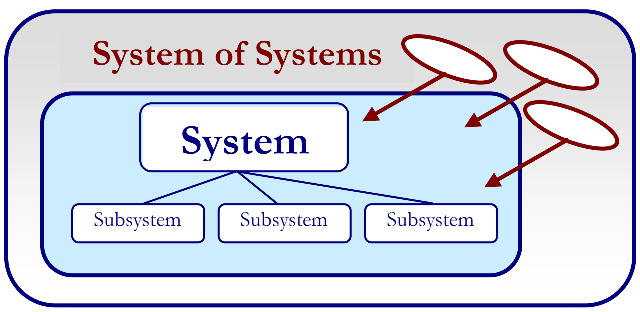
SMC Systems Engineering Primer & Handbook

Concepts, Processes, and Techniques



Space & Missile Systems Center
U.S. Air Force

2nd Edition **15 January 2004**

SMC Systems Engineering

Concepts, Processes, and Techniques

Table of Contents

List of Figures

List of Tables

Foreword

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 for your use. It is not all-inclusive and should be supplemented with Air Force and Department of Defense (DoD) directives, policies and procedures – see http://deskbook.dau.mil/jsp/default.jsp.

Preface

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

The authors recognize that systems engineering subject matter is very broad and that approaches to performing systems engineering 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 systems engineering practices and pitfalls to avoid. Many references are provided for the reader to consult for more in-depth knowledge.

This handbook describes systems engineering as it could be applied to the development of major space and launch systems. Systems engineering provides a disciplined approach that covers the entire lifecycle of a system to include development, design, manufacture, and operation. Consequently, the handbook's scope properly includes systems engineering 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 systems engineering training program. It is written to accompany formal SMC systems engineering training courses. The first chapter introduces the reader to concepts of systems and systems of systems. Chapter 2 expands on systems engineering concepts and terms and provides a more detailed explanation of the systems engineering 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 -- systems engineering management. Chapter 5 introduces the reader to common systems engineering tools and methods; Chapter 6 on specialty engineering integration, and Chapter 7 on validation and verification. The chapters are supplemented by appendices which include templates and examples to perform focused systems engineering related tasks.

Many different sources were used to prepare this book including the latest DoD Instruction and guidance on the subject, previous systems engineering 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. Over the next few years, the SMC Systems Engineering Revitalization (SER) initiatives will undoubtedly induce many changes to the conduct of engineering and acquisitions at the Center.

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@losangeles.af.mil or Barry Portner at bportner@tor.bdsys.com.

Acknowledgments

This work was conducted under the overall direction of Mr. Dave Davis, SMC Engineering. 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 it's predecessor – The GPS Systems Engineering Handbook.

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

Chapter 1
SMC Systems Engineering Primer

What is a System?

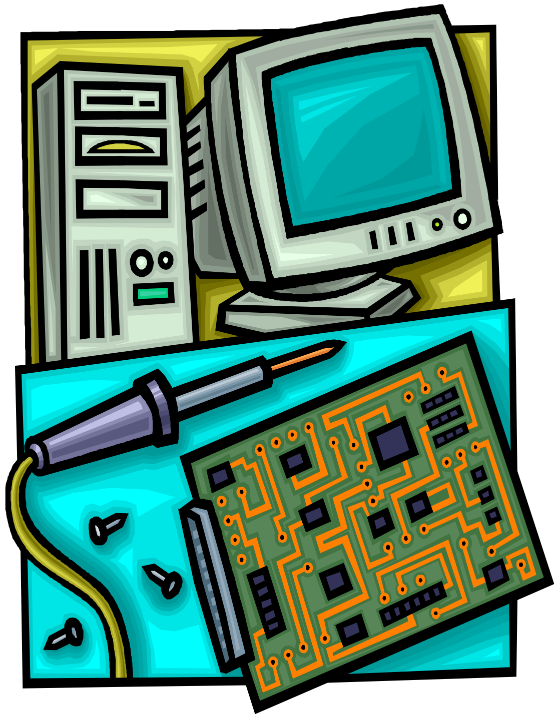


Figure 1. The personal computer system

A system can be thought of as a set of elements which interact with one another in an organized or interrelated fashion toward a common purpose which 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 is an example of a system. The elements of the PC include the processor, display, software, and keyboard. The soldering iron in Figure 1 is symbolic of the manufacturing, test, and maintenance equipment that are also system elements. The elements are organized or interrelated to achieve the purpose of the PC. The organization is facilitated by electrical cables and connectors and mechanical fasteners.

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 consist of the motherboard, the power supply, the case etc., all organized to carry out the processing. The motherboard is further made up of parts and materials which have been assembled via processes such as soldering. Parts, materials, and processes are the building blocks of most man-made systems.

Military Systems

The purpose of military systems is to provide a needed or desired operational capability to the military forces or to support the military forces in achieving or maintaining an operational capability. Thus, some military systems are weapon systems applied in combat while others are operational support systems used for training, testing, or characterizing the natural or threat environment in which the forces and equipment must operate.

Types of System Elements

The elements of a system may be quite diverse, consisting of hardware (equipment), 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 (to keep all elements working), training elements (to train people in the use of all the elements), and deployment elements (to install and checkout 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 of expendables or equipment no longer needed. Government facilities include control centers, launch pads, test and training facilities, and connecting roadways and utilities such as power and water.

What is a Space System?

Space systems include satellite systems which provide force enhancement or "force multipliers" (such as the Global Positioning System which enhances or multiplies the effectiveness of weapons by increasing their accuracy). Other space systems provide space support and include launch systems (such as the Atlas V and Delta IV) and terrestrial control systems (such as the Air Force Satellite Control Network or AFSCN). Concepts for space control may lead to space weapon systems in the future.

A satellite system is typically made up of one or more satellites (or space vehicles), terrestrial satellite control and maintain elements, and user elements that permit the operational military forces to take advantage of the capabilities of the space system. Each satellite is made up of its elements, typically the payload (that provides the basic mission capability such as communications, surveillance, navigation, etc.) and the spacecraft or bus (that typically supports the payload by providing electrical power, thermal control, and attitude control, etc.). 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.

Similarly, a launch system is typically made up of the launch vehicles (which 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 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 is a launch system that can 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.

What's different about a Space System?

Important differences derive from the fact that some of the elements of space systems are deployed or launched beyond the earth's atmosphere. Three major differences that significantly affect the engineering of the space systems will be addressed here: the space environment, unattended operation, and the implications of the ultimate high ground.

The Space Environment

The space environment places additional constraints on the satellites and the components and parts that make up the system – near total vacuum, ambient thermal inputs varying from direct sun illumination in one direction to the near absolute zero of deep space in others, and passage through belts of charged particles to name three. These constraints must be characterized, and the hardware must be designed to survive and operate in space. Special test facilities such as thermal vacuum chambers are required to verify that the hardware can operate in the 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 irretrievable failures during launch and early on-orbit operations.

Unattended Operation

All military space systems developed so far operate unattended. For that reason, if a component fails, only remote maintenance actions can be carried out. Such actions must usually be preplanned and take advantage of provisions designed into the hardware such as redundant hardware or re-loadable software. As a result, satellites are usually designed to eliminate (or at least minimize) single point failures. Also, redundancy has been increasingly designed into launch systems. When the redundant hardware also fails, the satellite may no longer provide the intended capability. Therefore, high reliability parts are also used. Further, care is taken to verify that the hardware has a positive margin with respect to the launch and space environments described above. 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 systems engineering task for all systems, but the high cost of space equipment places an extraordinary premium on balancing the operational capability to be provided with other factors such as cost, reliability, and service life. To achieve balance, alternative approaches or concepts must be compared or traded off against each other with respect to effectiveness, affordability, and risk.

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 enemy 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 by in joint operations by elements of all the military 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 systems engineering task of balancing effectiveness and cost can be still more demanding and important. The extreme example is the Global Positioning System (GPS) which 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).

What is a Family of Systems or a System of Systems?

Most modern systems operate in the context of a broader system of interrelated systems. An example is shown in Figure 2 below.

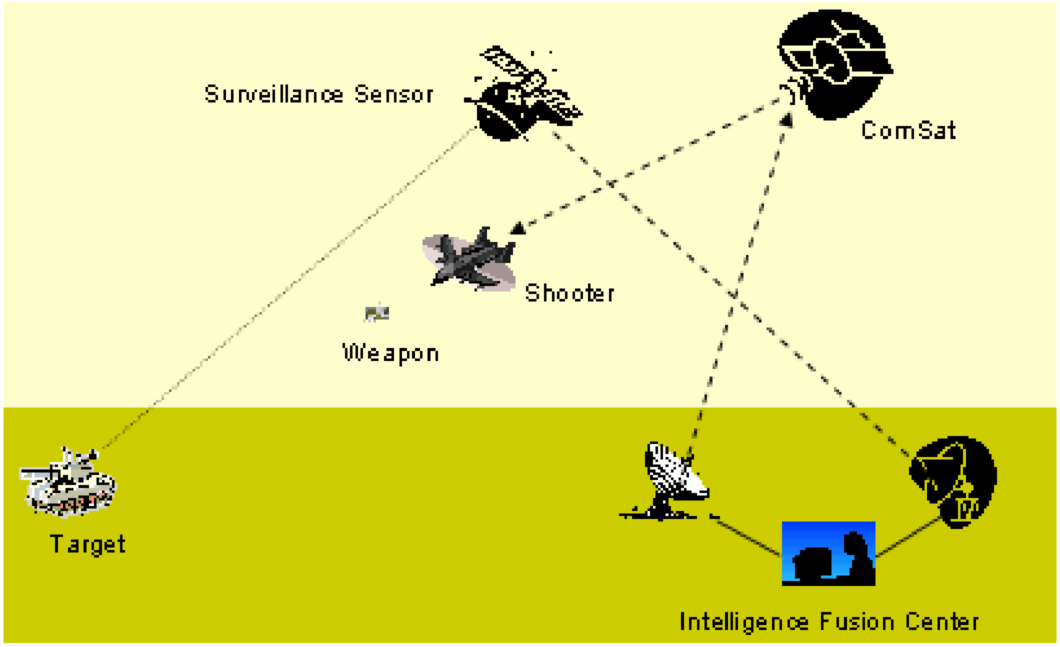


Figure 2. A Family of Systems–sensor to intelligence fusion center to shooter to weapon

As shown in the figure, each system must not only operate individually to provide the needed capability but must interface with or be interoperable with a number of other systems. To achieve interoperability, such systems must be engineered and evaluated in the family-of-systems context. When the individual systems become interdependent on one another to provide even a minimal capability that is needed, they become a system of systems. Managers of a family or system of systems, on the advice of their systems 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 a family or system-of-systems may have oversight authority over the design and operational decisions for each system.

What is a System Acquisition Program? (How does a military system come to be?)

Unlike the personal computer example cited above, 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 requirements to be satisfied) are defined. Then, alternative concepts to provide the capability (including maintenance and training) may be developed and evaluated to compare capability performance (effectiveness), affordability (cost), 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 and initial spare parts to a logistics center (depot).

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.

Department of Defense (DoD) Process for Acquiring a System

In the Department of Defense (DoD), the acquisition steps summarized just above take place as part of far-reaching governmental processes that formally define the needed capabilities, provide oversight of the acquisition programs, and provide the necessary budgets and other support to the programs that result in a new or improved military system. The DoD has three overarching and interactive management systems to implement these processes. The Joint Capabilities Integration and Development System (JCIDS)[1] oversees the definition of the capabilities (or operational requirements) that are to be satisfied – it is directed by the Vice Chairman of the Joint Chiefs of Staff. The Defense Acquisition System[2] oversees the research, development, test and evaluation, production, and deployment of a military system or system upgrade that provides new capabilities – it is managed by the Under Secretary of Defense for Acquisition, Technology, and Logistics, USD(AT&L). For selected space programs, the National Security Space Acquisition Process (NSSAP)[3] may apply streamlined acquisition procedures in lieu of the detailed requirements of the Defense Acquisition System. The budget for each program is developed within the DoD through the biennial Planning, Programming and Budgeting Execution (PPBE)[4] process which is managed by the Undersecretary of Defense. All three of these systems are supported by systems engineering activities that provide assessments of cost, schedule, and risk based on the evolving design of the system that is to provide the capability.

Many other Government processes or management systems support the acquisition of a new capability. After approval by in the PPBE process, the budgets are submitted by the President to the Congress for the annual Authorization and Appropriation of public funds. After public funds are appropriated by the Congress, they are managed by the DoD Financial Management

1. Chairman of the Joint Chiefs of Staff (CJCS) Instruction 3170.01C and CJCS Manual 3170.01.
2. DoD Instruction 5000.2, Operation of the Defense Acquisition System, May 12, 2003.
3. NSSAP 03-01, 28 July 2003.
4. See DoDD 7045.14; The Planning, Programming, and Budgeting System (PPBS) (Including Change 1); 28 July 1990. On May 22, 2003, the DoD announced that the PPBS would be streamlined to form the PPBE process which would focus on a biennial (two-year) cycle and use the off-year to focus on fiscal execution and program performance.

System.[5] The intelligence services provide information on the threat which could potentially act as a constraint on the operation of a system – the potential threat is usually dependent on the design of the system so that threat assessment should usually be carried out interactively with systems engineering. 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 systems engineering. Finally, the operational test community validates that a system provides the needed capability[6] – any deficiencies that are discovered must be resolved by systems engineering.

What is a System Capability Need (or Requirement)?

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 technology push or opportunities push. The second type of activity giving rise to new capabilities consists of operational problems or challenges which may be identified during training, exercises, operational testing, or military operations. Such capabilities are sometimes referred to as technology pull, 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 (the latter was called development planning in the past). As noted above, the operational capabilities to be provided by a military system are formalized by the Joint Capabilities Integration and Development System (JCIDS) for major programs.

Until recently, the needed capabilities were stated in terms of operational requirements by a predecessor of the JCIDS called the Requirements Generation System. Many programs continue to apply Mission Needs Statements (MNSs) or Operational Requirements Documents (ORDs) developed under the Requirements Generation System. The change in focus from mission needs and operational requirements to needed and desired capabilities has many facets including support to evolutionary acquisition by distinguishing between incremental capabilities and program end-state capabilities or requirements.

Early in the JCIDS process, an analysis of doctrine, organization, training, materiel, leadership, personnel, and facilities (DOTMLPF) capabilities and deficiencies is conducted in an integrated, collaborative process. If the analysis of DOTMLPF finds that a material solution is needed to resolve a deficiency or gap in existing military capability, then the JCIDS may conduct an Analysis of Materiel Approaches (AMA). As an example, the AMA might focus on the preferred approach between a space-based, aircraft, or ship-based approach to provide a surveillance capability but usually would not identify the specific system concept to be developed. Consistent with the preferred approach, the JCIDS defines the desired or needed capabilities to guide the selection of an alternative concept, develop a system, family of systems, or system of systems, and, ultimately, production of the equipment to fill the gap. The needed capabilities are defined in the Initial Capabilities Document (ICD) which describes the specific gap in capability that is to be filled and makes the case to establish the need for a materiel

5. DoD Financial Management Regulation (FMR).

6. 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.

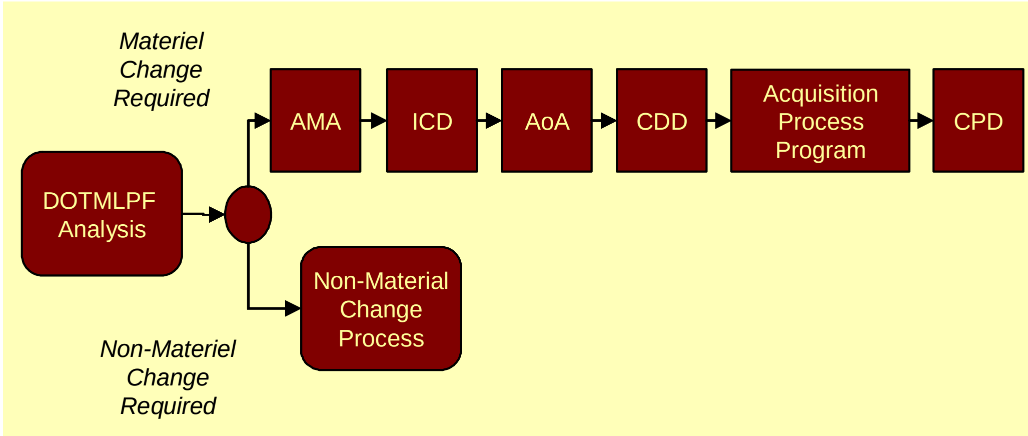


Figure 3. Key steps in the JCIDS process

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. These key steps in the JCIDS process are summarized in Figure 3.[7]

The JCIDS process just described will usually be applied to major programs (those with high projected cost or high-level interest). For non-major programs, the approach to defining the capability needs may be somewhat less formal but will usually include documentation of the need in documents such as the ICD and CDD.

Systems Engineering support to developing the need documented in the ICD, CDD, and CPD is covered in more detail later in this Primer and still further in Chapter 3 starting with Requirements Analysis.

What is a Technical Requirement or Constraint?

Usually, the operational capabilities to be provided must be subsequently translated into verifiable and allocable system technical (or engineering) requirements by the System Program Office (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 Document (SRD), a Technical Requirements Document (TRD), or a system specification and associated interface control documents or interface specifications. The terms used in the previous sentences of this paragraph are explained in the following paragraphs.

First, it can be helpful to distinguish between two aspects (or types) of system technical requirements: a functional requirement and a performance requirement. A functional requirement 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

7. For more detail, see Enclosure A in both the CJCSI 3170.01C and CJCSM 3170.01, both dated 24 June 2003.

operational capability. Others often result only from diligent systems engineering. Experience in systems engineering 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 performance requirement 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 as developed by the JCID. The statement of other performance requirements usually requires thorough systems engineering.

A constraint, as the word implies, is an imposed requirement such as an interface requirement (for example, 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. One constraint sometimes imposed by policy (or by program decision makers) is in the form of cost such as a maximum on the estimated cost for production.

System technical requirements and 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 are dependent on the design solution (and so are sometimes called design requirements). They include internal interface constraints between the elements of the system.

One term used above remains to be discussed. “Verifiable” means that the compliance with a requirement can be determined or verified by achievable and objective means, i.e., not subjective. Most requirements are verified by test, demonstration, inspection, or analysis.

The System Environment

All systems must both operate in the natural environment and can also affect that environment. The storage, transportation, and operating environments all usually give rise to constraints on the system. These include such factors as high and low temperature extremes, humidity, salt water spray (for ocean-bound equipment), and the like. Environmental factors applicable specifically to space systems are discussed above. The effect of the system on the environment is typically constrained by public law and by governmental regulations that implement the law – these must also be identified by the systems engineering process.

In addition, most military systems must be capable of functioning in a combat environment in which they are being directly attacked or in which the natural environment has been modified by intentional or unintentional enemy or friendly activity. The definition of the constraints imposed by the combat environment usually start with a system threat assessment which may be conducted by the DoD intelligence community. But such constraints may also come from other sources. For example, public law requires that many systems be exposed to live-fire testing as part of the verification that they will provide the needed capability in the combat environment – such may give rise to the requirements both for features that facilitate the testing as well as very specific survivability features.

What are Interfaces?

As noted under the discussion of Family and System of Systems above, systems usually do not operate alone. The relationship between two systems is called the interface. 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.

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. To distinguish, 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 by groups with different engineering specializations or different subcontractors.

Interfaces can be physical or functional. Physical interfaces include definitions of the means of attachment (bolt patterns, connectors, fasteners, etc.) and keep-out volumes. Functional interfaces include electrical, radio-frequency, and software.

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 equipment such as a 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.

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). Similarly, the 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.

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 physical interface between the Personal Computer and the printer may be via a cable that meets the requirements of a standard parallel interface specified by a standards-issuing organization such as the Institute for Electrical and Electronics Engineers (IEEE) or one that has become a de facto standard such as those followed by certain cable manufacturers. 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.

