analysis Processes

Functional Analysis is often one of the major Systems Engineering activities. Functional analysis typically is first employed to assist in performing concept trades. Here, various functional/logical views are created that may focus on the operations/missions depicted by each concept under study. The functional analysis complement may reveal additional strengths and weaknesses that should be factored in to the ultimate selection of a final concept. Also, the functional analysis results may be cause to reconsider the Functional Area Analyses results.

Once systems definition begins, functional analyses usually the starting point to complement the concept architectures with system oriented functional views. There are usually two classes of views – those that are continue to focus on operations and those that focus on functionality supporting system design. In either case, the common functional elements are tracked between the two classes.

Functional analysis provides a number of benefits to support the system definition process:

- Provides information regarding system functionality essential to drive toward the best solutions.
- Initiates interface definition activities
- Discourages single-point solutions
- Aids in identifying lower-level functions/requirements
- Initiates and supports other activities such as failure modes analyses, fault detection/management, hazards analyses, operations procedures development, maintenance procedures development.

The systems definition team is rightfully influenced by the designers. Their knowledge makes for a better design. A potential drawback is that those with extensive design experience tend to start designing items before sufficient requirements have even been identified. It's like a reflex; they can't help it. Designers often drive towards single-point solutions without sufficiently considering/examining alternatives. Functional analysis yields a description of actions rather than a parts list. It shifts the viewpoint from the single-point physical to the unconstrained solution set. Although this may sound like functional flows deal only with the abstract that is not the case. The set of functional flows eventually reflects the choices made in how the system will accomplish all the user’s requirements. This characteristic is more apparent as you progress to the lower levels of the functional hierarchy.

Products have desired actions associated with them. These are usually actions that are visible outside the system/product, and directly relate to satisfying the customer's needs/requirements. Those that are internal to the system/product reflect functional and physical architectural choices made to implement the higher-level functions/requirements. Actions/functions are of interest in Systems Engineering because they really reflect requirements. Requirements associated with subordinate functions, themselves, will have to be accomplished by subordinate system elements. Functions, their sequential relationships, and critical timing need to be determined clearly to derive the complete set of performance requirements for the system or any of its subordinate system elements.

Functional analysis supports optimal functional and physical groupings to define interfaces. Verification, testability, and maintainability also improve through functional and interface analysis. Systems are less complicated and easier to support if the inputs and outputs of the subsystems and the interactions between subsystems are minimized.

Functional Analysis, alone, does not yield requirements. It does provide the essential framework for deriving the performance requirements for the system/product. Functional Analysis, working in tandem with requirements analysis provides a different approach for developing requirements for subordinate system elements. Other approaches flow requirements down to subordinate elements in the spec tree. Functional (requirements) analysis, on the other hand, by decomposing functions to produce the next level functional diagrams (e.g., FFBDs, IDEF0s), initially flows functions down without regard to what system element will perform them. Following the initial decomposition, alternate functional groupings are assessed to minimize interface complexity and determine candidate physical elements/resources that may be required for each alternative functional grouping. Of course, technology, risk, and cost trades are performed on the viable functional/physical choices as necessary.

Requirements are then derived to accomplish the functions, and each requirement is allocated/assigned to the system element that will then perform it. This approach facilitates system integration because as requirements are derived, those that identify a need to receive inputs from, or identify a product that needs to be output to, another entity can be worked to find a solution with minimal impact. In this way, functional analysis allows better functional and physical groupings for interfaces. Verification, testability, and maintainability improve through function and interface analysis. Systems are less complicated and easier to support if the inputs and outputs of subsystems and the interactions between subsystems are minimized.

The first step in this process is identifying the system’s functions. For any system/product, while there may be relatively few functions that can be identified from analysis of system-level user requirements and desired behaviors; there may be a larger number of possible functional architectures. There is no single right answer. Some approaches will be more productive in supporting the derivation of requirements than others. If the architecture selected starts to become a hindrance, go back and regroup. Knowing the shortcomings of the present architecture will help in developing its replacement. If the customer has provided their concept of a system's functionality, the functional analyst has additional insight into what the customer relay wants. However, this may not be the one on which to base your functional analysis. This is not license to ignore the customer’s wants, merely an invitation to explore other alternatives. The odds are that the functions chosen by the customer may not have been well thought out. Besides, functions' boundaries and scope are usually more than a little fuzzy until systems definitions are well underway. Sometimes the customer's description of the system provides more insight as to what is wanted than does their concept of the functions, or the requirements portion of their requirements document. The functions ultimately developed/chosen must accurately model the system's performance. Usually the architecture chosen is presented to the customer in a design review to make sure there is comfort with your choice.

Most engineers have little difficulty identifying primary or active functions of the product. For any communications system it’s easy to recognize the need for a data transmitting, a data receiving, and an operations control function. Supporting functions seem to be harder to grasp. Although not specified by the user, it may be customary (or mandated by overlooked directives) to archive data transferred. The archiving and retrieval would have to be captured by the functional architecture. The fact that the user wants the product to be continuously available, operable in an automobile, and transportable on his wrist is a little harder to work into lower-level functional requirements. These are design constraint requirements, and with the exception of the "continuously available", would not even need to be reflected in lower level flows. The means of achieving the availability would eventually have to be reflected in the much lower level flows. If there were redundant components, the automatic switching from the failed component to the operable spare would need to be portrayed in the flows, as would the sensing that a failure had even occurred.

The application of Functional Analysis is not limited to the system as a whole. It can be applied at any given level of product hierarchy within the system. Similarly, Functional Analysis is not limited to the Operational System; it may, and should, be applied to the development of requirements for the support equipment, training equipment, and facilities. These functions interrelate with the Operational System functions and coexist with them.

No single functional analysis methodology is sufficient by itself. Function analysis does not describe limitations, iteration, information flow, performance, or environments. However, it is a significant and essential tool is systems engineering activities Different types of requirement related information may be handled by the various implementation methodologies. Discussed below are two of the common methodologies widely used, the functional flow block diagram and timeline analysis.

C.2 Functional Analysis Allocation Categories

An effective systems engineering approach must perform, at a minimum, the following activities to produce an optimal system:

1. Accurately assess available information and find what is missing
2. Define performance or effectiveness measures that define success or failure
3. Manage and analyze all source requirements that depict user needs
4. Conduct systems analysis to formulate a behavioral design that meets all functional and performance requirements
5. Allocate functional behavior to the right physical architecture
6. Perform trade-off analysis to support decision making of alternative designs or architectures
7. Create executable models to verify and validate system operation
8. Use results and return to step 1

Steps four and five refer to developing a behavior model and allocating this behavior to the physical architecture. This allocation of functions and requirements to various system levels is both a rigorous discipline and an art form. It is a systematic definition of a system by starting at its simplest form and breaking it down into increasing more complicated components. The system engineer must simultaneously maintain his objectivity and independence from a design solution while keeping in mind the physical realities of cost, schedule and performance. The following examples are provided to illustrate the various types of allocations required to develop this behavioral design and allocate it to the physical architecture.

- Decomposition of functions to lower level functions
- Assignment of performance requirements to functions
- Assignment of constraints to functions
- Decomposition of constraints to lower level constraints
- Allocation of requirements to solutions

For the purpose of these examples, we will define the system of systems using the following taxonomy:

Level 1	Project
Level 2	Program
Level 3	Systems
Level 4	Elements
Level 5	Subsystems
Level 6	Components

C.2.1 Decomposition of Functions to Lower Level Functions

There are many tools available to assist the engineer in organizing his thoughts and work to accomplish the systematic breakdown and functional analysis of complex systems. The FFBD example in Figure C-1only illustrates the process.