What is Architecture or a System Architecture?

Many of the ideas discussed so far are often brought together in one way or another in what is often called an architecture. 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.[8] During the 1960s, the term was extended to computer systems where it generally referred to the way the electronic hardware was organized

8. 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.

to process software instructions which 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.[9] 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 systems engineering 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.

Architectural Standards

Instead of the focus on the elements of a system and their organization as discussed above, some definitions of architecture address some higher level property or attribute of the system somewhat analogous, for example, to the Doric architecture of classical Greek buildings. In some cases, this higher level property is captured in architectural standards that define some important property such as the example of upward compatibility for computer architectures mentioned earlier. Two such properties are important to military systems: openness and interoperability. Before discussing those properties, there is one more definition of architecture that is should be described.

CJCS Definition

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.”[10]

Open-Standards Systems Architecture Approach

Responding to changes in national policy, modernization needs, user requirements, mission application constraints, and DoD mandates on open standards, the SMC is emphasizing use of open standards to reduce system acquisition cost, foster vendor competition, and reduced program risk.

There are many, often self-serving, definitions advanced to describe an open-standard system architecture (OSSA). The objective here is not to produce yet another, but to reiterate what is already accepted within DoD mandates. There are six basic elements of an open architecture: Open-Standards, Interoperable, Interchangeable, portable, modular, and scalable. The definition of these elements, listed below, is based on Joint Technical Architecture (JTA), DoD 5000.2-R, Open Systems Joint Task Force (OSJTF), Technical Architecture Framework for Information Management (TAFIM), and IEEE documents:

Open Standards: Parts, modules, objects, products, and systems are based on vendor-independent, non-proprietary, publicly available, and widely accepted standards. Standards

---

9. Maier, Mark W. and Eberhardt Rechtin, The Art of Systems Architecting, 2nd edition, CRC Press, 2002, p. 285ff.
10. See CJCSI 3170.01C, 24 June 2003, p. GL-4.

allow for a transparent environment where users can intermix hardware, software, and networks of different vintages from different vendors to meet differing needs. [JTA, OSJTF, DoD 5000.2-R, TAFIM, IEEE P1003.0]

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. [Joint Pub. 1-02, DoD/NATO, JOPES ROC, TAFIM]

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. [JTA, OSJTF, TAFIM, IEEE P1003.0]

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. [IEEE STD 610.12, TAFIM]

Modular: Physical or logical modularity to meet functional requirements. [JTA, OSJTF, TAFIM, IEEE 1003.0]

Scalable: The ability to grow (and interlink hardware and software) to accommodate increased loads. [TAFIM]

As mandated by the JTA, the standards selection criteria include (i) publicly available, (ii) consistent with authoritative sources, (iii) interoperable, (iv) maturity, and (v) implementable (JTA para 1.6). Furthermore, "… JTA mandates commercial standards and practices to the maximum extent possible (JTA para 1.5).

Many open standards are built by industry groups or academic/engineering organizations. Yet, other many of the "widely accepted" commercial standards start their life as "mandates." The PCI bus was forced on to the electronics industry by Intel; ActiveX and other Microsoft inventions have become de facto but largely proprietary standards with published interfaces! There are less dramatic examples: Apple Computer developed Firewire, a high-speed bus to connect computer peripherals, before it became IEEE 1394; and AMD developed the HyperTransport, a high-speed on-board bus architecture, before it handed over the work to an industry-led open consortium early in 2002. Even Linux, the open-source operating system for computers, started its life as a fully formed brain-child of T. Linus. 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 has failed in the marketplace as the technology is 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 sense.

Why OSSA

DoD use of an open systems approach will reduce 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 facilitating program manager 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. [11]

What is Forward Compatibility?

Backward compatibility is a very desirable end. However, many fail to recognize that achieving the cost-benefit of backward compatibility is a rather elusive goal. “Black box” design of legacy systems and products do not lend themselves easily to cheap backward compatibility. Backward compatibility must be designed into the product. Backward compatibility works when the design is thoughtfully modular, open, and scalable. The product is designed and built with the current and future technology and requirements in mind. We call this concept forward compatibility. OSSA provides this opportunity.

The C4ISR Architecture Framework

The JTA is intended to be applied in conjunction with other DoD initiatives including the C4ISR[12] Architecture Framework. The Framework provides direction on how to describe and integrate architectures developed by the various DoD Commands, Services, and Agencies. Toward that end, the Framework describes three major perspectives, i.e., views, that logically combine to describe an architecture. These are the operational, systems, and technical views. The C4ISR Architecture Framework is discussed further below in Chapter 3 and in more detail in Appendix C11. [13] For more information, refer to the Project Officer's Handbook (POH), Chapter 10 Concept of Systems Architecture Tools at www.smc.sparta.com/golive/site16.

Architecture–A Recap

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

Acquisition strategy to foster and develop open standards based program architectures is mandated by the DoD and adapted by the SMC 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 will

---

11 Office of the Inspector General, Department of Defense Report, "Use of an Open Systems Approach for Weapon Systems," No. D-2000-149, June 14, 2000

12. C4ISR -- Command, Control, Communications, Computer, Intelligence, Surveillance, and Reconnaissance.

13. For still more detail, see the Joint Technical Architecture User Guide and Component JTA Management Plan, Section 5.1, at http://www.disa.mil/main/jta.html.

be remiss if we did not emphasize that the success of OSSA 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.

Openness and interoperability depend on standards that are openly available, documented, and maintained. The JTA is a documented set of architectures for achieving interoperability between systems and forces operating jointly.

The C4ISR Framework defines steps toward implementing standards in the Joint Technical Architecture to achieve interoperability between systems and forces.

What is Systems Engineering?

Before considering the meaning of systems engineering, it is useful to reflect on the meaning of engineering more generally:

Engineering is the application of science to develop, design, and produce logical and/or physical objects such as buildings, machines, or a computer program to fulfill a desired need or to achieve an objective.

So the object or goal of engineering is a design. For example, aerospace engineering is the application of physics, material sciences, and other science to design space systems such as launch systems and satellite systems. It is important to note that in most cases the engineer has no direct way to arrive at the design such as by a set of formulas that can simply be evaluated. Instead, he or she must create (or invent) and plan. Often, as the engineer works through each aspect of the design, multiple alternatives come to mind. In the case of a launch system, alternative booster configurations, internal payload adaptive configurations, materials selections and many other plausible solutions might be considered. Thorough engineers will trade-off the promising alternatives. That is, the engineer will first compare the promising alternatives by evaluating or assessing each, usually by analysis or test. He or she will then select one that meets the objective of the design and otherwise balances such factors as cost, producibility, and the design margin that accounts for uncertainties such as material properties and worst case dynamic loading on the structures. Thus, although engineering applies science, it is an art rather than a science.

The engineer must also communicate his creation and plan to those who will build the system. The engineer therefore documents or specifies the design, often via drawings, parts lists, and step-by-step manufacturing instructions for hardware and such means as a use-case or an activity diagrams for software.

To state the obvious then, systems engineering is the engineering of a system – it is the application of science to design a system. As the details of systems engineering are discussed below, the reader will find it helpful to keep in mind that the ultimate objective is a design for the system. All else is important and useful only to the extent that it contributes to the efficient achievement of that objective.

In keeping with the definition of a system given above, you can probably appreciate systems engineering addressing those aspects of design having to do with specifying and organizing (or interrelating) the elements of a system to achieve the purpose that goes beyond what the individual elements of the system can achieve acting alone. To understand in more detail how systems engineering is different from other types of engineering such as electrical or mechanical engineering and why it is important to the success of a military acquisition program is not so easy for several reasons. First, systems engineering has evolved and continues to evolve as the systems that are to be engineered become more complex and engineering the systems becomes correspondingly more demanding. In particular, systems engineering for military space systems is evolving to keep pace with and support the evolving space systems acquisition policy. A brief history is given near the end of this primer to provide an indication of the trajectory on which practicing systems engineers for military space systems find themselves as they try to stay abreast of their profession.

Second, the details of systems engineering is usually described in terms of the steps in a process flow diagram. But starting with a fully developed flow diagram describing a particular process can obscure why those steps have been found to be useful to achieving a system design.

Third, no single process has achieved universal acceptance. There are differences in the flow diagrams, terminology, and the specifics in the processes described in various textbooks, standards, and corporate policies. Thus starting with a particular process flow diagram can also obscure the steps that are common to most processes.

For the reasons just cited, the following will first provide an appreciation for each of the steps that are common to most processes through a text discussion. A series of flow diagrams of increasing complexity will then be used to fill in the details applicable to processes typically used in military acquisition programs.

Systems engineering 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, systems engineering 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, systems engineering 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 formalize the relationship between multiple teams of engineers, systems engineering focuses on allocating the system requirements to the system elements to be designed by each team.

But before the allocation can take place, the systems engineer 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 technical requirements.[14] 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 a few paragraphs earlier that there is no direct way for an aerospace engineer to arrive at the design of a 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. Thus, systems engineering is an art, not a science.

14. 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.

If a system element is of sufficient complexity, the art of systems engineering will be applied in turn to it. In this sense, systems engineering 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.

The steps just described are summarized in Figure 4.

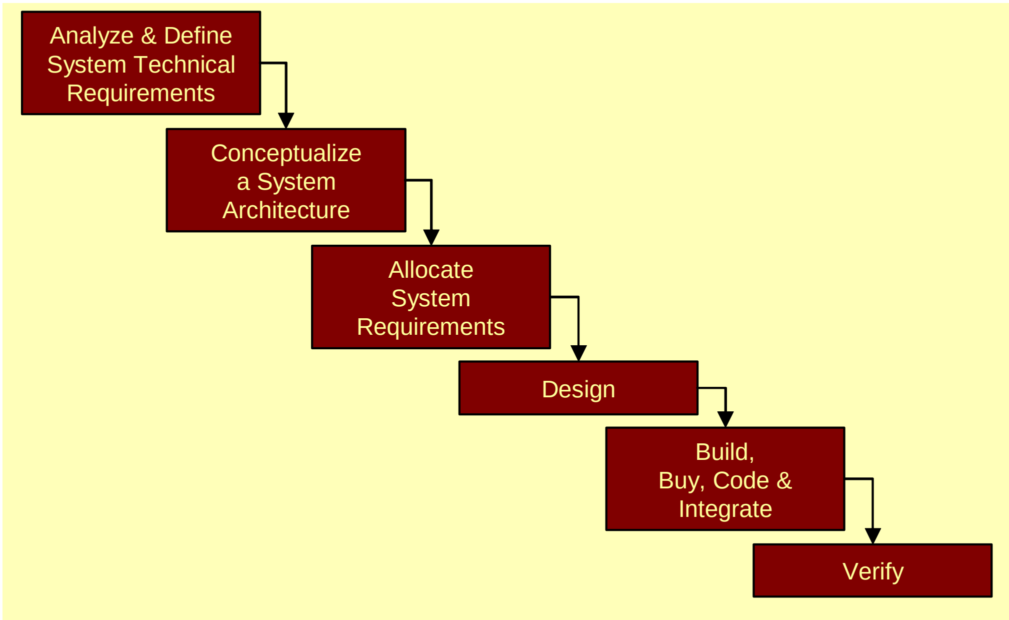


Figure 4. Engineering process to develop a system

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 systems engineering is an art, not a science.

Each iteration is guided by the primary tool of other forms of engineering, the trade-off to compare alternative statements of the system technical requirements, alternative system concepts or architectures, alternative requirements allocations, and alternative designs to achieve a balance between such factors as effectiveness, cost, schedule, and risk. For complete iterations that precede manufacturing and coding of system elements that can be directly verified, the effectiveness of the system and the cost, schedule, and risk to develop, produce, and deploy it are assessed by analysis. Achieving and maintaining a focus on the balance is usually essential to the success of a military acquisition program – an unbalanced emphasis on one requirement or factor over others usually results in a poor solution and the concomitant schedule delays.

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 systems engineering 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, systems engineering must also provide for the integration of specialists in safety, reliability, EMI, and other areas to help define and allocate the requirements, complete the design, and verify that the design satisfies the requirements.

What is the Systems Engineering Process?

Most descriptions of systems engineering are in terms of the process for carrying out the iterations and associated tradeoffs that result in a design for the system that fulfills the needs and desires of the users and other stakeholders. At this time, there is no single accepted systems engineering process. Instead, the process varies somewhat among textbooks, handbooks, standards, and corporate policies. The following description recounts typical systems engineering process for developing a military space system within the context of the 5000 series of DoD acquisition directives and instructions and the National Security Space Acquisition Policy (NSSAP) as well as instructions and manuals for the capability needs process issued by the Chairman of the Joint Chiefs of Staff (CJCS).

In DoD programs, the process usually 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.

Requirements Analysis and Concept Refinement

Systems Engineering Program Foundation

A simplified systems engineering process, shown in Figure 5, begins with the identification of the needed capability and other related stakeholder issues that establishes the foundation for systems engineering 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 usually have a significant role in the selection of the concept for further development. For example, the Operator/Users will usually 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[15] may establish objectives or goals which indicate that increased

15. 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.

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 systems engineering trade space in which the effectiveness, cost, schedule, risk, and growth potential of alternative design concepts can and should be assessed and compared to select architectures for further development that is balanced with respect to those factors.

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 development is essential to defining and evaluating the architecture or design concept.

The program foundation usually includes considerably more than the needs identified by the Operator/Users. For example, constraints such as external interfaces imposed by other systems; the storage, transportation, and operating environments (terrestrial or space) such as temperature, electromagnetic, and; 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 more demanding capabilities might lead to a satellite requiring a more capable launch system (Atlas V or Delta IV) having a different interface, such as a vibration 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 environmental impact and safety hazards are important to understanding which concepts will be useful and practical.

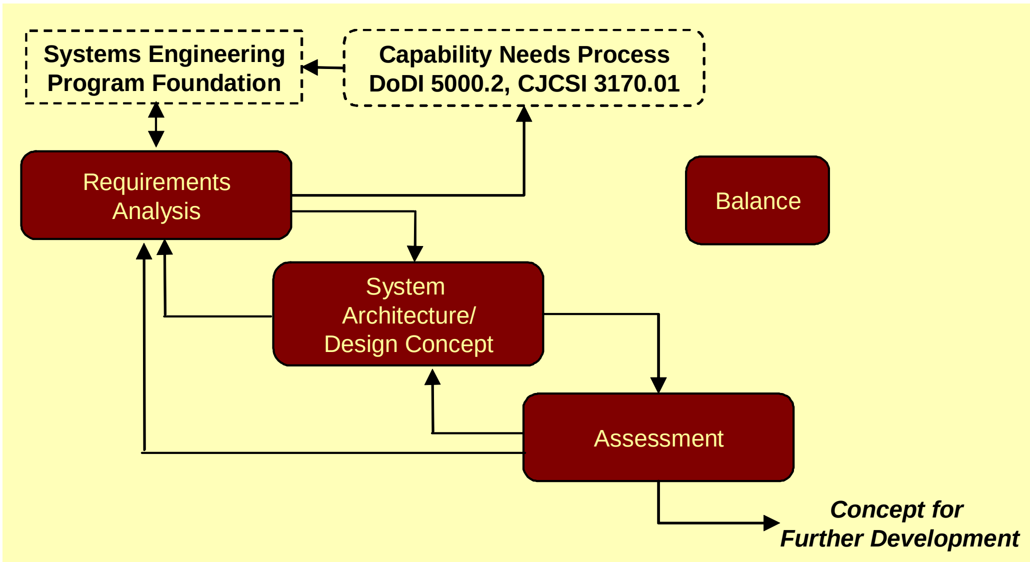


Figure 5. Simplified systems engineering process

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).

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, we will see that 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. By next stating how well each function must be performed, the performance requirements can be specified. 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 should then be validated to demonstrate that they completely and accurately reflect the needed capabilities and constraints. In part, this can be 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 electromagnetic compatibility in the operational and test environments and meets RF allocations and constraints, integrates human factors, effects safe use and controls any associated hazards, eliminates or controls the vulnerabilities to security threats, is in accord with DoD and service regulations and public law, and 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 systems engineering process.

Architecture or Design Concept

In most cases, there is no analytic synthesis approach for defining a system architecture or design concept responsive to the technical requirements and constraints. Instead, each is created or invented based on an understanding of the technology and design state of the art followed by an assessment of its responsiveness to the technical requirements and constraints as well as its balance with respect to 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.

Assessment

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. Instead, cost can be assessed only by extrapolating historical experience. 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 which 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 a 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.

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 the indicated by the feedback arrow to the Capability Needs Process in Figure 5 above. 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. An iteration can be confined to a single step or, as the feedback arrows in Figure 5 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. Consider the following data, shown in Figure 6, which might have been formed by a series of iterations through the process shown in Figure 5.

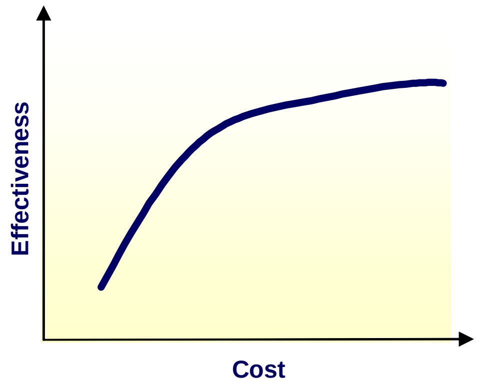


Figure 6. Cost effectiveness curve

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 the above figure 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 the significance of the arrow in Figure 5 between requirements analysis and the Capability Needs Process. One outcome could be the decision for further technology development to achieve risk reduction before formally starting a new program.

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 it may be selected by the program decision makers as a basis for further development to fulfill the need.

Relationship of Systems Engineering to Program Decision Making

Several examples are cited just above in which the products of systems engineering could be used by the program decision makers for decisions on the needed capabilities, technology maturation, concept selection, and program approval for further development. Other decisions that can be made based on systems engineering 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 systems engineering assessments and other products. It can be useful to view systems engineering as a staff function to the Government and Contractor program managers for such tasks as (1) requirements analysis, definition, and allocation, (2) system status assessment, and (3) risk management.

Both the 5000 series of DoD acquisition directives and instructions and the NSSAP 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. Based on the systems engineering assessments flowing from a process like that in Figure 5, the decision makers can cancel further development or approve either further concept and technology development or the start of a program to complete the system development and design.

Further Concept and System Development and Design

When the concept is approved for continued development, the simplified process in Figure 5 is usually expanded along the lines shown in Figure 7.

As in Figure 5, an iteration can involve a single step, two or more steps, or all of the steps. In comparison to Figure 5, however, several steps have been re-titled somewhat to show their more general application over the next iterations of the process. In addition, two 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 5 has been expanded into systems engineering 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.

Baselines

The objective of the process in Figure 7 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 systems engineering process. It can be helpful to maintain a clear distinction between the products of systems engineering and products that are defined by the design process and constitute the elements of the system.

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 systems engineering 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 instead of requirements baseline.

Functional Analysis and Allocation

This step starts with the functional requirements identified in the requirements analysis step and decomposes them and their associated performance requirements into sub functions to the point that they can be unambiguously related to the system elements or products that make up the design that flows out of a later step. 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.[16]

---

16. 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.