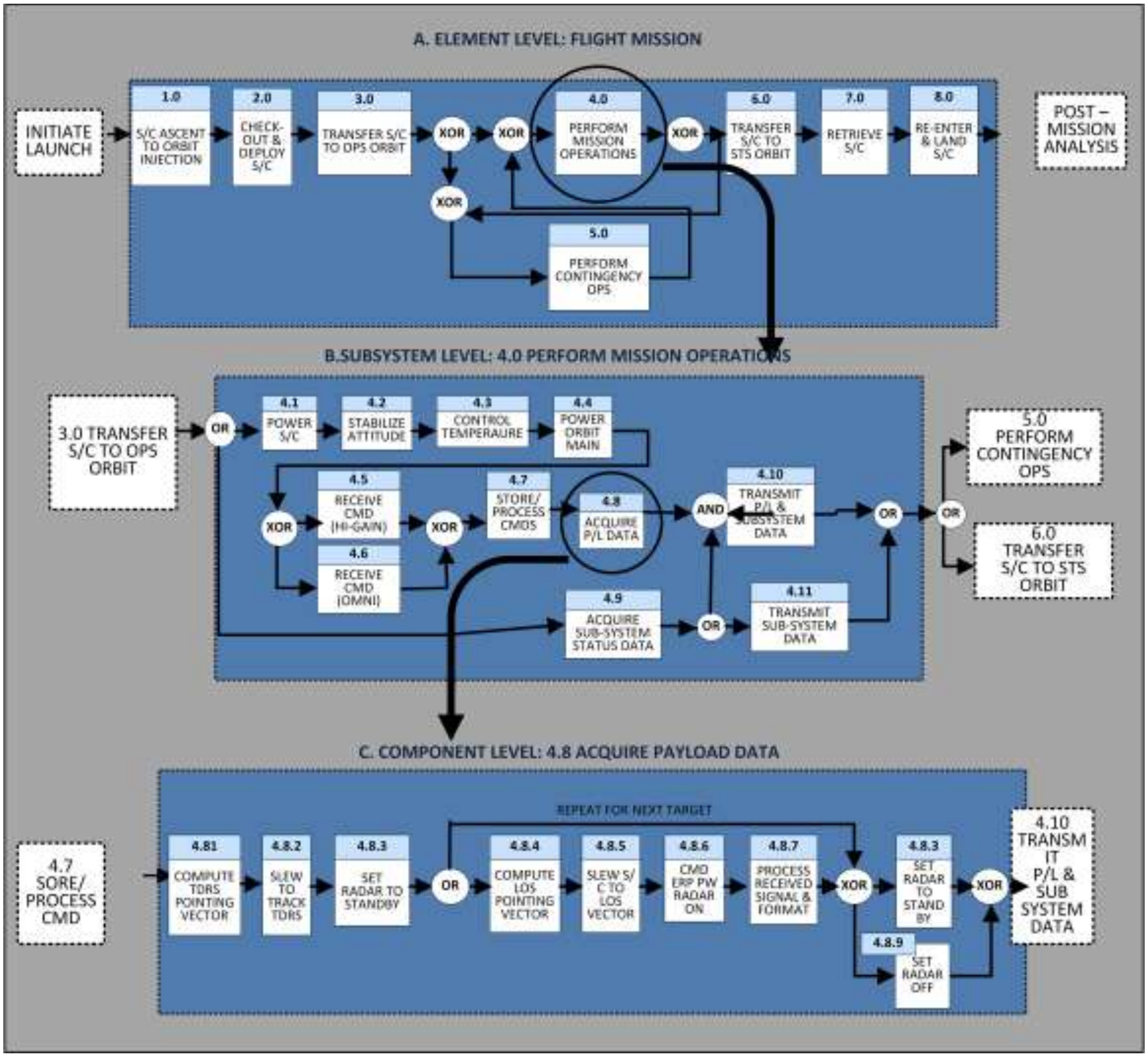


Figure C-1 A Functional Analysis Example

C.2.2 Assignment of Performance Requirements to Functions

This is one of the most obvious and most important aspects of functional analysis. This process will insure you have a complete set of requirements and also that you have properly modeled your system. Depending on the function you must ask yourself, how fast, how much, when, etc. The questions you ask will be dependent on the function. For example, refer to the function in Figure C-1 at the subsystem level called "Acquire P/L Data." The following types of requirements could be assigned to this function.

- The mission computer shall process payload data at a maximum rate of 3 Gbit/sec.
- The mission computer shall be available to collect payload data 23 hours a day, 7 days a week.

C.2.3 Assignment of Constraints to Functions

An example of a constraint assignment that applies to the same "Acquire P/L Data" would be as follows:

- The mission computer shall be available to collect payload data from a maximum of 12 sensors.