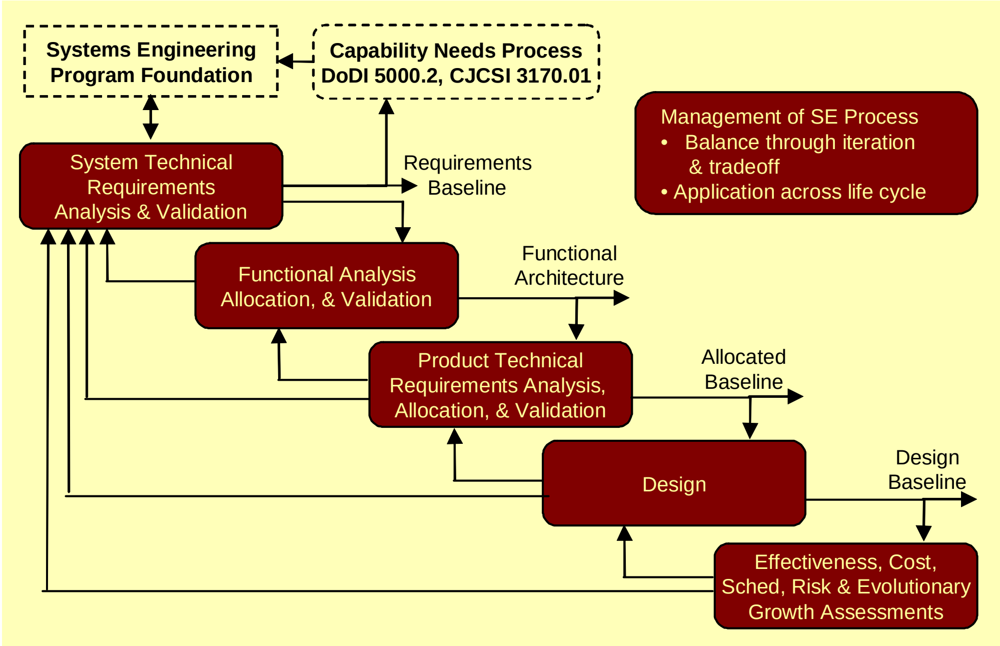


Figure 7. Systems engineering process for design

An often used view or presentation of the functional architecture is the functional flow block diagram (FFBD) supplemented by other devices such as timing diagrams that identify the timing relationships between functions. The FFBD shows the order in which the functions must be carried out to provide the capability. The figure below shows the eight primary lifecycle functions organized into a simple, generic top-tier FFBD.

To develop the functional architecture, the top tier FFBD in Figure 8 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. Also, Appendix C5, Techniques of Functional Analysis, provides a more comprehensive treatment of functional analysis.

Approaches Other Than Functional Analysis–the Logical Solution Representation

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 "object-oriented analysis, structured analysis, and information engineering analysis" for developing the "logical solution representation,"[17] a more general title for the functional architecture 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.

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

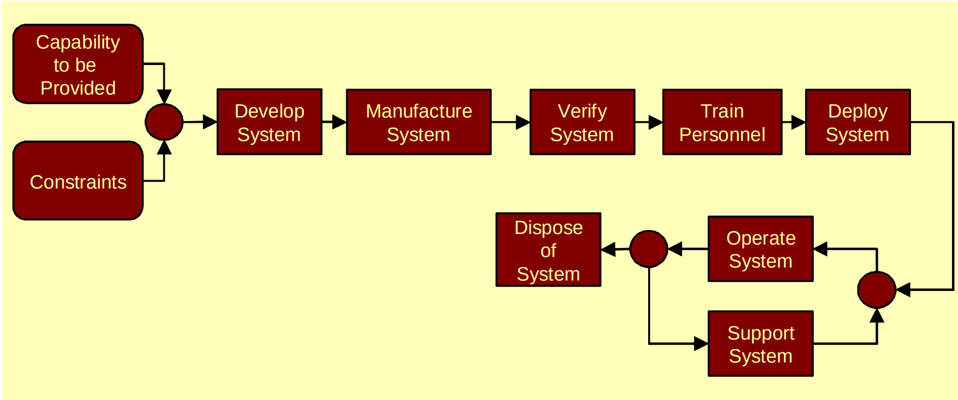


Figure 8. A top level functional flow block diagram showing a simple, generic relationship among the eight primary life cycle functions

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.

Product Requirements Analysis and Allocation

An important objective of the systems engineering process in Figure 7 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.

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 9 below.

The physical hierarchy can be a powerful tool for organizing many of the tradeoffs that form the iterations depicted in Figure 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.

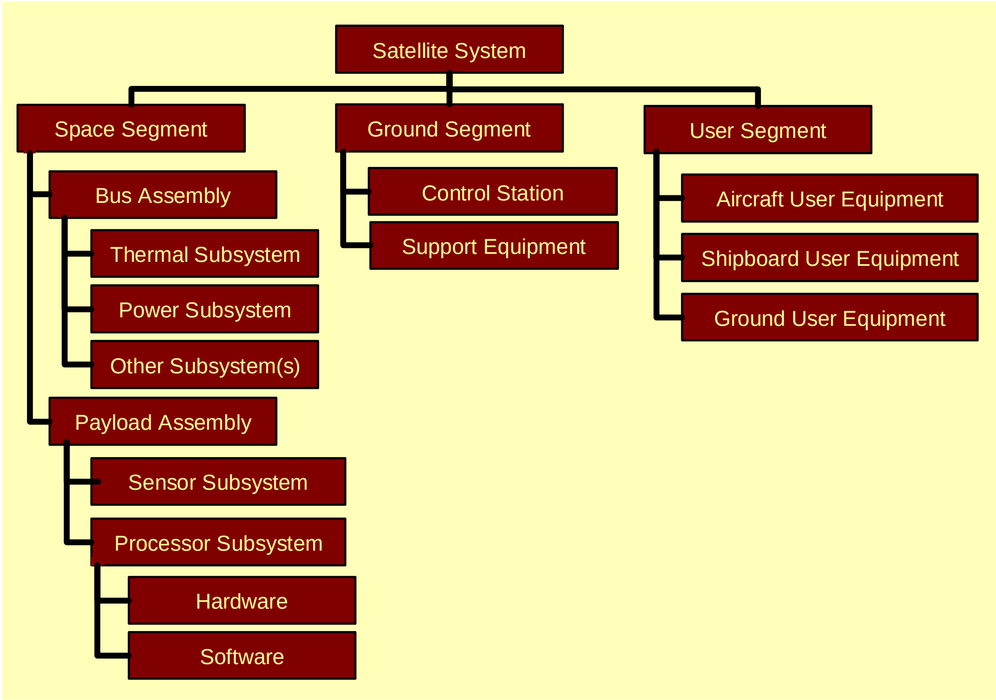


Figure 9. A simple satellite system physical hierarchy (product tree)

Many of the terms in Figure 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 in Figure 4 may define the technical requirements for the system and segment levels.) In the simple example in Figure 9, 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.[18]

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 the 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

18. 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.

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.

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 producibility, reliability, supportability, and the like. 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 which 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 systems engineering 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.

The Design Baseline

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 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 systems engineering process to balance the system design. The documented design forms the design baseline.

Systems Engineering Management

Systems engineering 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.

Completion of Development and the Product Configuration Baseline

When the design is complete, several steps still remain before the capability is realized in operational use. These steps are shown in Figure 10 below.

When the iterative process has produced an approved design baseline, it provides the basis for building (manufacturing), buying (procuring), coding, and subsequently integrating 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 to include such factors as field and depot maintenance equipment 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 then later to complete the planned deployment. For satellites and launch systems, that 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 10, the ultimate objective of systems engineering . . . at least until a deficiency that must be corrected is identified or the system or one of its products is to be modified or upgraded for some reason (such as a change in the threat or the obsolescence of a part) requiring that the process in Figure 10 be iteratively followed once again.

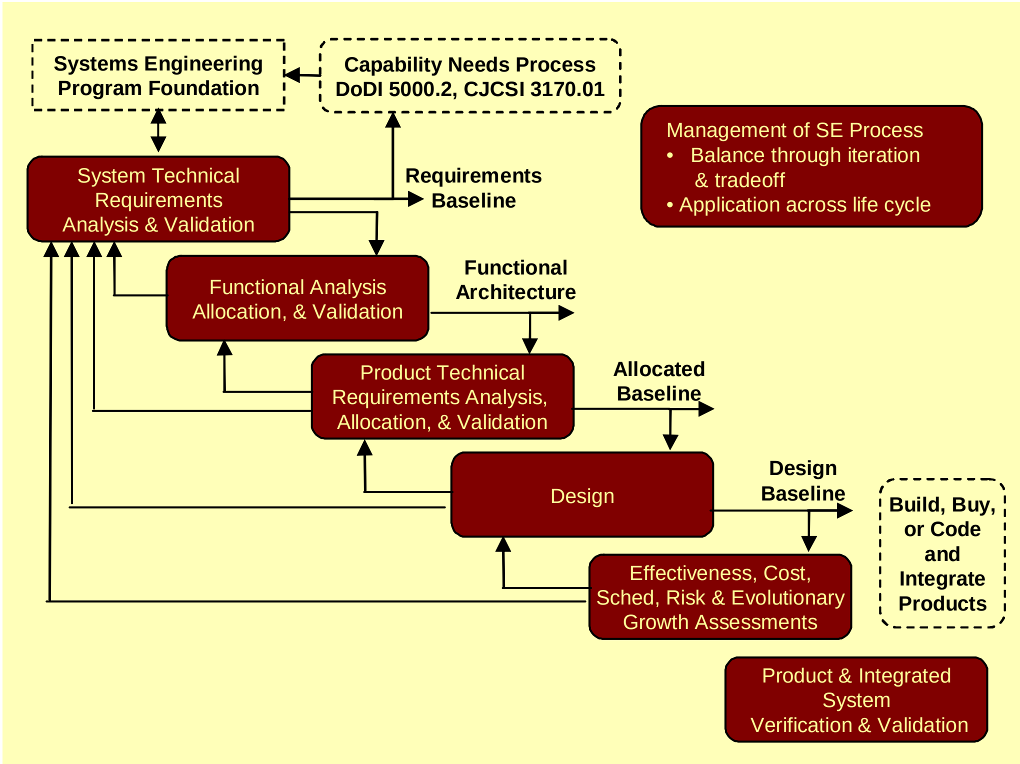


Figure 10. Overall systems engineering process

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.

Configuration Management

As the baseline shown in Figure 10 evolves, experience has shown that specialized management activity traditionally called configuration management or configuration control is beneficial to ensure that the baseline is fully recorded to guide the further development and production and that subsequent changes are made in a way that correct identified deficiencies or responds to new requirements while maintaining the resulting design balanced with respect to effectiveness, cost, schedule, and potential for evolutionary growth. To achieve this, the baseline is said to be 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 6.

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.

Table 1. Reviews and audits objectives

Technical Review or Audit	Objective	DoDI 5000.2 Phase	NSSAP 03-01 Phase
Alternative System Review (ASR)	Concept selection.	Concept Refinement	Pre KDP-A
System Requirements Review (SRR)	Review SE program foundation.		A
System Design Review (SDR)	Approval of the requirements baseline.	Technology Development	B
Preliminary Design Review (PDR)	Approval of the allocated baseline.	System Integration	
Critical Design Review (CDR)	Approval of the design baseline.		
Functional Configuration Review (FCA)	Qualification of the design.	System Demo	C
System Verification Review (SVR)	Readiness for production, training, deployment, ops, support, & disposal.		
Physical Configuration Audit (PCA)	Approval of the product configuration baseline.	Production and Deployment	

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 the following table.

Table 2. Baseline maturity levels expected at technical reviews

Technical Review or Audit	Typical Required Maturity of the Baselines			
	Requirements	Allocated	Design Release	Product Configuration
ASR	Preliminary	Preliminary	Preliminary	–
SRR	Draft	Preliminary	Preliminary	–
SDR	Approved	Draft	Preliminary	–
PDR	Maintained	Approved	Draft	–
CDR	Maintained	Maintained	Approved	–
FCA	Maintained	Maintained	Maintained	–
SVR	Maintained	Maintained	Maintained	–
PCA	Maintained	Maintained	–	Approved

In some programs, some of the reviews in the above table may not be necessary or the purposes of two of the reviews may be merged into one review. In others, 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. See MIL-STD-1521B including Notices 1 and 2 for more detail on the above reviews except for the ASR and SVR. See the draft MIL-STD-499B for more detail on the ASR and

SVR. Finally, in some programs, the program structure may differ from the nominal DoDI 5000.2 and NSSAP 03-01 phases shown in the table or the reviews and audits may be held in phases different from those shown in the table.

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

As used in the table above, a preliminary baseline is one that documents the results of initial iterations through a process similar to the one summarized in Figure 10 above. 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.

What is the Relationship of Systems Engineering to Program Management?

The systems engineering process described above cannot be assigned to a single organizational element of a program office or contractor program organization (such as a systems engineering and/or integration team). Instead, the iterations and associated tradeoffs, assessments, and decisions of the systems engineering process ultimately involve all of the relevant specialty engineers (including those who are called systems engineers) as well as the design engineers for all the products making up the system. Since the activities are pervasive, program management is ultimately responsible for the products of systems engineering, for managing risk, and for managing (controlling) the configuration of the products that make up the system. As a result, it can be useful to consider systems engineering as a cross-product process and the systems engineering organization as a cross-product staff function serving program management. For that reason, in some programs the program director or program manager may choose to retain the chair or leadership of activities such as the risk management or configuration management boards or the interface control working group (especially for external interfaces).

But no matter how the program is organized to manage risk and other matters, systems engineering is inextricably linked to program management. For example, in most programs, a key management tool is the product-oriented work breakdown structure (WBS). The product-oriented WBS starts with the physical hierarchy or product tree developed as described above as part of the iterative systems engineering process. Using the product tree shown in Figure 9 above as a point of departure, a simple, partially populated, product oriented WBS outline is shown below. [19]

19. You may also see another form of the WBS called the functional WBS which is a hierarchical outline of the functional specialties required to complete the program activities. Usually, the definition of the required specialties also flows out of the iterative systems engineering process.

Table 3. A simple satellite systems product-oriented work breakdown structure

Level 1	Level 2	Level 3	Level 4	Level 5
00000 System	1000 Space Segment	1100 Bus Assembly	1110 Thermal Control	
			1120 Power System	
		1200 Payload Assembly	1210 Sensor	
			1220 Sensor Processor	1221 Sensor Processor H/W
				1222 Sensor Processor S/W
	2000 Ground Segment	2100 Ground Control Station		
		2200 Support Equipment		
	3000 User Segment	3100 Aircraft User Equipment		
		3200 Ship User Equipment		
		3300 Ground User Equipment		
	4000 Program Mgt/ Sys Engineering	4100 Program Management		
		4200 System Engineering	4210 Requirements Analysis	
			4220 Functional Analysis	
			4230 System Analysis & Control	
			4240 Synthesis	
			4250 Specialty Engineering	

In the WBS shown above, required work not directly associated with a single product such as program management and systems engineering has been added to that for the products. Each entry has been given a number that can facilitate the creation of related definitions of the work or a Statement of Work (SOW) which defines the scope of the work for each entry in the WBS (down to a selected level) for a development contract. A more detailed product-oriented WBS and the associated definitions are shown in Appendix C2.

The product-oriented WBS is, in turn, used as a starting point for program management to organize and plan for the program. For example, the product-oriented WBS can be used to assign work to the individual elements of the Program Office or contractors' organization. In a program management approach or framework called Integrated Product Development (IPD), the product-oriented WBS maps directly to the organization chart so that the work responsibility of each organizational element is summarized by one or more WBS entries and its associated definitions and SOW. Each of the organizational elements is sometimes called an Integrated Product Team (IPT).[20] IPD and IPTs evolved as a response to "stove-piped" engineering organizations in which producibility or reliability specialists, as examples, might NOT be integrated into the design activity with the result that the associated constraints might be given inadequate attention or "band-aided" on late in the development with a resultant lack of balance in the design.

The product-oriented WBS is also used in the IPD framework as one of the starting points for developing the Integrated Master Plan (IMP) which identifies the significant accomplishments to be completed by each major program event or milestone for each major entry in the WBS (and therefore, by association, for each organizational element or IPT). The events or milestones usually include the technical reviews and audits discussed above. IMPs can be sorted to show the work to be completed for each event and then by WBS or to show the work to be completed for each WBS. IMPs sorted by WBS define the accomplishments to be completed by each IPT or other organizational entity to which a WBS entry is assigned.

The IMP can also include a description of the processes (such as systems engineering) that guide completing the significant accomplishments. Alternatively, such process descriptions can be in other documents such as the Systems Engineering Management Plan (SEMP).

In the IPD framework, the IMP can, in turn, be extended to develop the Integrated Master Schedule (IMS) which shows the tasks needed to achieve each significant accomplishment on a calendar schedule. And finally, the tasks in the IMS can be extended to the work packages in an Earned Value Management System (EVMS) that define the resources planned to complete each task.

To summarize, systems engineering is a process for planning, directing, monitoring, and controlling all the engineering on a program to develop, deploy, and sustain a system. The organizational element that is called systems engineering in a Program Office or at a Contractor's facility can be usefully thought of as a staff function to help the program director or manager orchestrate the program-wide engineering activity to understand and refine the purposes of the program and develop a balanced design to fulfill those purposes. Finally, the physical hierarchy or product tree for the system concept created and refined by the systems engineering process is the starting point for the product-oriented WBS, the IPT structure, the IMP, the IMS, and the EVMS.

How We Got to Where We Are?

A brief history of systems engineering for military programs – standards, reform, non-governmental standards, and SMC's Systems Engineering Revitalization (SER).

---

20. The term Integrated Process and Product Development (IPPD) is sometimes used instead of IPD. Also, the term IPT was initially used circa 1990 on Air Force programs to describe a development team assigned responsibility for specific program products via the product-oriented WBS. It has since been used more widely, in some cases where the products may be less well defined or more ephemeral. It is used here in the sense of its original application.

The years of the Cold War starting in the 1950s brought very complex ballistic missile and space endeavors including the Polaris submarine launched ballistic missile; the Atlas, Titan, and Minuteman intercontinental ballistic missiles; and the Apollo program to place man on the moon. As a result, engineering a system became correspondingly more demanding. Simply planning the engineering and related technical work was challenging. That led to the preparation of detailed schedules that identified the various tasks that must be completed and their interrelationships (which can be complex) to help engineers plan their work initially and then, using updates to the schedules, cope with unanticipated outcomes of the work (which helps make it interesting). This led, in turn, to various methodologies and techniques such as critical path schedules or networks which identify the series of tasks that are predicted to require the most time relative to the desired finish date, i.e., are the most critical. In some engineering organizations, schedule planning, monitoring, and maintenance became a primary focus of the systems engineers. Other organizations saw that the design task had to be divided up because either the engineering skills for one part of the design (propulsion, for example) were different from another (communications, for example) or because the scope of the design task was otherwise unmanageable. In some cases, it was immediately recognized that there was the need to formally allocate the system level requirements to each of the responsible design teams. Still other organizations came belatedly to this conclusion when interface or performance shortfall issues were subsequently found during integration, significantly delaying the introduction of the system and adding significantly to its cost (and often leading to its early retirement and replacement by the next generation which was even more costly). Thus, as the systems became more complex, engineering managers have evolved systems and other engineering specialties to meet the demand.

One approach to capture the lessons learned as military systems became more complex was through the development and publication of specifications and standards. Military specifications and standards became one of the primary approaches to capture lessons learned. Some specs and standards sought to prevent the recurrence of some types of failures by defining certain types of technical requirements such as design and test margins (which, since space operations are so unforgiving, are critical to engineering space systems), for fuel or lubricant characteristics. Others were developed to specify processes for systems engineering, configuration management, reliability management, and so on – these latter process standards became know as how-to requirements. As they evolved, the number of military specs and standards became very large. Moreover, most of the documents referred to a number of others so that it was sometimes difficult to understand all the requirements on a contract. In addition, the imposition of military specs and standards often precluded the use of commercial hardware which was often more advanced and lower in cost after the explosion of commercial computation and other information technologies in the late 1980s and '90s.

So military specifications and standards, though they addressed real problems, grew into a monster in which in acquisition contracts might impose, directly or by reference, tens to hundreds of military standards which were often very prescriptive including "how-to" requirements to the point that it was judged that they unnecessarily increased the cost, that no one understood or could enforce the totality of the requirements on many contracts, and that the use of the growing commercial technologies was severely impeded.

In the early to mid 1990s, as a result of the growing problems with military specs and standards, DoD sought to restrict the use of such documents. The resulting effort, which became known as the Military Specifications and Standards Reform Program (MSSRP) soon grew into a wider acquisition reform initiative. However, acquisition reform had unfortunate unintended consequences. The SMC Systems Engineering Revitalization effort was established to deal with those consequences. As noted in a recent policy letter from the SMC Commander, "One key element of acquisition reform was to eliminate . . . contractually dictated prescriptive "how-to" instructions or processes used by contractors. For a decade, we have limited and reduced our use of specifications and standards in RFPs, proposal evaluations, contractor performance assessments, and on contracts as compliance documents. The unintentional result was that technical baselines and processes were compromised. With the turnover, consolidations, and retirement of many industry and government personnel, we have hampered our ability to pass on lessons learned from generation to generation."[21]

As a result, a key element of the Systems Engineering Revitalization effort is the use of specifications and standards as part of the technical baseline of the SMC acquisition process. There is no intent to return to the pre-acquisition reform approach of using an excessive number of specifications and standards and prescribing detailed processes. A list of high-priority critical specifications and standards is being reviewed and established for appropriate use in the acquisition process." Many of the specifications and standards selected for the SMC technical baseline have been tailored and in some cases may be updated or revised. All should be reviewed and further tailored as necessary to bring them in line with the objectives of each contractual action. "Tailored specifications and standards and contractor responses must be relevant and hold members of the government industrial partnership appropriately accountable to sound technical disciplines. They should be used in new acquisitions and can be used on legacy programs to modify contracts if benefits can be shown to warrant the changes. They will be used in a less prescriptive manner than in the past. For example, the contractor may propose the listed specification/standard or another government, industry, technical society, international or company version provided it is comparable in vigor and effectiveness. Proof of this comparability must be provided. Acceptable responses will be put on contract as compliance documents with follow-up at program reviews to ensure they are appropriately addressed."

Systems Engineering Revitalization also includes other initiatives: strategic planning, a web site to support the application of systems engineering on SMC programs, and this Primer and the associated Handbook.

Why is Systems Engineering Useful?

If the system is sufficiently simple that the engineering design can be completed out by a single, experienced team that can maintain communications with the customers for the new capability, then a formal systems engineering process and systems engineering organization may be unnecessary. But most systems, especially most military systems, are more complex than that. For example, the development team cannot practically stay in direct contact with the operational forces that have the need for the years or even decades it can take to develop, produce, and sustain the system. As a result, a formal definition of the needed capability is required. So that program decision makers can allocate limited resources to fulfilling such needs wisely, a concept that can be the basis for satisfying the need must be defined and the associated effectiveness, cost, schedule, risk, and evolutionary growth of must be assessed. That, in turn, means that the need and related constraints must be translated into unambiguous, consistent, and complete terms against which the concept can be assessed, i.e., into the formal requirements baseline. And since the work can require many specialized industrial design teams involving perhaps hundreds or even thousands of engineers at a number of prime and subcontractor facilities, there is a need to allocate the system level requirements to each design team – that leads to the need for the functional architecture and allocated baseline.

---

21. The quotations here and in the next paragraph are from the Policy Letter on Specification and Standards Usage at SMC, SMC/CC, Jan. 14, 2003.

As systems engineering has evolved, it has taken on increasing importance through other specific tasks or products that the process integrates into the overall engineering effort to obtain a balanced design. These include:

- the integration of engineering specialists who are experts in reliability, producibility, and other these areas into the requirements analysis, allocation, and design activities, especially to meet the extremes of the space environment and its need for high reliability design approaches (such as redundancy) and parts, materials, and processes selection, application, and control,
- the definition and control of the external interfaces between the system and other systems or facilities and between elements in the system (internal interfaces), and
- the definition of and adherence to adequate design margins, and
- a reemphasis on system products (through IPD) after the emergence of specialty engineering stovepipes threatened the needed balance in the system design through inadequate (or delayed) emphasis on some requirements or an over emphasis on other requirements.

Chapter 2
How Does the Systems Engineering Process Work?

In the previous chapter we described the systems engineering process as it is commonly defined for military systems. In this chapter, we expand on key constituents of the process such as requirements analysis, functional analysis & allocation, synthesis, and system analysis & control and explain how it works.

The Systems Engineering process is a series of repetitive operations whereby a universe of possible solutions to a stated need are narrowed to a single system that optimally satisfies the need. It is a continual excursion between the general and the specific… top down, bottom up… to propose solutions, check their possible implementation, and then propose new or modified solutions to be checked again. Even the most talented Systems Engineer 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 DoD 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 can be met much sooner and at lower cost. There is no need to reinvent the wheel!

The systems engineering process starts with the identification of the needed capabilities. First, it would be beneficial to understand how the needed capabilities are determined as the inputs to the systems engineering process are the outputs of the capability needs process.

The Capability Needs Process

The Chairman of the Joint Chiefs Of Staff Instruction, JCSI 3170.01C, 24 June 2003 establishes the policies and procedures of the Joint Capabilities Integration and Development System (JCIDS). JCIDS implements a capabilities-based approach that better leverages the expertise of all government agencies, industry and academia to identify improvements to existing capabilities and to develop new warfighting capabilities. The top down capability need process is provided in Figure 11. This approach requires a collaborative process that utilizes joint concepts and integrated architectures to identify prioritized capability gaps and integrated DOTMLPF solutions (materiel and nonmateriel) to resolve those gaps.

As joint concepts and integrated architectures are developed, a capabilities identification methodology emerges that flows from top-level strategic guidance. Based on this guidance, the Joint Operations Concepts (JOpsC) portrays the linkage between how the joint force operates today and the vision for the future. Supporting Joint Operating Concepts (JOC) and Joint Functional Concepts (JFC) provide the foundation from which integrated architectures are developed and refined. As they are developed, the integrated architectures provide the construct for analysis to identify capability and supportability shortfalls, compare alternatives for improving joint warfighting capabilities, and associated resource implications. For more details on this process, refer to the JCSI 3170.01C.

A brief description of Functional Area Analysis (FAA), Functional Needs Analysis (FNA), and Functional Solution Analysis (FSA) follows.

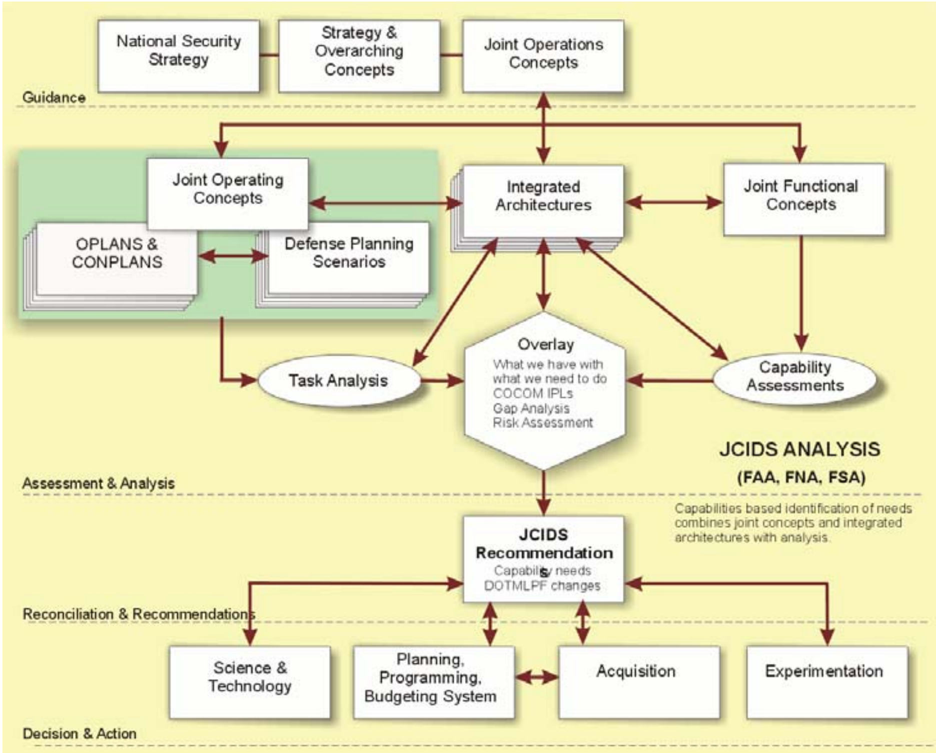


Figure 11. JCIDS top-down capability need identification process

Functional Area Analysis

Functional Area Analysis (FAA) identifies the operational tasks, conditions, and standards needed to achieve military objectives. It uses the national strategies, JOCs, JFCs, integrated architectures and the Universal Joint Task List (UJTL) as input. Its output is the tasks to be reviewed in the follow-on functional needs. The FAA includes cross-capability analysis and cross-system analysis in identifying the operational tasks, conditions and standards.

Functional Needs Analysis

Functional Needs Analysis (FNA) assesses the ability of the current and programmed joint capabilities to accomplish the tasks that the FAA identified under the full range of operating conditions and to the designated standards. Using the tasks identified in the FAA as primary input, the FNA produces as output a list of capability gaps or shortcomings that require solutions and indicates the time frame in which those solutions are needed. It may also identify redundancies in capabilities that reflect inefficiencies. The FNA includes supportability as an inherent part of defining capability needs.

Functional Solution Analysis

Functional Solution Analysis (FSA) is an operationally based assessment of all potential DOTMLPF (doctrine, organization, training, materiel, leadership and education, personnel and facilities) approaches to solving (or mitigating) one or more of the capability gaps (needs) previously identified. On the basis of the capability needs, potential solutions are identified, including (in order of priority) integrated DOTMLPF changes that leverage existing materiel capabilities; product improvements to existing materiel or facilities; adoption of interagency or foreign materiel solutions; and finally, initiation of new materiel programs. Identified capability needs or redundancies (excess to the need) establish the basis for developing materiel approaches in ICD and/or DOTMLPF approaches.

JCIDS Process and Document Descriptions

A simplified depiction of the relationship between the JCIDS process and key acquisition decision points is provided in Figure 12 below.

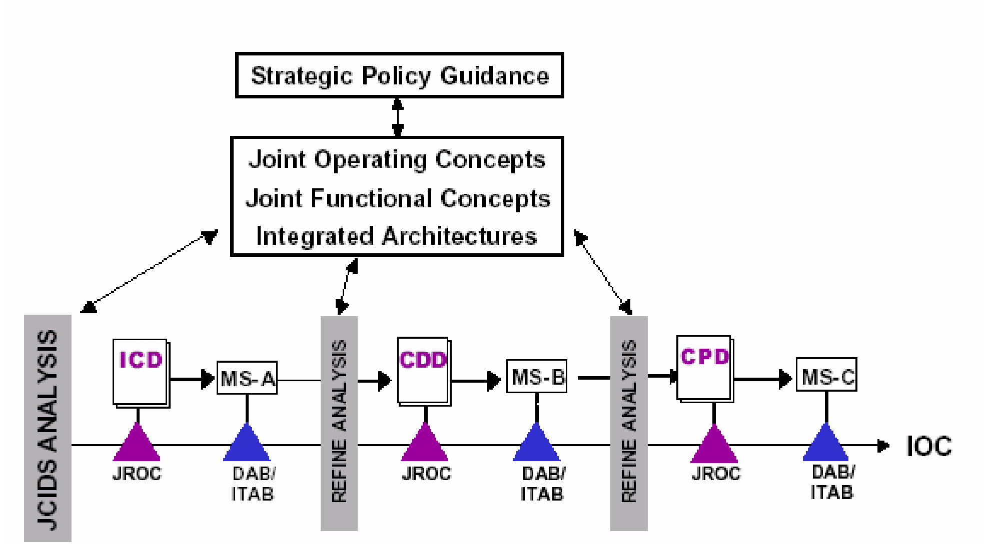


Figure 12. JCIDS process and acquisition decisions

JCIDS Document Descriptions

Services and other DOD Components may develop ideas and concepts leading to draft ICDs, CDDs, CPDs and CRDs (when CRDs are directed by the JROC). An ICD is generated in all cases to define the capability in a joint context, review the options to provide the capability, and ensure that all DOTMLPF alternatives, impacts and constraints have been adequately considered. All initiatives transitioning to the acquisition process will have a corresponding validated and approved CDD and/or CPD prior to entering Milestone B or C, respectively. Brief descriptions of the documents are provided below. For more information, refer to the POH Primer, DoD & AF Systems Acquisition, JCIDS at www.smc.sparta.com/golive/site16.

Initial Capabilities Document (ICD)

The ICD makes the case to establish the need for a materiel approach to resolve a specific capability gap derived from the JCIDS analysis process. The ICD supports the analysis of alternatives (AoA) (for ACAT I/IA programs), the Technology Development Strategy, the Milestone A acquisition decision, and subsequent Technology Development phase activities as described in reference e. The ICD defines the capability gap in terms of the functional area(s), the relevant range of military operations, time, obstacles to overcome and key attributes with appropriate measures of effectiveness, e.g., distance, effect (including scale), etc. ICDs will eventually be based entirely on integrated architectures.

The ICD also captures the evaluation of different materiel approaches that were proposed to provide the required capability. The ICD proposes the recommended materiel approach(s) based on analysis of the relative cost, efficacy, sustainability, environmental quality impacts and risk posed by the materiel approach(s) under consideration. The analysis that supports the ICD is the beginning of the Analysis of Alternatives (AoA) that will be used through the life of the system.

The ICD describes how the recommended approach best satisfies the desired joint capability. It supports the AoA by providing operational context for assessing the performance characteristics of alternatives. Once approved, an ICD is not normally updated. When approved, CDDs (described below) bring the desired capability specified in the ICD into the System Development and Demonstration (SDD) phase, and the ICD is archived for reference. The CDD then serves as the living document to carry contributing systems and subsequent increments through the SDD phase.

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 a proposed program(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 will have its own set of attributes and associated performance values with thresholds and objectives established by the sponsor with input from the user. The CDD supports the Milestone B acquisition decision.

The CDD provides the operational performance attributes, including supportability, necessary for the acquisition community to design the proposed system, including key performance parameters (KPP) that will 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 shall 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.

Capability Production Document (CPD)

The CPD addresses the production attributes and quantities specific to a single increment of an acquisition program. When the CPD is part of an FoS/SoS (family of Systems, System of Systems) solution, the CPD will reference the originating ICD and provide the linkages to related CDDs/CPDs and supporting analyses (e.g., AoA) to ensure the system production is synchronized with the related systems required to fully realize the capability(s). The sponsor finalizes a CPD after critical design review when projected capabilities of the increment in development have been specified with more accuracy. The CPD must be validated and approved before the Milestone C decision review.

Performance and supportability attributes in the CPD will be specific to the increment. The design trades from the SDD phase will have been completed and a specific production design determined for the increment. The threshold and objective performance values of the CDD are, therefore, superseded by the specific production values detailed in the CPD for the increment.

Reduction in threshold KPP performance will require an assessment of the military utility of the reduced capability and, possibly, a reexamination of the program to determine if an alternative materiel or nonmateriel solution should be adopted.

Capstone Requirements Document (CRD)

The JROC may approve the development of a new CRD when existing concepts and integrated architectures are not sufficient to support development of capabilities. As joint concepts and integrated architectures are developed, straight-forward CRDs that are a clear statement of the military task that must be accomplished will continue to induce the development of interoperable capabilities by describing overarching thresholds/goals and standards in functional areas, especially where an FoS or SoS approach is required. In general, the existence of an approved integrated architecture will obviate the need for a CRD.

The Systems Engineering Process

The systems engineering process as defined in Military Standard 499B is shown schematically in Figure 13. This representation is the most commonly used on DoD programs. The reader may want to refer to the systems engineering process, Figure 10 of Chapter 1, to correlate the evolutions of technical baselines with the constituents of this model.

The customer's/User's needs, objectives and requirements in terms of capabilities, measures of effectiveness, environments, and constraints initiate the process. Each increment of capability is provided with its own set of attributes and associated performance values. Measures of effectiveness 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. Customer-imposed constraints may take the form of interoperability with existing or other planned systems, operations and maintenance personnel skill level requirements, and costs and schedules.

The technology base and prior development efforts are natural inputs to the process. Any good Systems Engineer builds on what has been done before. However, in analyzing existing technology for use on the current program, the System Engineer must identify critical areas where proof of the use of the technology in the given application is required. This may indicate the need for additional research.

The major constituents of the Systems Engineering 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 fed back to the requirements analysis to verify their compliance or to determine whether modification of the requirements is compatible with achieving the mission.

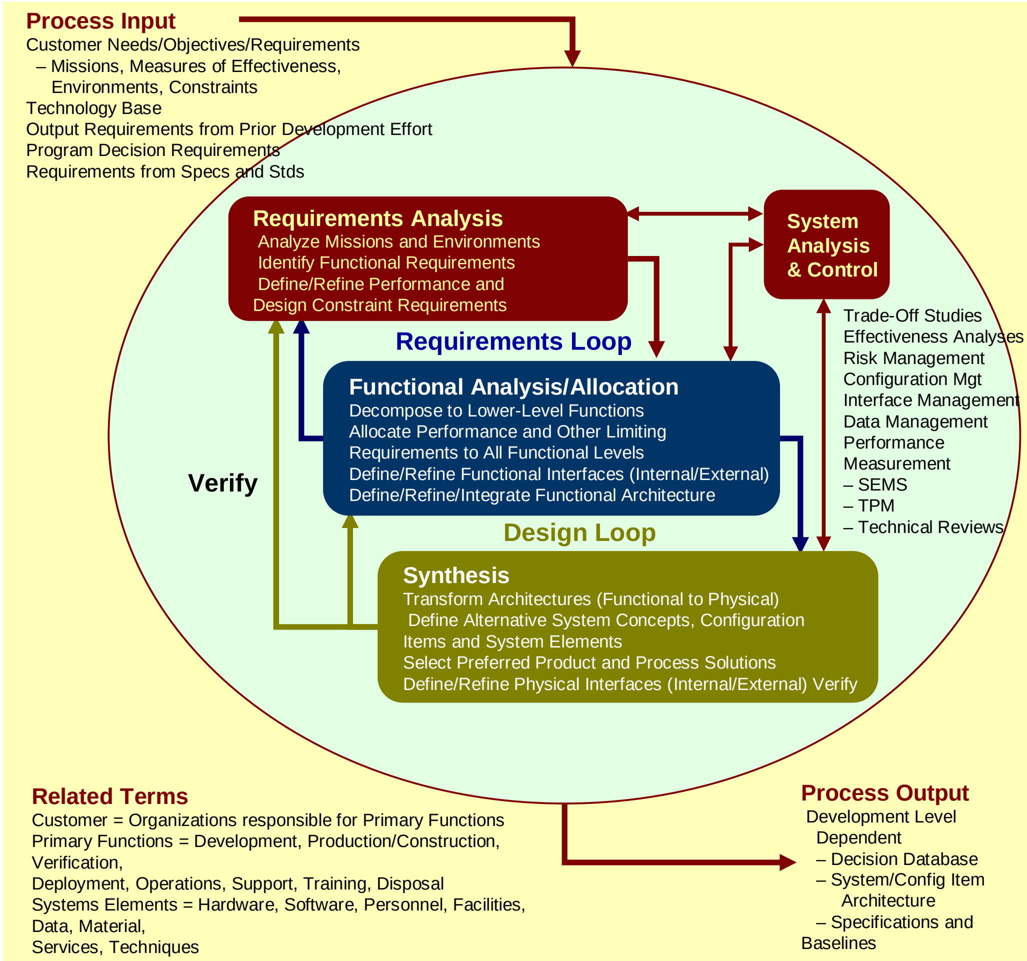


Figure 13. The systems engineering process

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 designs, 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 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 System Engineering Process includes a decision database and a balanced system solution.

The database documents include:

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, we will look in more detail at the efforts involved in the four basic activities of the System Engineering Process. Sub-activities are identified to aid the discussion and to highlight specific efforts and outputs. However, they are not meant as isolated operations that toss their outputs "over the wall" for someone else to process further. Such activities are highly interactive and often performed by the same Systems Engineer. In fact, because they are usually so closely connected, at any given time the System Engineer may have difficulty determining on which he is working. This does not vitiate their value as discussion points and, on large programs they may, in fact, be conducted as separate operations.