C.2.4 Decomposition of Constraints to Lower Level Constraints

Decomposition of constraints to lower level constraints should follow the same process as the functional decomposition. The engineer could perform an analysis or trade study to support the decomposition.

At the program level

- The ABC program shall distribute intelligence information via the XYZ Distribution Network.

At the system level

- The ABC Space System shall interface with the XYZ Distribution Network at the Peterson Node using TBD protocols.
- The ABC RPV System shall interface with the XYZ Distribution Network at the Bellows Node using CITRIX II protocols.
- The ABC Ground Vehicle System shall interface with the XYZ Distribution Network at the Polk Node using CITRIX protocols.

At the Space System element level

- The Space System ground element shall interface with the Peterson Node using TBD protocol.

C.2.5 Allocation of Requirements to Physical Architecture (i.e., Solution)

During the functional analysis process, the engineer assigns requirements to specific functions. As the functional analysis and the system design mature and become more detailed and specific, the engineer should be able to assign specific functions to specific physical components. The requirements associated with the functions become the source for developing verification and validation criteria. If functions cannot be mapped to the physical architecture, either the model or the design is flawed and the situation must be resolved.

Using one of the previous example requirements:

"The mission computer shall process payload data at a maximum rate of 3 Gbit/sec."

This requirement was previously assigned to the "Acquire P/L Data" function. When assigning to the physical architecture, the engineer should assign the requirement to the lowest level that will be verified. In this case the requirement may be assigned to a single processor or to the mission computer as appropriate.

C.3 Example of System Allocation and Assessment Process

The example selected is of a two-satellite system with redundant ground facilities. The customer only requires one of the two satellites to operate to meet the minimum mission requirements. The requirement for mission life is one year with a desire to continue it for at least four or more years. Of course there is a strong desire that both satellites operate throughout their lifetimes. The required probability of success of completing the one year mission is 0.9 with a goal of 0.97. An assumption is made that the launch is successful.

C.3.1 Preliminary Requirements Allocations:

Step one is to assign a preliminary set of reliability and maintainability requirements that meet the system requirement usually based on engineering judgment.

- Accepted goal of 0.97 as requirement
- Mission payload equipment needed to perform mission defined in system specification to be in an up and operable state at least 97% of the mission time
- Space Allocation
- SV design life = 5 years
- SV MMD = 4.5 years
- Ground Allocation
- Ground station A (MTBF = 450 hours; MTTR of any individual unit = 72 hours)
- Ground station B (MTBF = 475 hours; MTTR of any individual unit = 72 hours)
- MTTR of the satellite after a downing anomaly = 67 hours

C.3.2 Methodology for analysis:

For the System, develop reliability block diagrams using baseline design

- Describe all satellite subsystems, radar payload, and ground
- Identify redundancy and cross-strapping
- Total number of units
- Heritage of each unit
- Software items.

For the Space Segment

- Develop reliability model for the spacecraft system based on block diagrams
- Establish a design life and calculate mean mission duration (MMD)
- Modify model to reflect a single string design for spacecraft availability prediction
- Calculate mean time between failure (MTBF)
- Develop a mean time to restore function (MTTR) model based on historical data from other space systems.

For the Ground Segment

- Estimate MTBF for each unit
  - Vendor supplied data
  - Comparison with equipment in standard reliability handbooks
  - Engineering estimates
- Establish preliminary estimate of MTTR for each unit considering
  - Minimum sparing to support availability (formal provisioning analysis deferred)

  - Maximum use of commercial maintenance contracts with vendors assumes no logistics or administrative delays for this example.

Figure C-2 presents the results of the reliability assessment using reliability block diagrams, statistics, and failure rates in Mil-Hdbk-217. Reliability functions are calculated for each major element of the satellite and combined into an aggregate curve. Integration of this function from time 0 to the design life determines the mean mission duration (MMD) or average satellite lifetime.

Satellite dependability is calculated using a standard equation. Mean time between failure (MTBF) is calculated by integrating the satellite reliability function from time 0 to infinity. Mean time to restore (MTTR) is based on historical information of known orbital anomalies.