Requirements Analysis

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. (The insert in the upper right of Figure 14 shows the relationship of Requirements Analysis to the other Systems Engineering activities previously presented in Figure 13.) 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 14. The Missions and Environments Analysis firms the customers 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 which 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.

Detailed descriptions of the activities of the Requirements Analysis are provided in the Figure 14 below.

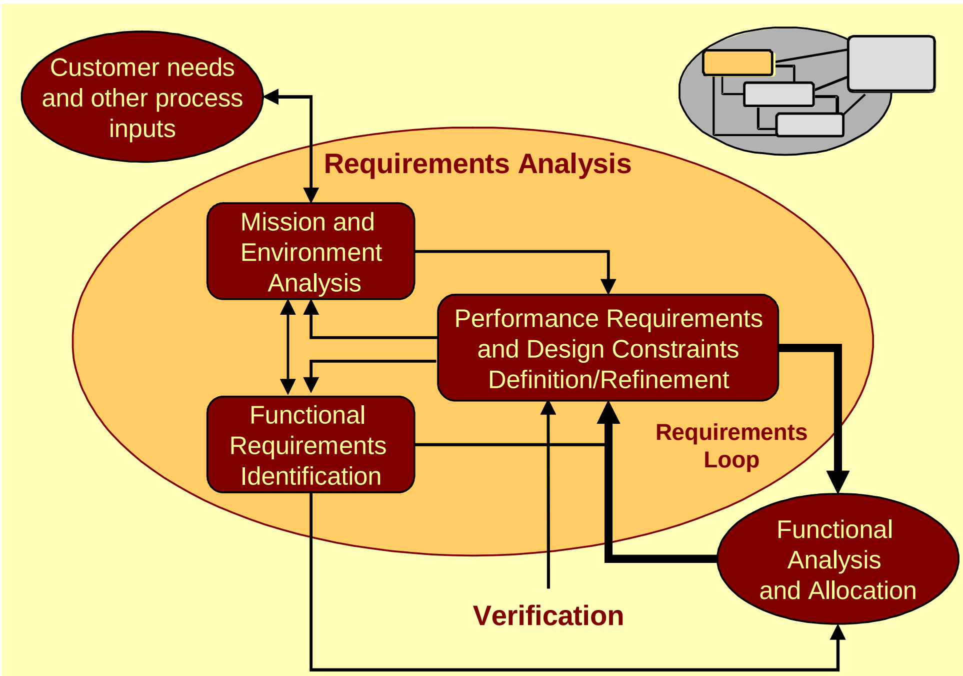


Figure 14. Requirement analysis–converting customer needs into system requirements

Missions and Environments 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? 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 System Engineer has an obligation to provide all the information necessary to make that decision.

Functional Requirements Identification

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 microradians 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 microradians 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.

Performance Requirements and Design Constraints Definition and Refinement

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.

Functional Analysis and Allocation

The Functional Analysis and Allocation bridges the gap between the high level set of system requirements and constraints (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 (See insert at top right of Figure 15.) 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 the both the Requirements and Design Loops.

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 architecture(s) are developed. These activities track and interact so that as details evolve, they are continually validated against each other. Should anomalies occur — for example, GPS user equipment signal processing might require greater receiving sensitivity — or should a different decomposition appear more advantageous — say detection might be more easily accomplished with increased processing rather than greater signal strength, then re-evaluation of the driving requirements might be undertaken. 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.

Detailed descriptions of the activities of the Functional Analysis and Allocation are provided below.

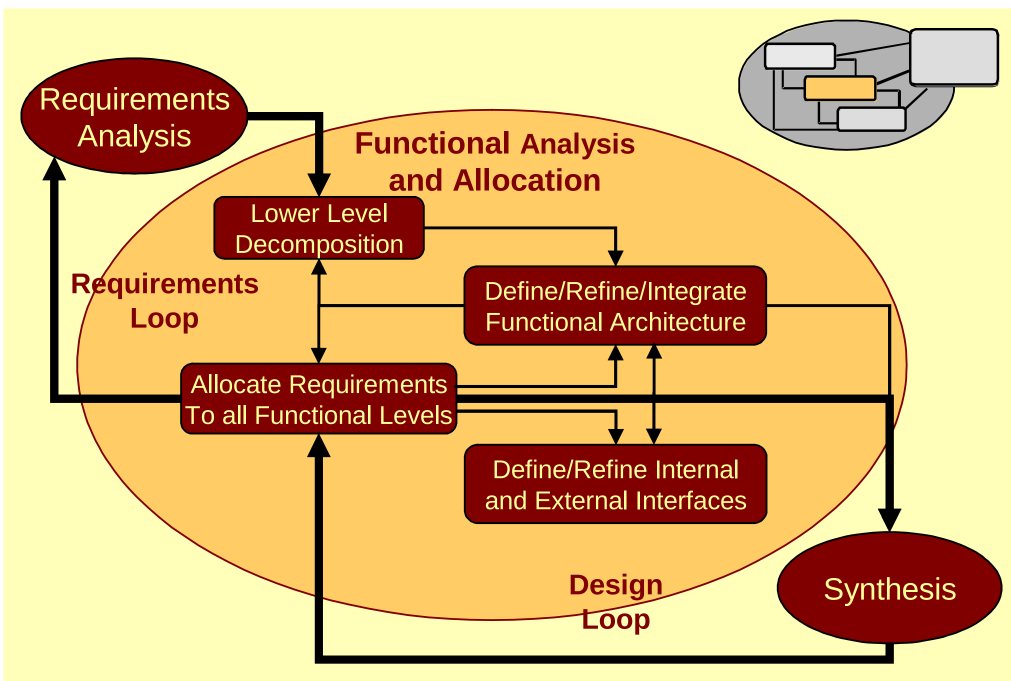


Figure 15. Functional analysis & allocations–create lower level requirements to aid synthesis of solutions

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.

Allocation

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 that 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 System 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 requirement. Design constraints defined in the Requirements Analysis must also be flowed 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.

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 its set of associated 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 Space Network to provide some capabilities for our system under consideration. Figure 16 provides a top level notional view of integration of system functions with the NSN. 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 subfunctions. This leads to the definition of the internal and external interfaces.

Interfaces

System interfaces are both physical and functional. Interface definition and control are two Systems Engineering 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 CDDs, CRDs, 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 tightly coupled to the to system architectures. This one-for-one correlation initiates the interface architecture that is triggered by and traceable to identified 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. For further discussion on interface standards, see Chapter 1 -- Architecture Standards. 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.

Synthesis

Synthesis is the process whereby the functional architectures and their associated requirements are translated 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

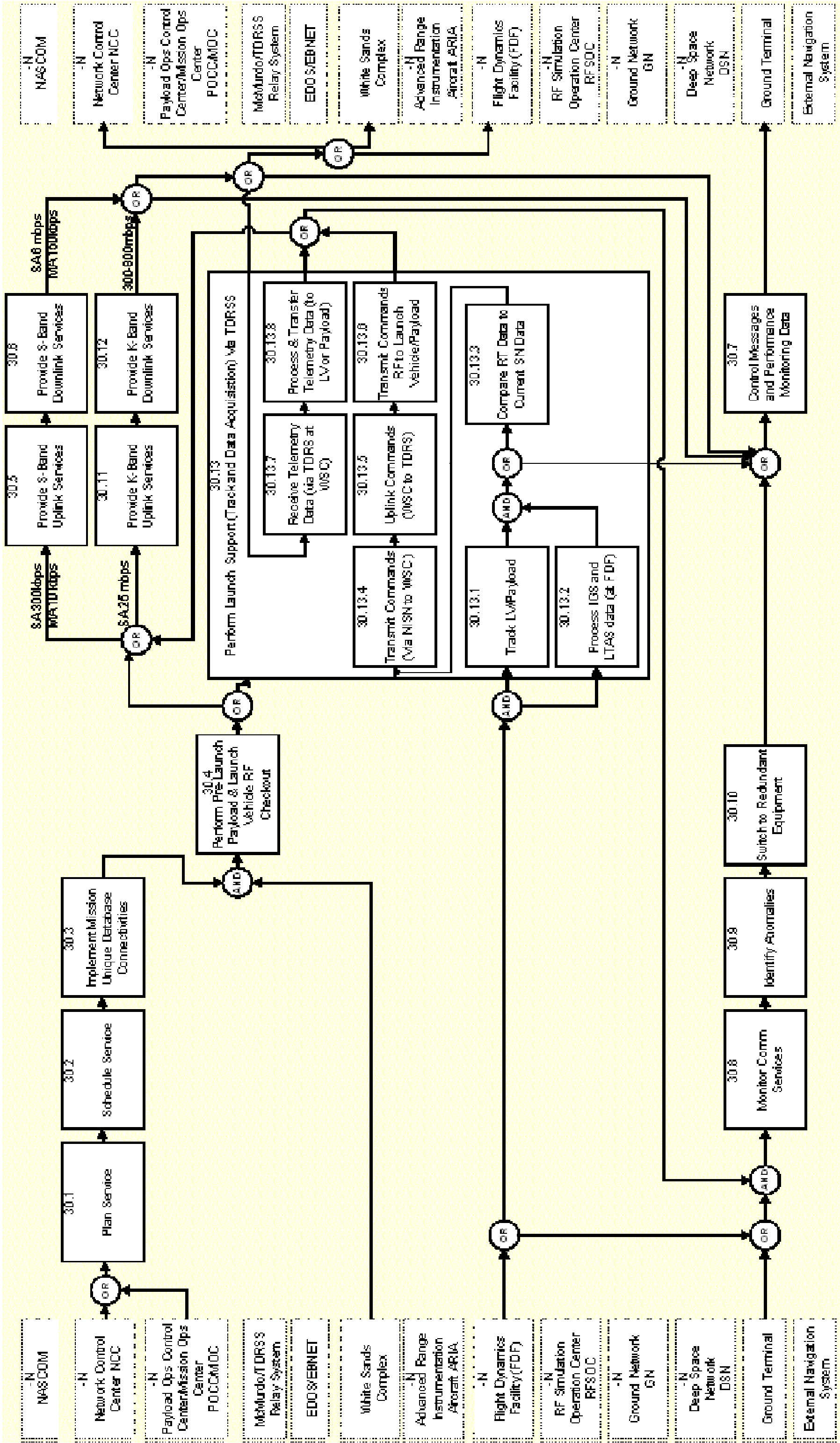


Figure 16. Notional functional architecture–integration of space systems functions with existing space assets

which describe a balanced, integrated system meeting the requirements, and a database which documents the process and rationale used to establish these specifications. The first step of Synthesis (Figure 17) 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 Systems Engineering Process product output. Detailed descriptions of the activities of Synthesis are provided below.

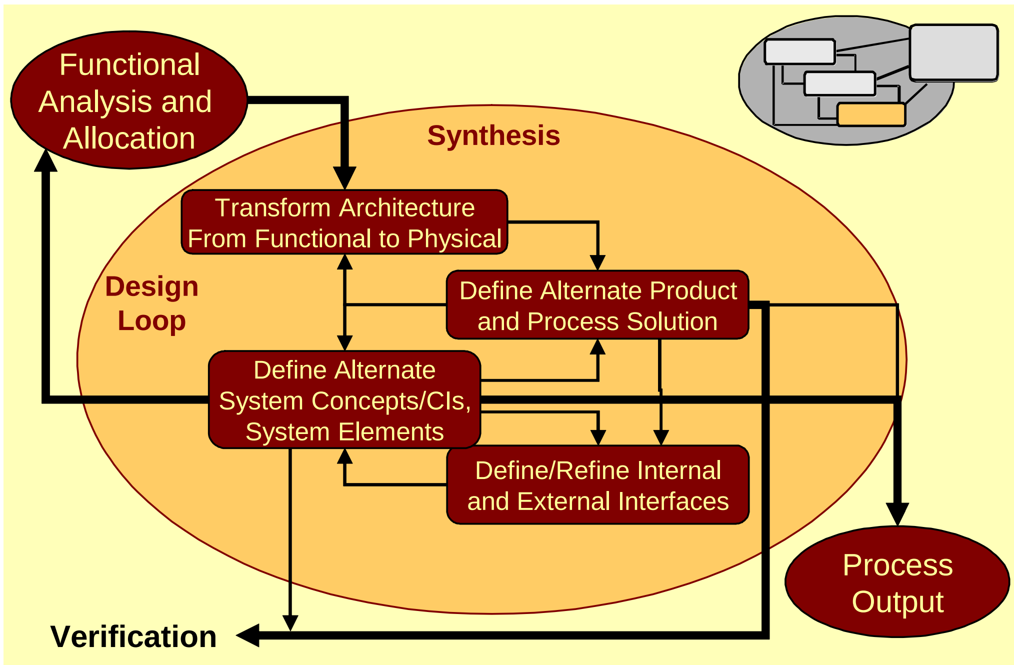


Figure 17. Synthesis–developing detailed solutions

Architecture Transformation

Until this point, the emphasis has been on identification of functions with lesser consideration of how they may be implemented. 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 "tweaking " 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 exiting or prospective related systems. The flexibility provided by open systems architecture during all phases of system development, recommends its consideration in making all Systems Engineering decisions.

C4ISR Architecture Framework

The principal objective of the C4ISR architecture framework is to define a coordinated approach for DoD architecture development, integration, and presentation. The framework is intended to ensure that architecture descriptions can be compared and relate across organizational and system boundaries. In February 1998, the DoD Architectural Coordination Council mandated the use of this framework for all C4ISR architecture descriptions. It behooves the architectural system engineer to understand this methodology.

The framework prescribes three views of an architecture: operational view, system view, and technical view. The operational view is a description of tasks and activities operational nodes, and informational exchange between nodes. The system view is a graphical and textual description of systems and interconnections used to satisfy operational needs. The technical view is the minimum set of rules governing the arrangement, interaction, and interdependence of system parts and elements. Refer to Chapter 3 and Appendix C-11 for more detailed discussion on the C4ISR subject.

Through iterations of the Design Loop, some architectures are discarded because they do not satisfy the requirements, because others satisfy them more completely, or because they are solutions that differ only slightly or offer little advantage over others. For those few that survive, the architectures are used to derive/refine work breakdown structures (WBSs), specification trees, specifications, and configuration baselines that define the system and the effort needed to develop it. For the verified design(s), these defining documents become part of the Process Output database.

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 "black boxes" which will be the subsystems and components that will be developed to perform the system functions. Functions might be distributed among several black boxes. Likewise, there may be several ways in which the boundaries of each box are defined, i.e., 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. As they are developed, 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 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.

Physical Interfaces Definition

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.

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. Much of the output of system concept definition activity is fodder for the cost/benefit and risk analyses performed as part of the System Analysis and Control (Figure 19).

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, Systems Engineering 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.

Work Breakdown Structure (WBS)

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 systems engineering 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 systems engineering 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. Program offices develop a program WBS tailoring the guidance provided in MIL-HDBK-881. The WBS is also an integral part of preparation of the Cost Analysis Requirements Description (CARD). A sample WBS of a launch system is shown in Figure 18. 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 established in Military Handbook 881. 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 systems engineering develops the first three levels of the program WBS, and to provide contractors with guidance for lower-level WBS development. As with most standards and handbooks, use of MIL-HDBK-881 cannot be specified as a contract requirement. Though WBS development is a systems engineering 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. Chapter 5 What is Systems Engineering Management, further addresses the WBS as a means to organize system development activities based on system and product decompositions.

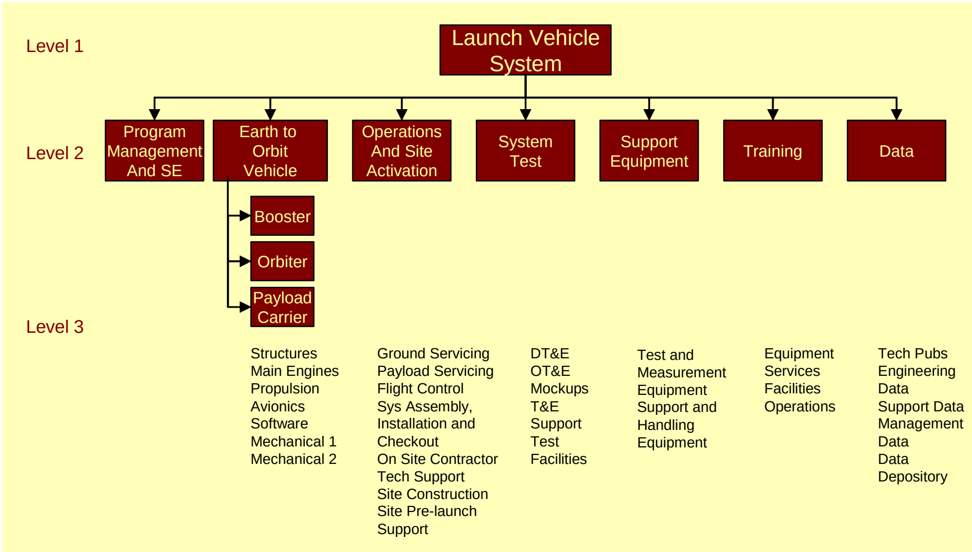


Figure 18. A sample WBS of a launch system

System Analysis and Control

System Analysis and Control is the welding that holds all the other Systems Engineering Process activities together, the steering wheel that gives them direction, and the map that shows where the process is going and where it has been. It is the activity that spans the whole life of the program. It involves the initial analysis of system requirements to prepare the work views discussed in Chapter 1, the management of the activities shown in those views and their interactions, the review and measurement of work progress, and the documentation of work actions and results.

System Analysis and Control (Figure 19) interacts with all the other activities of the Systems Engineering 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 systems engineering entries in the Integrated Master Plan (IMP) which define the overall systems engineering 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.[22] Once the SEMP, IMP, and IMS are in place, the control and manage activity shown in Figure 19 directs their accomplishment.

22. 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.

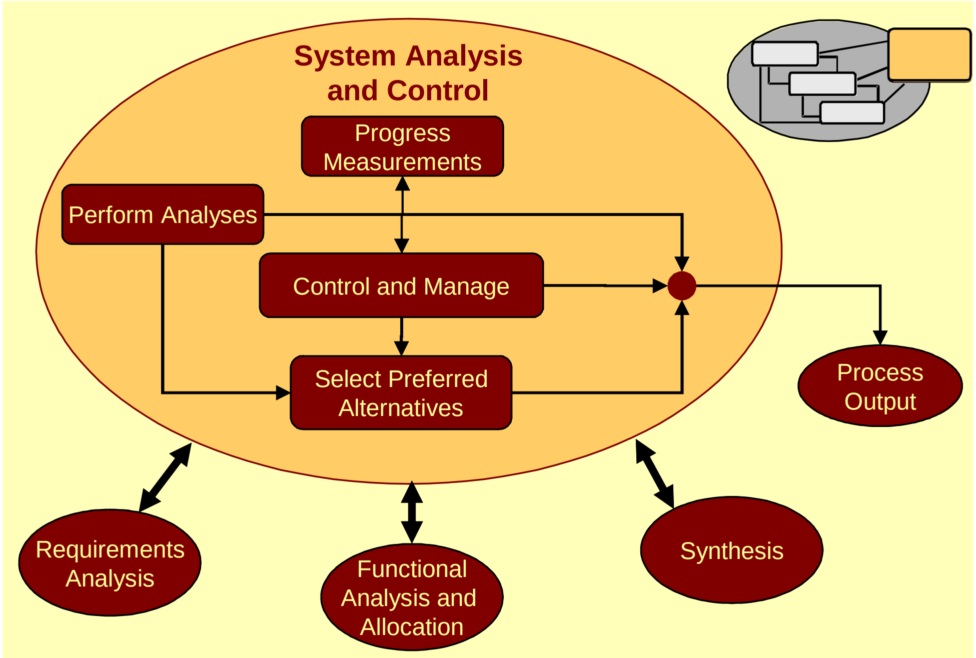


Figure 19. System analysis & control

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.

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 systems engineering 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.

Trade Studies and Analyses

Initial analyses identify the salient factors of the program and its requirements providing the basis for planning the Systems Engineering effort. Subsequent analyses support the selection and refining operations of the other activities of the Systems Engineering 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 Systems Engineering Process advances from Requirements Analysis through Synthesis, the analyses become more detailed.

The trade-off studies supporting the other System Engineering activities are as follows:

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.

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.

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.

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 Systems Engineering 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.

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 systems engineering entries in 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, What is Systems Engineering Management?

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. In support of Synthesis, the selection revolves around selection of physical systems architectures, product and process specifications, and determinations as to which technologies will be used initially to prove concepts and which will be inserted later as technology evolves and designs mature.

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.

A Few Words About Time

The process described above 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 5000.2 and NSSA Acquisition Policy 03-01 provides acquisition models used in developing DoD systems. In Chapter 3 we relate these models to the Systems Engineering functions of documentation, baselining, and review/audit, and to the requirements documents driving these functions. The Acquisition models may undergo further revision. Hence, the text boxes provide practices, products, reviews in the context of the interim acquisition frameworks. In addition, program decision points are intended to impose interim checks of 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.

For these reasons, specialty disciplines are highly concerned with the way interfaces are drawn and specified. Interface requirements are incorporated into the functional architectures used by the Synthesis activity.

Chapter 3
Life Cycle Phases of a Major System

Department of Defense Directive 5000.1[23], states that acquisition programs shall be managed through the application of a systems engineering approach that optimizes total system performance and minimizes total ownership costs. Department of Defense Directive 5000.2, further states that effective sustainment of weapon systems begins with the design and development of reliable and maintainable systems through the continuous application of a robust systems engineering methodology.

Hence, systems engineering must be applied over the life of the program. In Chapters 1 and 2 we described the systems engineering process, expanded on key constituents of the process, and explained how the process works. In this Chapter, we highlight the latest life cycle model for developing a system and associate engineering practices that typically apply for each phase. Figure 20 below depicts the Defense Acquisition Management Framework of DoDI 5000.2. This framework identifies 5 phases: Concept Refinement, Technology Development, System Development And Demonstration, Production And Deployment, Operations And Support.

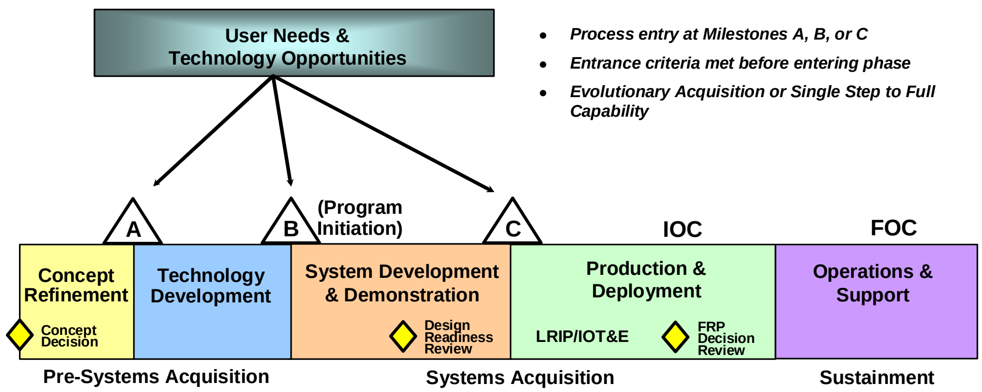


Figure 20. The defense acquisition management framework

The NSS Acquisition Policy 03-01 on the other hand, defines a Pre-KDP-A, a Study Phase (A), Design Phase (B), and a Build, Test, Launch Phase (C). This SMC Systems Engineering Handbook attempts to closely follow both the NSS acquisition policy 03-01 and DoDI 5000.2 frameworks. At the beginning of each section that introduces a new phase the NSS AP phrase terminology followed by the DODI 5000.2 terminology in parentheses. The National Security Space (NSS) Acquisition Process[24] is tailorable and includes acquisition phases, acquisition decision points based on program maturity with focused program assessments, and periodic reports and reviews. See Figure 21. For more information, refer to the POH: Primer, Acquisition Process, Phases of Acquisition at www.smc.sparta.com/golive/site16.

23. DoD Directive 5000.1, May 12, 2003, Operation of the Defense Acquisition System

24 National Security Space Acquisition Policy, Number 03-01, Version 3.0, July 28, 2003. The NSS Acquisition Process is a streamlined, tailorable method for the DoD Space MDA to use in the executive management and oversight of the DoD space programs under his authority.

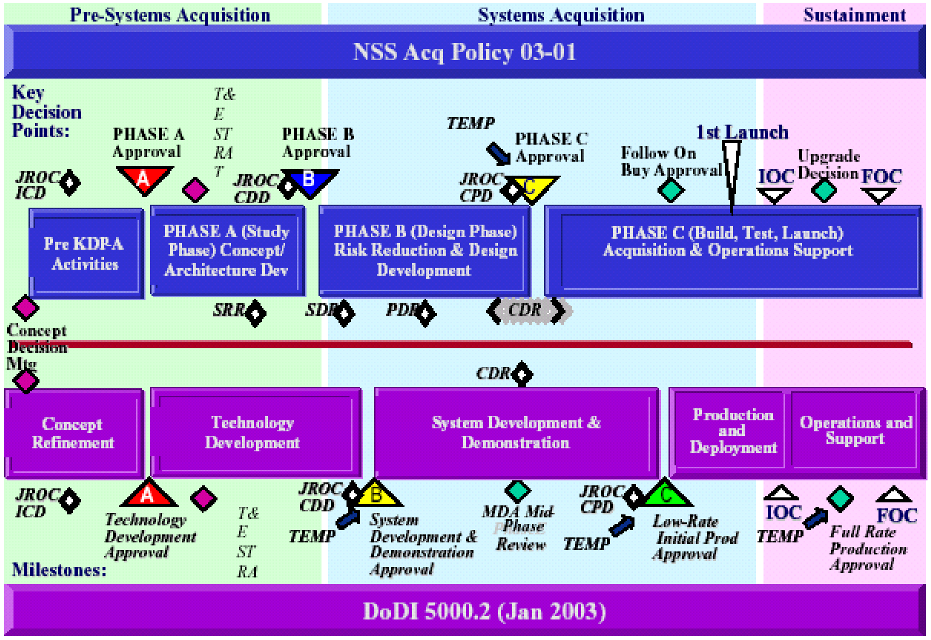


Figure 21. NSS acquisition phases

Pre-KDP-A (Concept Refinement)

Entrance into this phase depends upon an approved ICD resulting from the analysis of potential concepts across the DoD Components, international systems from Allies, and cooperative opportunities; and an approved plan for conducting an analysis of alternatives (AoA) for the selected concept, documented in the approved ICD.[25]

The ICD and the AoA plan guide the Concept Refinement. The focus of the AoA is to refine the selected concept documented in the approved ICD. The AoA assesses critical technologies associated with these concepts, including technology maturity, technical risk, and, if necessary, technology maturation and demonstration needs.

Pre-KDP-A (Concept Refinement)

Goal: Refine the initial concept and develop a Technology Development Strategy (TDS)

Common Practices and Products:

Prepare Integrated Architectures

Joint Concepts -- Develop System Level Conops

Conclude Analysis of Alternatives

Conclude Technology Dev Strategy (TDS)

- Rationale for adopting strategy type
- Program cost, schedule, and performance goals
- Specific cost, schedule, and performance goals
- Test plan to ensure that goals & exit criteria for first technology spiral demonstration are met.

Reviews:

Mission Concept Review/Concept Decision Meet

JROC ICD Review

25 DoDI 5000.2, Para 3.5., Concept Refinement

Phase A–Study Phase (Technology Development)

The Study Phase further examines the feasibility and desirability of a suggested new major system before seeking significant funding.

The NSS Acquisition Policy 032-01 provides the following instruction for Phase A. The activities of this phase typically include concept studies, system architecture development, technology maturity assessments, requirements development, support concept trade studies, initial test and evaluation planning, initial PESHE planning, and industrial capability assessments for key technologies and components. The results of Phase A activities will provide critical input to the JCIDS process, allowing a well-founded CDD to be generated and validated in time to support KDPB.

The DoDI 5000.2 provides that this phase, Technology Development, is to reduce technology risk and to determine the appropriate set of technologies to be integrated into a full system. Technology Development is defined as a continuous technology discovery and development process reflecting close 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.

Phase A— Study Phase (Technology Development)

Goal: Determine the feasibility and desirability of a suggested new major system and its compatibility with air force strategic plans. Establish confidence in a selected alternative.

Common Practices And Products:

Define Operational And Threat Environments: Perform System Threat Assessment

Update Operational Capabilities & Requirements

Evolve Alternative Design Concepts: Feasibility & Risk Studies, Cost And Schedule Estimates, Advanced Technology Req'ts; Conclude AoA

Identify Alternative Ops & Logistics Concepts

Demonstrate Credible, Feasible Design(S) Exist

Acquire Systems Engineering Tools & Models

Initiate environmental impact studies: Perform Programmatic Environment Safety and Occupational Health Evaluation (PESHE)

Commence Systems/Requirements Definition

Prepare Acquisition Strategy

Perform Technology Readiness / Independent Technology Assessment

Prepare C4I Support Plan

Perform Affordability Assessment

Prepare Cost Analysis Requirements Description (CARD)

Prepare Test & Evaluation Master Plan (TEMP)

Technology Development Strategy (TDS)

Prepare Acquisition Program Baseline (APB)

Spectrum Certification Compliance

Reviews

Systems Readiness Review (SRR)

IPA Readiness Review

JROC CDD Review

Preliminary Program/Project Approval Review

In Phase A, a larger team, often associated with an ad hoc program or program office, readdresses the mission and operations concept to ensure that the project justification is sufficient to warrant a place in the budget. The team's effort focuses on analyzing mission requirements and establishing an operational architecture. Activities become formal, and the emphasis shifts toward establishing optimality rather than feasibility. The effort addresses more depth and considers many alternatives. Goals and objectives are solidified, and the project develops more definition in the system requirements, top-level system architecture, and operations concept. Conceptual designs exhibit more engineering detail than in the previous phase. Technical risks are identified in more detail and technology development needs become focused. The ICD and the technology Development Strategy (TDS) guides this effort. Multiple technology development demonstrations may be necessary before the user and developer agree that a proposed technology solution is affordable, militarily useful, and based on mature technology.

The NSS AP model provides for a SRR to occur in this Phase. Hence, systems definition formally begins during Phase A. Of course systems engineering planning is integral to all other program planning, e.g., program acquisition and technology strategic planning.

Chapter 5, What is Systems Engineering Management, provides guidance on planning and implementing systems engineering activities and processes to commence systems definition and requirements development activities. In addition, SMC has developed a Systems Engineering Guide Standard[26] to assist the program office to establish the foundation for systems engineering on a military space acquisition program and allocate or assign other systems engineering tasks between the Program Office and the system Contractor(s). The Guide Standard is designed to assist the Program Office to define the minimum essential contract compliance requirements for the tasks allocated or assigned to the system development or sustainment Contractors. In addition, the Guide Standard provides example RFP Section M evaluation criteria and source selection standards for evaluating Offerors' proposals for either alternative standards or corporate policies or further tailoring of a standard listed in the RFP by the Offerors.

Systems definition and requirements development activities actually may be initiated in Phase A and continue through Phase B as the process is recursive as the design matures. One of the Phase A goals is to define the system baseline and associated practices and products to meet this goal. Each Program Office must identify the practices/products that apply to each phase based on their planning and program needs.

Systems Definition establishes an initial project baseline, which includes a formal flow-down of the operational capabilities and performance requirements to a complete set of system and subsystem requirements and design specifications for space/flight and ground elements and corresponding preliminary designs. The technical requirements should be sufficiently detailed to establish firm schedule and cost estimates for the project.

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

The effort initially focuses on allocating functions to particular items of hardware, software, personnel, etc. System functional and performance requirements along with architectures and designs become firm as system trades and subsystem trades iterate back and forth in the effort to seek out more cost-effective designs.

26 SMC Guide Standard Systems Engineering Products, 18 July 03.

Phase B–Design Phase (System Development & Demonstration)

This Phase initiates the systems development efforts. However, The NSS AP model provides for a SRR to occur in the previous Phase A. Hence, Systems definition may have already started.

The NSS Acquisition Policy 03-01 states the purpose of this phase is to conduct risk reduction and design development activities. Phase B is designed to increase confidence in the selected NSS system alternative(s) by assessing the estimated risk levels and projected performance envelope at a detailed engineering level. Additionally, Phase B provides critical input to the JCIDS process, allowing a well-founded CPD to be generated and validated in time to support KDP-C.

The DoDI 5000.2 provides that in this phase, System Development & Demonstration, we develop a system or an increment of capability; reduce integration and manufacturing risk (technology risk reduction occurs during Technology Development); ensure operational supportability with particular attention to reducing the logistics footprint; implement human systems integration (HSI); design for producibility; ensure affordability and the protection of critical program information (CPI) by implementing appropriate techniques such as anti-tamper; and demonstrate system integration, interoperability, safety, and utility. Development and demonstration are aided by the use of simulation-based acquisition and test and evaluation integrated into an efficient continuum and guided by a system acquisition strategy and test and evaluation master plan (TEMP).

Phase B – Design Phase (System Development & Demonstration)

Goal: Complete the detailed design of the system

Common Practices and Products:

Conduct risk reduction, update Risk Management Plan, technology development, and continue component test and evaluation activities

Add remaining lower-level design specifications to the system architecture

Refine requirements documents

Refine verification plans

Prepare interface documents

(Repeat the process of successive refinement to get "build-to" specifications and drawings, verification plans, and interface documents at all levels)

Augment baselined documents to reflect growing maturity of system: system architecture, verification req'ts matrix, work breakdown structure, project plans

Monitor project progress against project plans

Develop system integration plan and system ops plan

Perform and archive trade studies

Complete manufacturing plan

Develop the end-to-end information system design

Refine Integrated Logistics Support Plan

Identify opportunities for p3 improvement

Update PESHE

Technical Information Baselined:

System requirements

Verification requirements

Requirements traceablity

System architectures (functional, physical, interface)

Work breakdown structure

Concept of operations

Complete set of specifications necessary to initiate detailed design

All remaining lower-level requirements and designs, including traceability to higher levels

"Build-to" specifications at all levels, drawings, TOs, ..

Reviews:

System Design Review

Preliminary Design Review

Safety review(s)

Subsystem (and lower level) Critical Design Reviews

System-level Critical Design Review

The Design Phase establishes a complete design (“build-to" baseline) that is ready to fabricate (or code), integrate, and verify. Trade studies continue. Engineering test units more closely resembling actual hardware are built and tested so as to establish confidence that the design will function in the expected environments. Engineering specialty analysis results are integrated into the design, and the manufacturing process and controls are defined and validated.

Configuration management continues to track and control design changes as detailed interfaces are defined. At each step in the successive refinement of the final design, corresponding integration and verification activities are planned in greater detail. During this phase, technical parameters, schedules, and budgets are closely tracked to ensure that undesirable trends (such as an unexpected growth in spacecraft mass or increase in its cost) are recognized early enough to take

corrective action. This phase culminates in a series of critical design reviews (CDRs) containing the system-level CDR and CDRs corresponding to the different levels of the system hierarchy. The CDR is held prior to the start of fabrication/production of end items for hardware and prior to the start of coding of deliverable software products. Typically, the sequence of CDRs reflects the integration process that will occur in the next phase— that is, from lower-level CDRs to the system-level CDR. Projects, however, should tailor the sequencing of the reviews to meet their individual needs. The final product of this phase is a "build-to" baseline in sufficient detail that actual production can precede.

Phase C–-Build, Test Launch (Production & Deployment, Operations & Support)

Phase C includes building, testing, and delivering the space-related system elements (e.g., satellite, booster and ground segments) and ensuring that necessary interfaces with the user elements function smoothly. Unless otherwise directed, the Program Manager also conducts studies to ensure the long-term reliability and maintainability of the system, to resolve emerging hardware or software problems, and to maintain mission performance over the planned life of the system. As the program moves into operations, the Program Manager is responsible for maintaining the system to accomplish those requirements allocated during the KDP-C process, as well as others that may be assigned by the USECAF. The Program Manager is expected to track these requirements closely as they evolve over time. The NSS acquisition model reflects a final Phase C to include Build, Test, and Launch. However, the DODI 5000.2 model provides for two more phases – Production & Deployment and Operations & Support. In order to address both acquisition models (DoDI and NSS AP) sufficiently, this handbook provides for four sub-phases for Phase C:

- Phase C1 -- Build, Test (Production)
- Phase C2 – Launch (Deployment)
- Phase C3 – (Operations & Support)
- Phase C4 – Disposal

Phase C1–Build, Test (Production)

The purpose of this phase is to build and verify the system designed in the previous phase and prepare for deployment and operations. Activities include fabrication of hardware and coding of software, integration, and verification of the system. Other activities include the initial training of operating personnel and implementation of the Integrated Logistics Support Plan. For flight projects, the focus of activities then shifts to pre-launch integration and launch. For large flight projects, there may be an extended period of orbit insertion, assembly, and initial shake-down operations. The major product is a system that has been shown to be capable of accomplishing the purpose for which it was created.

Build & Test includes the fabrication of engineering test models and "brass boards," low rate initial production, full-rate production of systems and end items, or the construction of large or unique systems or sub-systems.

At Production Readiness and LRIP system-level demonstrations are accomplished and the product baseline is defined (although it will be refined as a result of the activities undertaken during this phase). The effort is now directed toward development of the manufacturing capability that will produce the product or system under development. When a manufacturing capability is established, a LRIP effort begins. The development of a LRIP manufacturing capability has multiple purposes. The items produced are used to proof and refine the production line itself, items produced on this line are used for Initial Operational Test and Evaluation (IOT&E) and Live Fire Test and Evaluation (LFT&E), is also the means by which manufacturing rates are ramped upward to the rates intended when manufacturing is fully underway.

Following submission of test reports, and a full-rate production decision by the MDA, the system enters the Rate Production and Deployment stage. After the decision to go to full-rate production, the systems engineering process is used to refine the design to incorporate findings of the independent operational testing, direction from the MDA, and feedback from deployment activities. Once configuration changes have been made and incorporated into production, and the configuration and production is considered stable, Follow-on Operational Test and Evaluation (FOT&E), if required, is typically performed on the stable production system. Test results are used to further refine the production configuration.

Phase C1— Build, Test (Production)

Goal: Build and verify the system designed in the previous phase.

Common Practices and Products:

Fabricate (or code) the parts (i.e., the lowest-level items in the system architecture)

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

(Repeat the process of successive integration to get a verified system)

Develop verification procedures at all levels

Perform system qualification verification(s)

Perform system acceptance verification(s)

Monitor project progress against project plans

Archive documentation for verifications performed

Audit "as-built" configurations

Document Lessons Learned

Prepare operator's manuals

Prepare maintenance manuals

Train initial system operators and maintainers

Finalize and implement Integrated Logistics Support Plan

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

Perform operational verification(s)

Information Baselined:

"As-built" and "as-deployed" configuration data

Integrated Logistics Support Plan

Command sequences for end-to-end command and telemetry validation and ground data processing

Operator's manuals

Maintenance manuals

Reviews:

Test Readiness Reviews (at all levels)

Acceptance Reviews

System functional and physical configuration audits

Flight Readiness Review

Operational Readiness Review

Phase C2–Launch (Deployment)

Deployment (Fielding) includes the activities necessary to initially deliver, transport, receive, process, assemble, install, checkout, train, operate, house, store, or field the system to achieve full operational capability. As the system is produced, individual items are delivered to the field units that will actually employ and use them in their military missions. Careful coordination and planning is essential to make the deployment as smooth as possible. Integrated planning is absolutely critical to ensure that the training, equipment, and facilities that will be required to support the system, once deployed, are in place as the system is delivered. The systems engineering function during this activity is focused on the integration of the functional specialties to make certain that no critical omission has been made that will render the system less effective than it might otherwise be. Achieving the user's required initial operational capability (IOC) schedule demands careful attention to the details of the transition at this point. Furthermore, as the system is delivered and operational capability achieved, the system transitions to the Sustainment and Disposal phase of the system life cycle–the longest and most expensive of all phases.