Dependability = MTBF/(MTBF + MTTR)

Mean time between failure (MTBF) is 17852.8 hours (from figure 54)

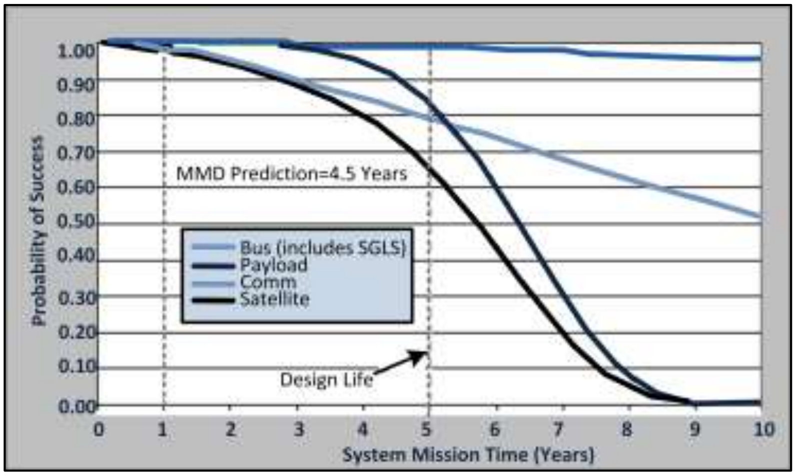


Figure C-2 Reliability functions calculated for each major element of satellite

- Historical on-orbit anomaly resolution
- 80% of all anomalies are corrected by switchover to redundant unit in 3 days
- 15% are watch and see
- 5% require functional workaround, further analysis, software mods, etc. in 8 days
- Mean time to restore (MTTR) is 67.2 hours

Figure C-3 predicts the probability that either one or both the satellites will fail during the mission lifetime. The results conclude that the probability of loss of a single satellite is less than 4 percent in the first year of the mission. The loss of both satellites in the first year is much less than one percent.

Table C-1 is an example of a ground segment allocation. Figure C-3 provides a depiction of the probability of loss of either one or both satellites due to random failure. The assumption is made that there is no loss due to wear-out of components or expiration of design life. In this example real equipment has been selected. MTBFs are based on historical data using the NPRD-25. MTTRs are based on engineering estimates.

Table C-2 below is the combined results of space and ground segment dependability. Either ground station can complete the mission without loss of

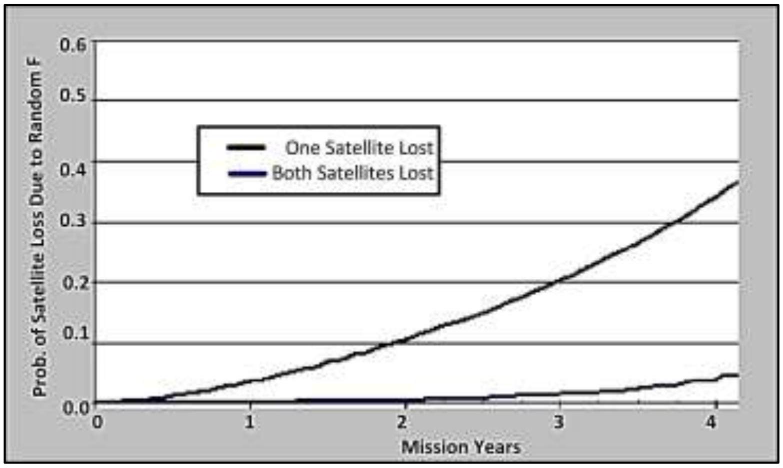


Figure C-3 Probability of loss of one or both satellites due to random failure data while the other is down. Combined availability for the ground segment is 0.98102. It can be seen that the mission can be successfully completed with one satellite out. Figure C-3 provides the summary results of a system dependability analysis. The conclusion is that the system will meet requirements.

Based on these results, the system engineer can allocate the preliminary requirements initially assumed to space and ground segment for implementation. The system engineer showed good engineering judgment at the beginning of this exercise. However, typically this is an iterative process to converge on an acceptable set of allocated requirements to meet the system requirement. Part of the iteration process is negotiations with segment managers to minimize their cost impacts.

Table C-1 Represents dependability of a single ground station

Ground Elements	Number	MTBF (hours)	MTTR (hours)	Individual Do
Operations Facility		475		
Antenna, trailer, Gimbal, and Electronics	1	6000	72	0.988142
Command &Telemetry Processor	1	9000	72	0.992063
Mission Data Archive (MDA)	1	9000	72	0.992063
Direct Demod/Bit Sync. (DDBS)	1	8265	72	0.991364
Data Formatter Unit (DFU)	1	75000	72	0.999041
IRIG-B	1	15000	72	0.995223
Adaptive Equalizer	1	15000	72	0.995223
Low Noise Amp.	2	9000	72	0.98419
SS High Power Amplifier	1	9000	72	0.992063
Common Imagery Processor (CIP)	1	5000	72	0.985804
Data Network	1	10000	72	0.992851
MYK-5	1	50000	72	0.998562
MYK-15	1	70000	72	0.998972
Fiber Optic Modem	2	15000	72	0.990469
SGLS Demodulator	1	9000	72	0.992063
SGLS Downconverter	1	9000	72	0.992063
SGLS Modulator	1	9000	72	0.992063
SGLS Upconverter	1	9000	72	0.992063
Dependability				**0.87252**
Total Ground Availability				**0.982367**

Table C-2 Summary results of a system dependability analysis

System Dependability Summary	1 Satellite Out	Both Operating
Space Segment	0.99999	0.99251
Ground Segment	0.98102	0.98102
System	0.98101	0.97367

Appendix D – Technology Readiness Levels

A widely accepted approach to systematically classifying individual technologies and comparing maturity between technologies is the Technology Readiness Levels (TRLs). The use of TRL approach has been in use for many years more predominantly for NASA space technology planning. This approach is now included in the NASA Management Instruction (NMI 7100) addressing integrated technology planning at NASA.

TRL 1—Basic Principles Observed and Reported. Lowest level of technology readiness. Scientific research begins to be translated into applied research and development. Examples might include paper studies of a technology's basic properties.

TRL 2—Technology Concept or Application Formulated. Invention begins. Once basic principles are observed, practical applications can be invented. The application is speculative and there is no proof or detailed analysis to support the assumption. Examples are still limited to paper studies.

TRL 3—Analytical and Experimental Critical Function or Characteristics Proof of Concept. Active research and development is initiated. This includes analytical studies and laboratory studies to physically validate analytical predictions of separate elements of the technology. Examples include components that are not yet integrated or representative.

TRL 4—Component or Breadboard Validation in Laboratory Environment. Basic technological components are integrated to establish that the pieces will work together. This is relatively "low fidelity" compared to the eventual system. Examples include integration of ad hoc hardware in a laboratory.

TRL 5—Component or Breadboard Validation in Relevant Environment. Fidelity of breadboard technology increases significantly. The basic technological components are integrated with reasonably realistic supporting elements so that the technology can be tested in a simulated environment. Examples include high-fidelity laboratory integration of components.

TRL 6—System/Subsystem Model or Prototype Demonstration in a Relevant Environment. Representative model or prototype system, which is well beyond the breadboard tested for TRL 5, is tested in a relevant environment. Represents a major step up in a technology's demonstrated readiness. Examples include testing a prototype in a high-fidelity laboratory environment or in simulated operational environment.

TRL 7—System Prototype Demonstration in an Operational Environment. Prototype near or at planned operational system. Represents a major step up from TRL 6, requiring the demonstration of an actual system prototype in an operational environment, such as in an aircraft, vehicle, or space. Examples include testing the prototype in a testbed aircraft.

TRL 8—Actual System Completed and Flight Qualified Through Test and Demonstration. Technology has been proven to work in its final form and under expected conditions. In almost all cases, this TR represents the end of true system development. Examples include developmental test and evaluation of the system in its intended weapon system to determine if it meets design specifications.

TRL 9—Actual System Proven Through Successful Mission Operations. Actual application of the technology in its final form and under mission conditions, such as those encountered in operational test and evaluation. In almost all cases, this is the end of the last bug fixing aspects of true system development. Examples include using the system under operational mission conditions.