Phase C3– Operations & Support

DODI 5000.2 states that the objective of this activity is to execute a support program that meets operational support performance requirements and sustains the system in the most cost-effective manner over its total life cycle. In this phase we demonstrate whether we truly meet the initially identified need or grasp the initially identified opportunity. The products of this phase include the results of the mission. Operations also encompass sustainment elements such as supply, maintenance, transportation, sustaining engineering, data management, configuration management, manpower, personnel, training, habitability, survivability, environment, safety, occupational health, protection of critical program information, anti-tamper provisions, and information technology, supportability and interoperability.

This phase includes evolution of the system only insofar as that evolution does not involve major changes to the system architecture; changes of that scope constitute new "needs," and the project life cycle starts over.

Phase C3—Operations & Support

Goal: Meet the initially identified need or to grasp the opportunity.

Common Practices and Products:
Train replacement operators and maintainers
Conduct the mission(s)
Maintain and upgrade the system
Document Lessons Learned

Information Baselined:
Mission outcomes, such as:
· Engineering data on system, subsystem and materials performance
· Mission data returned
· Accomplishment records ("firsts")
Operations and maintenance logs
Problem/failure reports

Reviews:
Regular system operations readiness reviews
System upgrade reviews

Phase C4–Disposal

For a flight system with a short mission duration, such as a launch vehicle payload, disposal may require little more than de-integration of the hardware and its return to its owner. Alternately, planned disposal may include orbital maneuvers to a predetermined location. On large flight projects of long duration, disposal may proceed according to long-established plans, or may begin as a result of unplanned events, such as accidents. Alternatively, technological advances may make it uneconomic to continue operating the system either in its current configuration or an improved one. In addition to uncertainty as to when this part of the phase begins, the activities associated with safely decommissioning and disposing of a system may be long and complex. Consequently, the costs and risks associated with different designs should be considered during the project's earlier phases.

Phase C4 -- Disposal

Goal: Dispose of the system in a responsible manner.

Common Practices and Products:

Dispose of the system and supporting processes

Dispose following all legal and regulatory requirements and policy relating to safety, security, and the environment.

Document Lessons Learned

Reviews:

Decommissioning Review

The DoD 5000 acquisition model stresses flexibility in the process to bring effective systems on line as quickly and affordably as possible. It fosters evolutionary development, whereby new system requirements are met by building on existing Government and commercial systems, equipments and technologies. Fielded systems may achieve full capability in a single step, or improvements may be added incrementally in subsequent blocks of production.

System Engineering has a continuing but changing role in each phase. In the Concept refinement and Technology Development phases, emphasis is on the Requirements Analysis activities in the definition/refinement of general requirements and overall feasibility. System Engineering assists the User in articulating capabilities to prepare the ICD and IDDs . The systems engineers also identify needed research for technologies that will reduce the system development risk. Competing concepts are developed as possible system solutions. The Systems Engineering Requirements Loop is exercised to convert User requirements and capabilities to system requirements and possible functional implementations. Analyses and trade studies are conducted to help select preferred alternatives. The costs of efforts at this stage are relatively small as compared to follow-on investments. Often several contracts are let to allow the procuring agency to choose two or three of the best among those proposed for further development. Prototypes are built to demonstrate the feasibility of components or complete equipment sets. Designs are implemented with existing technology or discrete components with the intent of substituting such items as advanced devices or large-scale integration (LSI) in later phases. Alternate Systems Review(s) (ASRs) evaluate the efficacy of each concept. If applicable, a System Threat Assessment Report (STAR) provides an evaluation of any threats, which could affect the performance of the system mission. Using the results of the ASR(s) and the STAR (if applicable), the User’s requirements are refined and detailed in an Operational Requirements Document (ORD).

Usually only a single concept survives to System Development and Demonstration (SD&D) phase. However, the Procuring Agency may occasionally retain more than one concept if funding is available. In this way the Agency can pursue a highly innovative concept that promises greater performance but entails greater risk while maintaining as insurance, a more conservative alternate approach that uses proven technology. Similarly, the Agency may wish to maintain cost competition into the next phase. Early in this phase (or possibly late in then previous phase) a System Requirements Review (SRR) is held to assure that all parties (User, Procuring Agency and Contractors) are aware and agree on the requirements for the system under development. During SD&D, the System Engineering activities begin to transition from the Requirements Loop to the Design Loop with analyses and trade studies performed to assist in selecting preferred solutions. The Functional Analysis and Allocation tasks become more prominent and the functional baseline and system specification for each concept are developed. When the functional baseline is sufficiently mature, a System Functional Review (SFR) is held. At the SFR the system specification and functional baseline for the concept is reviewed to determine whether the proposed system meets requirements, is achievable, and is ready to proceed to preliminary design. The CDD may be updated based on the SFR results and any updated STAR.

Engineering and Manufacturing Development (EMD) follows. The name is significant because not only does it indicate that the system is under development, but also anything needed to manufacture and test the system. Rarely is there more than one concept at this point, but occasionally another contractor is retained as a second source. During EMD the emphasis is on the synthesis activities with trade studies and analyses to narrow the selection of ways in which the hardware and software might be implemented. Configuration Item (CI) requirement allocation is finalized and design solutions are translated into system hardware and software that meet the User’s need. In addition, during EMD all the things necessary to manufacture and support the system are developed -- manufacturing processes, technology, equipment and facilities; special test equipment and facilities; support equipment; training for production workers, system maintainers and system operators; etc. In EMD, System Engineering is also engaged in developing test requirements which will indicate that the system design meets User needs (qualification testing) and that individual systems meet established performance norms (acceptance testing).

Three major system reviews occur during EMD: Preliminary Design Review (PDR), Critical Design Review (CDR) and a Test Readiness Review (TRR). The PDR confirms that the system detailed design approach satisfies the functional baseline, that risks are under control, and that the system is ready for detailed design. If applicable, a Software Specification Review (SSR) is usually held with the system PDR. CDR demonstrates that the total system design is complete and meets requirements, that hardware elements are ready for initial builds, and that software elements are ready for coding. The complete allocated baseline and development specifications for all CIs are reviewed and approved prior to committing to hardware. It confirms readiness for full-scale production. Also during EMD, similar reviews (PDRs and CDRs) are conducted on individual CIs and Computer Software Configuration Items (CSCIs). A Test Readiness Review (TRR) is held before system and CI testing is initiated. The test results and any new STAR information are used to update the ORD.

In the Production and Deployment phase the system is produced and fielded. A Functional Configuration Audit (FCA) and a System Verification Review (SVR) is conducted on the product specification, all associated process and material specifications, and on a representative system from the first production run. When the system has been approved for production, a system Physical Configuration Audit is conducted to establish the product baseline for subsequent production systems. PCAs on all constituent CIs are completed prior to the system PCA and reviewed as part of the audit. Preceding or concurrent with the system deliveries support equipment and facilities are provided along with operation/maintenance training.

Changes occur throughout the operational life of the system. Missions change or are augmented. Threats change or new threats appear. Deficiencies are uncovered. New devices or technology provide improved performance, reliability or availability. Parts of the system become obsolete or are no longer supportable. All these factors lead to product improvements in the fielded system. During the Operations and Support (O&S) phase, System Engineering’s role is to evaluate competing implementations and their relative effect on other elements of the system, chose the best, foster their development, orchestrate the changes, and maintain the evolving configuration baseline. Each change or group of changes is handled as a new development. For small changes, the process may be fairly informal. However, for major or critical changes, the complete formal review structure with SRR, PDR, CDR and PCA may be invoked. Throughout the remainder of the program, including the safe and secure disposal, System Engineering is responsible for the integrity of the system.

Milestones occur at major decision points in the acquisition model (Figures 20 and 21) with a Milestone Decision Authority (MDA), whose DoD level is dependent upon the size and criticality of the program, making a determination as to whether continuation into the next phase is warranted:

Milestone A – At start of Concept and Technology Development phase, authorizes initiation of concept studies. Requirements for these studies and related activities are documented in a Mission Need Statement. The MDA defines the goals of the activities in exit criteria that indicate what must be accomplished to support continuation into the System Development and Demonstration phase. A favorable Milestone A decision is not an authorization of a new acquisition program, merely a go-ahead to explore system concepts and underlying technology development.

Milestone B – At start of System Development and Demonstration phase, authorizes initiation of an acquisition program. Requirements for these activities are documented in an Operational Requirements Document. Since this is the DoD's commitment to a systems acquisition, in making the Milestone B decision the MDA must consider the validated ORD, the System Threat Assessment, an independent assessment of technology status and issues, early operational assessments or test and evaluation (T&E) results, analyses of alternatives, independent cost estimates, system affordability and funding, proposed acquisition strategy, cooperative opportunities, and infrastructure and operational support. At Milestone B the MDA confirms the acquisition strategy, the development acquisition baseline, low-rate initial production quantities (if applicable) and the System Development and Demonstration exit criteria.

Milestone C – At the start of the Production and Deployment phase, authorizes entry into low-rate production (for Major Defense Acquisition Programs – MDAPs, and major programs) into production or procurement (for non-major systems that do not require low-rate production) or into limited deployment for Major Automated Information Systems – MAISs, or software-intensive systems with no production components. In making the Milestone C decision the MDA must consider the independent cost estimate, manpower estimate, System Threat Assessment, Critical Program Information protection and anti-tamper recommendations, and the program for National Environmental Policy Act compliance. At Milestone C the MDA confirms the acquisition strategy, the development acquisition baseline update, exit criteria for low-rate initial production (LRIP), if applicable, or limited deployment.

Not all acquisitions follow the entire baseline model. Depending on the status of implementing technology and criticality of user's need, a program may enter the model at any of the three milestones, and advance through sub-phases as required. This flexibility takes full advantage of prior government and commercial investment in Commercial-Off-the-Shelf (COTS) and Non-Developmentally Items, and to facilitate rapid and effective transition from Science and Technology to Products, and from Acquisition to Deployment and Fielding.

Systems Engineering–Software Development

Software development has been touched upon periodically up to this point. Weapon system software development is considered high risk.

Systems Software Development

Software development is a labor intensive, costly, and often high-risk effort. We choose software in our designs to provide greater system performance, versatility, and flexibility of those functions that can be implemented through programmable processing. In recent years, our greatest challenges to finalize system design or major upgrades have been centered on software problems. For these reasons, emphasis on software development and test is as important as hardware. Though software is addressed throughout this SMC Systems Engineering Textbook, we provide more focused software discussion in this section. More information on this topic can be found in the SMC Software Acquisition Project Officer's Guide. This guide can be obtained through SMC/AXE.

Evolution of Software Development Standards

The DoD approach to managing software development efforts has changed dramatically over the last 10 years. As embodied in DoD 5000.1 and DoD 5000.2 the emphasis in acquisition management had shifted from government development of detailed specifications of system parameters to more performance-based measures of system requirements, allowing the designer more freedom to define the most appropriate means of implementing these requirements.

SMC is currently reestablishing contract compliancy requirements for software development. Though background discussion on software standards is provided below, the reader is advised to obtain the latest guidance from SMC/AXE to determine appropriate RFP and contract software requirements for their project. Software related military standards have been cancelled. However, some of the older active contracts may still impose requirements from these standards. DOD-STD-2167A DEFENSE SYSTEM SOFTWARE DEVELOPMENT was the first software standard to establishes uniform requirements for software development that are applicable throughout the system life cycle. The software development process prescribed by this standard included major activities that are applied iteratively or recursively:

- System Requirements Analysis/Design
- Software Requirements Analysis
- Preliminary Design
- Detailed Design
- Coding and CSU Testing
- CSC Integration and Testing
- CSCI Testing.
- System Integration and Testing

MIL-STD-498, SOFTWARE DEVELOPMENT AND DOCUMENTATION replaced DOD-STD-2167A on 5 December 1994. Several years later in 27 May, 98, MIL-STD-498 was cancelled. These software standards are still available at various DoD web sites.

The international standard for the development, acquisition, maintenance, supply, and operation of software, ISO/IEC 122071, was approved in 1995. A joint working group of the Institute of Electrical and Electronics Engineers (IEEE)/ and the Electronics Industries Association (EIA) adapted the ISO/IEC 12207 standard to be more in line with United States software lifecycle practices. The IEEE/EIA standard, IEEE/EIA 12207 "Information technology-Software life cycle processes", is packaged in three parts. The three parts are: IEEE/EIA 12207.0, "Standard for Information Technology-Software life cycle processes"; IEEE/EIA 12207.1, "Guide for ISO/IEC 12207, Standard for Information Technology-Software life cycle processes-Life cycle data"; and IEEE/EIA 12207.2, "Guide for ISO/IEC 12207, Standard for Information Technology-Software life cycle processes-Implementation considerations."

There are three fundamental differences between the DoD standards and these industry standards.

- The industry standards are designed to be implemented on a voluntary basis while the DoD standards were imposed contractually.
- The new standards are intended to be applied to the full software lifecycle: development, acquisition, maintenance, supply, and operation.
- IEEE/EIA 12207 is written at a much higher level than the DOD predecessor standards and avoids dictating particular software development approaches and life cycle models. IEEE/EIA 12207 does not provide detailed specification to perform the software development tasks.

Highlights of the IEEE/EIA 12207 industry standards are provided here:

- Covers the system lifecycle development, acquisition, maintenance, supply, and operation of software
- Written to be compatible with the ISO 9000 approach to quality systems, quality management, and quality assurance
- Includes references to other applicable industry standards
- Complies with the international version of the standard, ISO/IEC 12207

Currently SMC/AX is reviewing IEEE J-STD-16-1995, MIL-STD-498, IEEE/EIA 12207 and other to determine applicability for contractual compliance on our space systems contracts. For the latest guidance, contact SMC/AXE.

Software Acquisition Strategy Considerations

Mandatory and discretionary acquisition information pertaining to software development are located in Department of Defense Instruction 5000.2 and the Defense Acquisition Deskbook. Since the Milestone Decision Authority (MDA) approval at these Milestones are dependent on software related criteria being met, some of the mandatory directives are provided in this handbook. Of course it is prudent to be familiar with all software related mandates and Milestone criteria. Hence, a thorough review of the latest 5000 series instructions and directives is necessary.

Two acquisition strategy approaches are frequently used to structure a program to achieve full capability: evolutionary and single step. An evolutionary approach is preferred. Evolutionary acquisition is an approach that fields an operationally useful and supportable capability in as short a time as possible. This approach is particularly useful if software is a key component of the system and the software is required for the system to achieve its intended mission. Evolutionary acquisition delivers an initial capability with the explicit intent of delivering improved or updated capability in the future.

The approach to be followed depends on the availability of time-phased capabilities/requirements in the CDD, the maturity of technologies, the relative costs and benefits of executing the program in blocks versus a single step, including consideration of how best to support each block when fielded. The most recent practice requires that the rationale for choosing a single step to full capability, when given an CDD with time-phased requirements, be addressed in the acquisition strategy. Similarly, the rationale for choosing an evolutionary approach, when given an CDD with no time-phased requirements, should be addressed in the acquisition strategy.

For both the evolutionary and single-step approaches, software development and integration may follow an iterative spiral development process in which continually expanding software versions are based on learning from earlier development. In addition, programs with software components must be capable of responding to emerging requirements that will require software modification or periodic enhancements after a system is deployed.

Software Development Lifecycle

Mil-Std-2167A provided the DOD approach to software development and was based on the waterfall model, Figure 22, of software development. Two major shortcomings were recognized with the waterfall model. First, the characteristic sequential evolution includes phases of software development activities that allow only iterations between adjacent phases. Second, the iterations between the waterfall phases are often too long which tends to lengthen the time period from statement of User needs to production of a system. Barry Boehm introduced the spiral development model in 1986 to shorten the software development lifecycle Figure 23 represents a somewhat modified version of the first spiral lifecycle model.

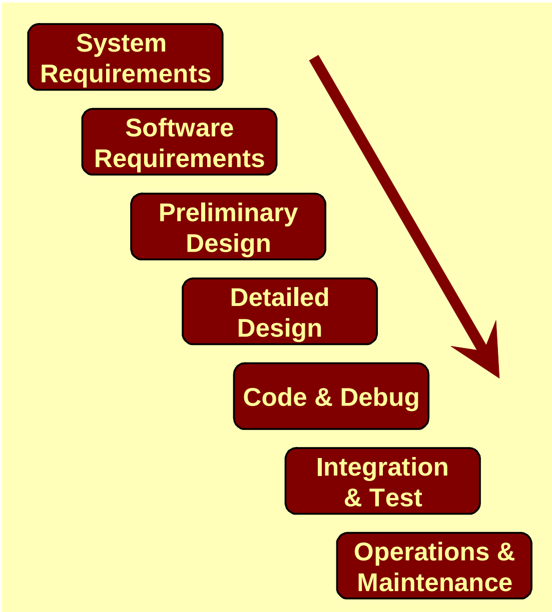


Figure 22. Waterfall method

The Program Manager might plan a spiral development process for both evolutionary and single-step-to-full-capability acquisition strategies. DODI 5000.2 characterizes spiral development as a cyclical, iterative build-test-fix-test-deploy process to yield continuous improvements in software. Boehm [1], on the other hand, describes his model as a risk driven approach rather than a document or code driven process. The spiral applies equally to new development and upgrades/enhancements. The spiral has four phases. Starting from the first quadrant clockwise: determine objectives, alternatives, and constraints; identify and resolve risks; develop and verify design and products; and plan next phase.

Determine Objectives, Design Alternatives and Constraints

The stakeholders' initial emphasis is to determine the performance objectives of the software and possibly the ability of the software to be upgraded. They also identify constraints (e.g., cost, schedule, security, environments, etc) that apply to each of the alternatives under review. The objectives and constraints provide the basis for the software requirements.

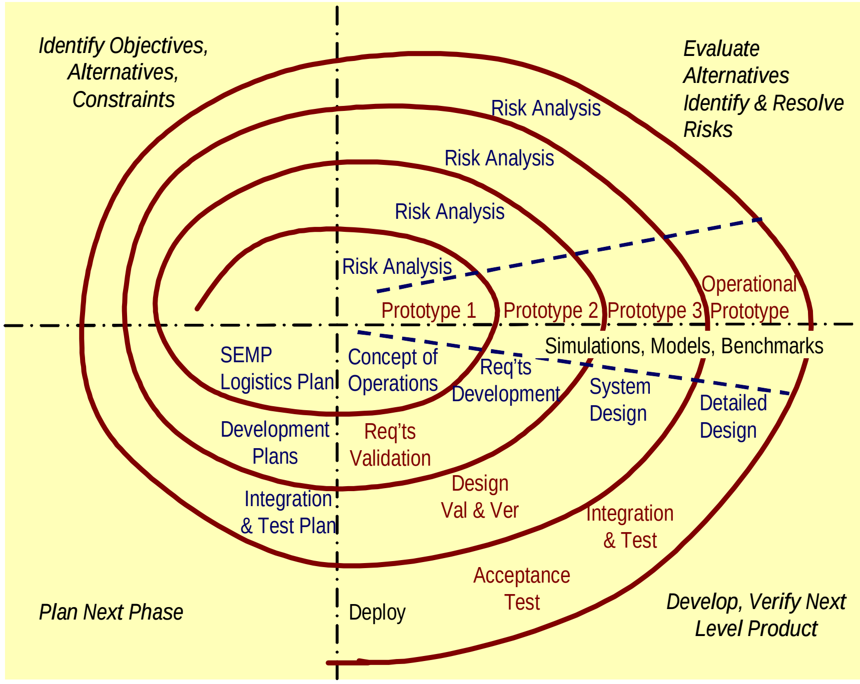


Figure 23. Spiral development lifecycle model

Identify and Resolve Risks

The emphasis for risk considerations distinguish Boehm's model from the rest. He cites numerous risk mitigation/resolution techniques such as prototyping, simulation, benchmarking, reference checking, user questionnaires, and analytical modeling. This development model includes software mock-up or prototyping activities [1]. The prototyping effort is initially for concept proofing and supports the Users/Operators to define their requirements. Subsequent evolutions of the spiral support prototyping of detailed design, and finally operational design. Rapid prototyping (not an element of the Boehm model) is intended to produce partially operational mock-ups/prototypes early in the design (initiated during preliminary design phase).