Appendix E – Space Industrial Base Program

E.1 Purpose

The Space Defense Industry has undergone significant and critical supply base changes since the end of the Cold War. Rapid changes in the electronics industry and the acquisition of space technologies in foreign countries including the advancement and commercialization of these technologies have all precipitated a potentially damaging effect on the ability of the United States Air Force to obtain parts, material and processes for essential space programs. A dominant side effect of these changes is the continuing shrinkage of the U.S. Space Industrial Base. Many of the previous suppliers are no longer in business or no longer find it profitable to provide critical parts for USAF Space Programs.

Recognizing the seriousness of the endangerment to this supply base, SMC is developing a concerted effort to deal with our supply base issues. On June 17, 2004 the SMC Commander issued a directive titled "SMC Industrial Base Initiative" which established a product center wide Industrial Base process to be carried out by the then Systems Acquisition Directorate (SMC/AX) and now Systems Engineering Directorate (SMC/EN).

E.2 What is the form of the Assessment?

SMC/EN is gathering and populating a database containing all SPO prime and subcontracting suppliers. This database is a repository containing parts, materials and processes of each system segment of each program. This global repository contains all critical and current technologies, of all suppliers and their parts. This repository also contains assessments of these technologies, suppliers and parts to provide a living status of the Space Industrial Base.

E.3 How will this benefit SMC and the SPOs?

Through the institutionalization of this Industrial Base effort, the processes and expertise are preserved to provide tools to deal with IB issues. The preservation of these processes of recognizing and dealing with current and potential Industrial Base problems is a vital and essential part of our Space efforts. The Process Owners of this Industrial Base process resides with the Systems Engineering Directorate and exists to provide support to the SPOs in their industrial base efforts.

Although most Industrial Base issues affect more than one SPO, some issues do exist that may affect only one SPO. Providing a centric focus for such IB issues permits the efforts of each SPO to work in partnership with other SPOs, to avoid duplication and contributes to a value added relationship.

Appendix F – Bibliography

F.1 Systems Engineering Methodology

Athey, Thomas H. Systematic Systems Approach: An Integrated Method for Solving Systems Problems. Englewood Cliffs, NJ; Prentice-Hall, 1982.

Buede, Dennis M. The engineering Design of Systems – Models and Methods, New York: John Wiley & Sons, 2000.

Blanchard, Benjamin S. and Wolter J. Fabrycky, Systems Engineering and Analysis, 3rd Edition, New Jersey, Pren-tice Hall, 1998

Chase, Wilton P. Management of System Engineering. New York: John Wiley & Sons, 1974.

Chestnut, Harold. Systems Engineering Methods. New York: John Wiley & Sons, 1967.

Conrow, Edmund H. Effective Risk Management: Some Keys to Success, American Institute of Aeronautics and Astronautics, Second Edition, 2003.

Department of the Air Force, Air Force Instruction 90-901, Operational Risk Management, 1 April 2000, pg. 1. http://www.e-publishing.af.mil/pubs/publist.asp?puborg=AF&series=90

Department of the Air Force, Air Force Pamphlet 90-902, Operational Risk Management (ORM) Guidelines and Tools, 14 December 2002 http://www.e-publishing.af.mil/pubs/publist.asp?puborg=AF&series=90

Defense Systems Management College (DSMC), Systems Engineering Fundamentals, Fort Belvoir, VA, Defense Systems Management College Press, Dec 2000

Department of Defense, Risk Management Guide for DoD Acquisition, Defense Acquisition University, Fifth Edition, Version 2.0, June 2003.

http://www.dau.mil/pubs/gdbks/risk_management.asp

Department of Defense, Defense Air University, Defense Acquisition Deskbook

Department of Defense, U. S. Department of Defense Extension to: A Guide to the Project Management Body of Knowledge (PMBOK® Guide), Defense Acquisition University, First Edition, Version 1.0, June 2003, pp. 127-151. http://www.dau.mil/pubs/gdbks/pmbok.asp

Department of Defense, Systems Engineering Guide for Systems of Systems, 2008

Department of the Navy, Top Eleven Ways to Manage Technical Risk, NAVSO P-3686, Office of the Assistant Secretary of the Navy (RD&A), October 1998.

Grady, Jeffrey O., System Engineering Planning and Enterprise Identity, Boca Raton, CRC Press, 1995

EIA-632-1998, Processes for Engineering a System

Hall, Arthur D. A Methodology for Systems Engineering. Princeton, J: D. Van Nostrand, 1962.

Hall, Arthur D. III. "Three-Dimensional Morphology of Systems Engineering." IEEE Transactions on Systems Science and Cybernetic, Vol. SSC-5, April 1969, pp. 156-160.

Hill, J. Douglas and John N. Warfield, "Unified Program Planning." IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-2, November 1972, pp. 610-621.

International Council of Systems Engineering, INCOSE Systems Engineering Handbook, July 2000

IEEE Std 1220-1998, IEEE Standard for Application and Management of the Systems Engineering Process, The Institute of Electrical and Electronics Engineers

Martin, James, N., Systems Engineering Guidebook, Boca Raton, CRC Press, 1997

Maier, Mark W. and Eberhardt Rechtin, The Art of Systems Architecting, 2nd edition, CRC Press, 2002

Sage, Andrew P. Methodology for Large Scale Systems. New York: McGraw-Hill, 1977

Shaik, Kamran Open-Standard System Architecture (OSSA)For M-Code GPS User Equipment, Interim Report, GPS Joint Program Office, Space And Missile Systems Center, 16 April 2001.

Mission Requirements

NASA, NASA Systems Engineering Handbook SP 6105, June 1995

NASA. Introduction to the Derivation of Mission Requirements Profiles for System Elements. NASA SP-6503. Washington, D.C.: NASA, 1967.

Space and Missile Systems Center, SMC Guide Standard Systems Engineering Products, July 03.

F.2 Joint Application Design

Wood, Jane and Denise Silver, Joint Application Design How to Design Quality Systems in 40% Less Time. New York: John Wiley & Sons, 1989.

F.3 NXN Chart

Lano, R.J. A Technique for Software and Systems Design. Amsterdam, The Netherlands: North-Holland Publishing Co., 1979.

F.4 Optimization

Orvis, William J. 1-2-3 for Scientists and Engineers. Alameda, CA: SYBEX, 1987.

Pierrer, Donald A. Optimization Theory with Applications. New York: Dover Publications, 1986.

F.5 Taguchi Methods

American Supplier Institute. Taguchi Methods: Selected Papers on Methodology and Applications. Dearborn, MI: ASI Press, 1988.

Ogadke, Madhav Shridhar. Quality Engineering Using Robust Design. Englewood cliffs, NJ: Prentice Hall, 1989.

Taguchi, Genichi. Introduction to Quality Engineering: Designing Quality into Products and Processes. Tokyo, Japan: Asian Productivity Organization, 1986.

Index

Customer Review & Feedback Form

This form is provided for your convenience.

	DATE:
TITLE OF DOCUMENT: **SMC Systems Engineering Primer & Handbook** **Concepts, Processes, and Techniques**	DOCUMENT: ***4th Edition***

ITEM NO.	PAGE NO.	SECTION NO.	LINE NO.	COMMENT AND/OR RECOMMENDATION
	Gen			
	1			
	2			
	…			
	n			

NAME	RANK	TITLE
ORGANIZATION	DSN	E-MAIL

Request a MS WORD version of this form and forward completed form to:

Mr. Dave Davis david.davis.3@us.af.mil

Mr. Naim Awwad naim.awwad@us.af.mil

Mr. Douglas Taffinder douglas.taffinder.1@us.af.mil

Ms. Alice (Frankie) Shelton alice.shelton@us.af.mil

or

Ms. Teresa Yeh teresa.yeh@iseservices.com ISE Services Inc Innovative Space Engineering

Prepared by

AIR FORCE SPACE COMMAND

SPACE & MISSILE SYSTEMS CENTER

SPACE & MISSILE SYSTEMS CENTER
LOS ANGELES AIR FORCE BASE
ENGINEERING