Develop and Verify Design and Products

The Develop and Verify phase has recognizable elements that are included in the waterfall model: requirements, design, code, integrate and test. For the spiral model, the Develop and Verify phase is entered four times. Planning, alternative assessments, and risk analysis are performed each time preceded by requirements development and validation, preliminary/product design, and detailed design.

Software Requirements Analysis and Definition

Software requirements analysis involves defining and baselining requirements for each Computer Software Configuration Item (CSCI) based on the system specification and operational User needs. The requirements analysis should concentrate on the capabilities and performance of the entire system. This includes the software and the environment in which the system is to operate.

Approaches for performing software requirements development are the same for hardware and have been touched upon in a previous section of this handbook. IEEE/EIA 12207 also describes this process for allocating requirements in a top-down fashion. The standard requires first that system-level requirements be allocated to software items and that the software requirements then be documented in terms of functionality, performance, and interfaces. The standard also specifies that software item requirements be traceable to system-level, and be consistent and verifiable. Of course, the requirements analysis includes a cost/benefit analysis to determine the costs associated with developing, maintaining and operating the software of each design alternative under study.

Preliminary Design

During Preliminary Design the overall software structure is developed. The software developer decomposes each software item into software components/modules. If a functional design approach is taken, the developer then identifies the functional interrelationship of the modules as well as the functions within each module.

However, the software developer may elect to use the object-oriented approach to design the software. Objects of this model are physical resources or elements that perform activities and transmit messages to other objects. Object oriented models of a system provide three views: the object (physical or data repository) view, the functional view (processing or data flow), and the behavior (state transition or nested state) diagrams. Data and functionality are localized within the objects rather than being scattered as occurs using functional decomposition methods. This method produces more robust modularity with fewer interfaces. One drawback to object oriented design is that functional compatibility and traceability to system requirements is difficult to assess.

Detailed Design–Coding

During Detailed Design, requirements are allocated from item level, to component, and eventually to unit level when using the functional design approach. IEEE/EIA 12207 requires that lower level of requirements allocations are documented or described. (See the IEEE/EIA standard for documentation requirements. The level of documentation detail required varies depending on project needs.) Obviously, all detailed design activities prior to coding are not complete when using the spiral development approach. Selected partitions/modules of the software will have been completed as a result of prototyping in the previous phase.

CSCI Integration & Systems Testing

CSCI testing involves testing an element of a system to ensure that it meets the requirements defined during system requirements review. System integration and test ensures that the software works within the system environment as specified. Over the last few years, much emphasis is being placed on interoperability of weapon systems. Interoperability is 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 5000.1 May 01, 2003, 2001, E1.13 and E1.16., address the goals to achieve interoperability within and among United States forces and U.S. coalition partners so that the Department of Defense has the ability to conduct joint and combined operations successfully. Interoperability is certainly applicable to SMC systems in many respects. Hence, the Program Offices' systems engineers need to ensure the full set of satellite, control, and user interoperability requirements are identified and met. They must consider the use of standardized data to facilitate interoperability and information sharing with other systems. To the highest extent possible, systems and software must be designed, consistent with U.S. export control laws and regulations, to permit use in a multi-national environment with provision made for current and future information disclosure guidance and constraints. Improved interoperability of weapon systems is expected to be achieved through the C4ISR architecture framework as well. The framework is intended to ensure that architecture descriptions can be compared and relate across organizational and system boundaries. Refer to Appendix C11 C4ISR Architecture Framework for more detailed discussion on the C4ISR subject.

Chapter 4
What is Systems Engineering Management?

The System Engineering 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. Systems engineering management interacts with all other activities of the systems engineering 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 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 systems engineering 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.

One engineering management philosophy that the Air Force has used to address the system complexities is summed in Integrated Product and Process Development (IPPD). In this approach, the product-oriented Work Breakdown Structure (WBS) introduced earlier under the work view becomes the outline for planning, organizing, and directing. The Integrated Master Plan (IMP), Integrated Master Schedule (IMS), and Earned Value Management System (EVMS) also introduced under the work view form much of the planning. The organization mirrors the upper levels of the WBS. The IMP, IMS, and EVMS supplemented by Technical Performance Measures (TPMs) and other specific risk monitoring devices form the basis for monitoring. Project control function is performed using immediate action plans and longer-term updates to the IMP, IMS, EVMS, and TPMs.

As it is applied to systems engineering, planning has two aspects: definition of the process and organization responsibilities for implementing the process ("how") and identification and flow of tasks to apply the process to the program at hand ("what"). "How" is typically defined in either process narratives in the IMP or in a Systems Engineering Management Plan (SEMP) (or both with an overview in the IMP and details in the SEMP). In an IPPD program, "what" is defined in increasing detail in the IMP, IMS, and EVMS. In a program not applying IPPD, less integrated program management plans and schedules, such as water-fall (Gantt) or critical-path charts, may be used in lieu of the IMP and IMS.

The success of the systems engineering 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.

What is Management?

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 systems engineering 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 systems engineering 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,
- estimate and manage cost, technology, and schedule, and to monitor program activities and trends,
- procure necessary materials, data, and services to ensure smooth running of the program,
- assess, manage, devise policy, and implement 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 systems engineering process, described in this handbook, then 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 systems engineering processes. She or he in turn 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. Via the contract, he or she 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 systems engineering process and the systems engineering organization. Typically, most to all of the organization has responsibilities associated with implementation of the systems engineering process while only one to a few organizational entities have systems engineering in their title. 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.

Relationship of Systems Engineering Management to Overall Program Cost, Schedule and Risk

It is the responsibility of systems engineering 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. Verification that the design provides the needed capabilities (or meets the requirements), estimating all elements of program cost, monitoring adherence to the schedules, and assessing and monitoring the risk are, therefore, all essential systems engineering management tasks, no matter how responsibility for them is assigned in the Government and Contractor organizations. Stated a different way, the assessment of all those factors is essential to monitoring the implementation of the systems engineering 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 (the Planning, Programming, and Budgeting System, PPBS) 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. Obviously, for capabilities, cost, schedule, and risk to be balanced, then the capabilities set by the Requirements Generation System, direction given by the Defense Acquisition System (including acquisition strategy, schedule, and the like), the budget set by the PPBS, and other program inputs must be in balance. Typically, the relationship between these factors is as shown in Figure 24 below.

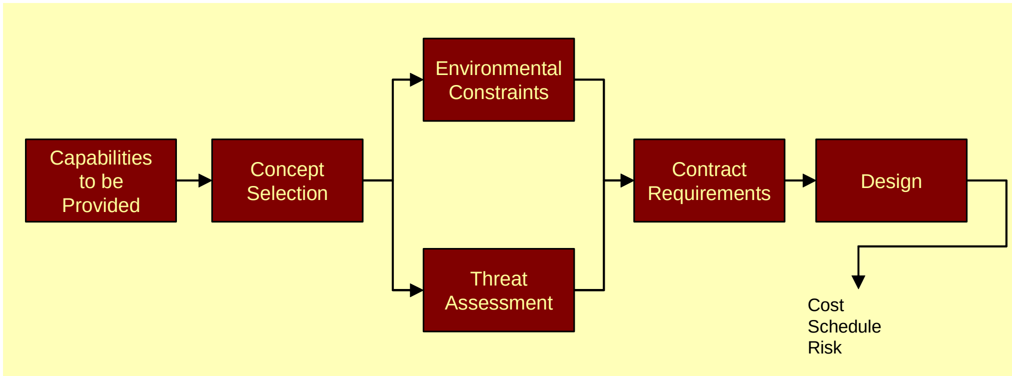


Figure 24. Typical relationship of capabilities and other program inputs to cost, schedule, and risk

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, and the systems engineering process prepares the contract requirements accordingly. The design that complies with the contract requirements then follows from the systems engineering 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 apace. It should be clear, therefore, that the systems engineering process has to interact closely with the Requirements Generation System, the Defense Acquisition System, the intelligence and environmental communities, and the PPBS to balance capabilities, cost, schedule, and risk. In a program where such interactions are not effective, cost growth and schedule slippage is almost certain as is an adverse impact on the careers of those involved, and program cancellation is a real possibility.

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

Planning and Organizing

The steps in planning and organizing for systems engineering 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 systems engineering 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.

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 which 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 IEEE P1220). 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[27] and.[28] 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.

Integrated Master Plan (IMP) Narrative/Systems Engineering Management Plan (SEMP)

As discussed earlier, the systems engineering process and responsibilities for its implementation are usually described in an IMP Narrative and/or SEMP. An outline for a SEMP showing the kinds of data that can be considered for inclusion is in Appendix C1.

All required technical specialties should be addressed as an integrated part of the systems engineering process. At times, some of these are covered in separate plans but, if so, the IMP Narrative 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 (CM), Data Management (DM), and Operational Safety, Suitability, & Effectiveness (OSS&E) are addressed in a separate subsections below. Still other specialties are covered in Chapter 6 and verification and validation are covered in Chapter 7 below.

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

28. IEEE Std 1220-1998, IEEE Standard for Application and Management of the Systems Engineering Process, The Institute of Electrical and Electronics EngineersEngineers, Inc.345 East 47th Street, New York, NY 10017-2394.

The Work Breakdown Structure

As noted in earlier discussions, the product-oriented Work Breakdown Structure (WBS) evolves with and reflects the physical design that is a product of the systems engineering effort so it is discussed further here. 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 systems engineering 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.

Since the WBS is a derivative of the physical and systems architectures, it is a direct output of the systems engineering 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. A sample WBS is shown under the discussion of the Work View in Chapter 1.

The Interim Guidebook for the DoD 5000 series directives lists a program WBS as a best practice to provide the framework for both 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 products. Program offices develop a Program WBS (or PWBS) tailoring the guidance provided in MIL-HDBK-881. The WBS is also an essential step in the preparation of the Cost Analysis Requirements Description (CARD) which is used as a basis for independent cost and other assessments. The Series 5000 Interim Guidebook suggests that Program Offices develop an overall PWBS and to initiate development of a contract WBS (CWBS) for each contract in accordance with common DoD practice established in MIL-HDBK-881.[29] The program WBS represents the total system and, therefore, reflects the system architecture. The contract WBSs relate to deliverables and tasks on a specific contract. The Program Office usually develops the first three levels of the program WBS to provide contractors with guidance for lower-level WBS development. As with many standards and most handbooks, use of MIL-HDBK-881 cannot be specified as a contract requirement. Though WBS is a product of the systems engineering 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.

A top level example program WBS for a space system is in Appendix C2.

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

29. MIL-HDBK-881, DoD Handbook -- Work Breakdown Structure, 2 January 1998

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.

Monitoring and Control

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.

Earned Value Management (EVMS)[30]

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.

Baseline

The baseline plan in Table 4, shows that 6 work units (A-F) would be completed at a cost of $100 for the period covered by this report.

Table 4. Baseline plan work units

	A	B	C	D	E	F	Total
Planned value ($)	10	15	10	25	20	20	100

Schedule Variance

As work is performed, it is "earned" on the same basis as it was planned, in dollars or other quantifiable units such as labor hours. Planned value compared with earned value measures the dollar volume of work planned vs. the equivalent dollar volume of work accomplished. Any difference is called a schedule variance. In contrast to what was planned, Table 5 shows that work unit D was not completed and work unit F was never started, or $35 of the planned work was not accomplished. As a result, the schedule variance shows that 35 percent of the work planned for this period was not done.

Table 5. Schedule variance work units

	A	B	C	D	E	F	Total
Planned value ($)	10	15	10	25	20	20	100
Earned value ($)	10	15	10	10	20	-	65
Schedule variance	0	0	0	-15	0	-20	-35 = -35%

30. Source: Defense Air University, Defense Acquisition Deskbook.

Cost Variance

Earned value compared with the actual cost incurred (from contractor accounting systems) for the work performed, provides an objective measure of planned and actual cost. Any difference is called a cost variance. A negative variance means more money was spent for the work accomplished than was planned. Table 6 shows the calculation of cost variance. The work performed was planned to cost $65 and actually cost $91. The cost variance is 40 percent.

Table 6. Cost variance work units

	A	B	C	D	E	F	Total
Earned value ($)	10	15	10	10	20	-	65
Actual cost ($)	9	22	8	30	22	-	91
Cost variance	1	-7	2	-20	-2	0	-26 = -40%

Spend Comparison

The typical spend comparison approach, whereby contractors report actual expenditures against planned expenditures, is not related to the work that was accomplished. Table 7 shows a simple comparison of planned and actual spending, which is unrelated to work performed and therefore not a useful comparison. The fact that the total amount spent was $9 less than planned for this period is not useful without the comparisons with work accomplished.

Table 7. Spend comparison approach work units

	A	B	C	D	E	F	Total
Planned spend ($)	10	15	10	25	20	20	100
Actual spend ($)	9	22	8	30	22	-	91
Variance	1	-7	2	-5	-2	20	9 = 9%

Use of Earned Value Data

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. For more information, refer to POH 7.7.1.4, Primer, NSS Acq. Process, EUMS at www.smc.sparta.com/golive/site16.

Reviews and Audits

Requirements reviews, design reviews, and configuration audits provide an opportunity to assess program status in considerable detail. In particular, requirements and design reviews can be essential to monitoring at points in the program prior to the availability of test and other verification data that provide a direct indication of contract compliance. MIL-STD-1521 provides a generic summary of what to look for at each review. The System Engineering Critical Process Assessment Tool (CPAT) provides more detail on what to look for. See http://ax.losangeles.af.mil/axm/axmp/CPAT/cpat.html or POH Chapter 9 at www.smc.sparta.com/golive/site16.

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 systems engineering 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 25. 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. For more information, refer to POH 7.72 at www.smc.sparta.com/golive/site16.

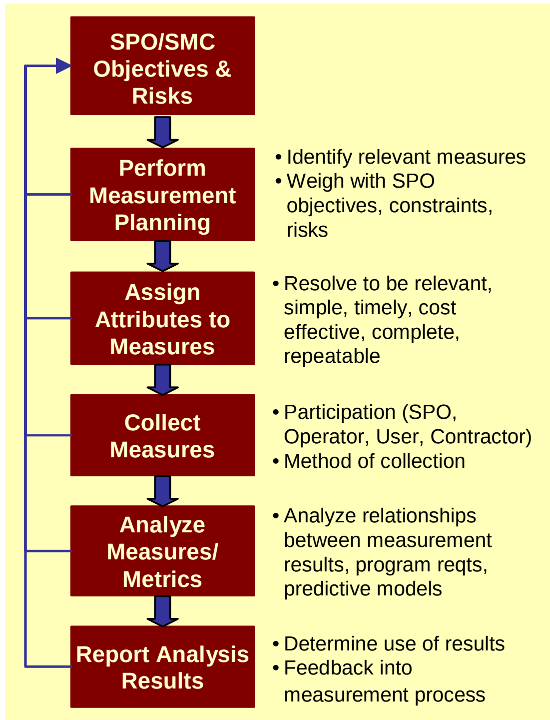


Figure 25. Metrics development and reporting process

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.

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 the Figure 26, shown below.

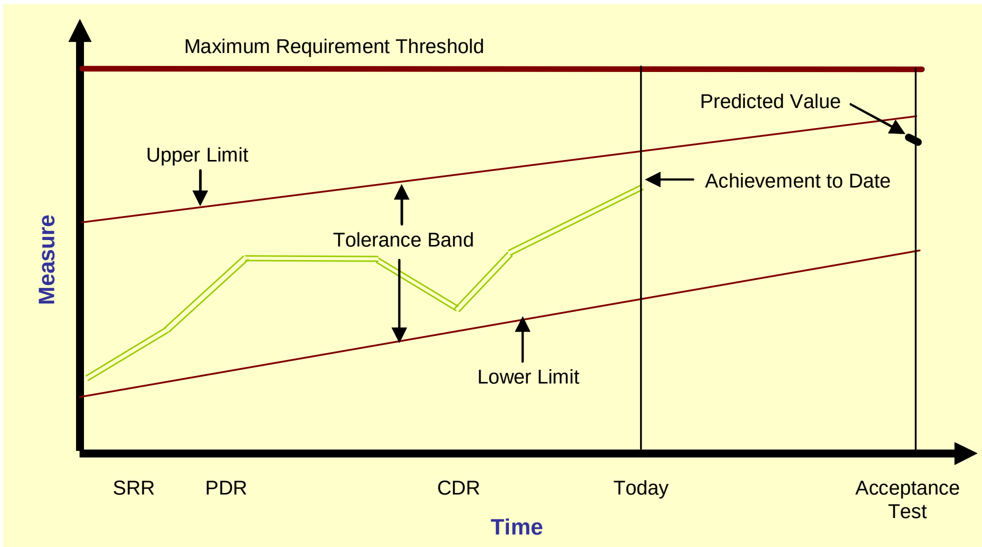


Figure 26. Performance measures tracked over time

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 systems engineering activities. That is, a TPM tracking program forges a relationship among systems analysis, functional and performance requirements definition, and verification and validation activities[31]:

- 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
- 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

31 NASA Systems Engineering Handbook SP 6105, June 1995

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:

- Performance parameters that are significantly qualifies 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 systems engineering managers but also adds visibility of program status reporting. Appendix C7 provides example TPMs using this hierarchy methodology. For more information, refer to POH 7.73.2 at www.smc.sparta.com/golive/site16.

Systems Analysis–the 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.

The INCOSE Systems Engineering Handbook[32] explains the purposed 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 a high-level system architecture to the selection of a specific COTS processor.

Dennis Buede[33], 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

32. The INCOSE Systems Engineering Handbook, July 2000

33. Engineering Design Of Systems, Dennis M. Buede, Wiley, 2000

analysis that focuses 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[34] describes a trade as a formal decision making methodology used by integrated teams to make choices and resolve conflicts during the systems engineering process.

For the systems engineering 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 27 below, 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. For more information, refer to POH 7.7.2 at www.smc.sparta.com/golive/site16.

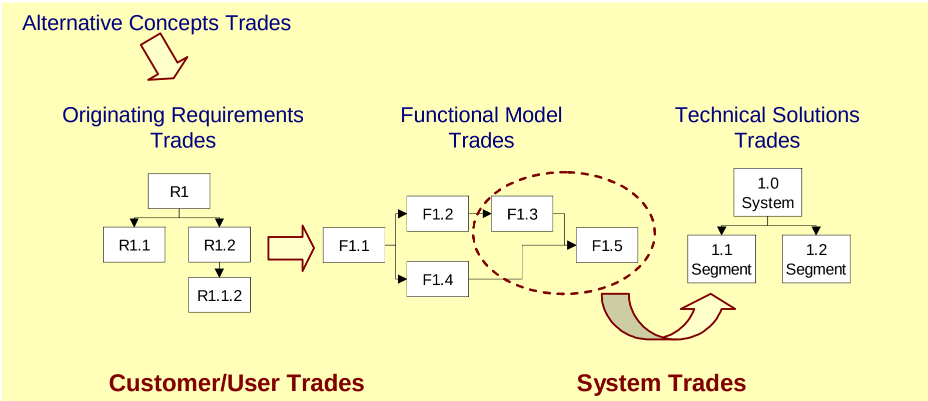


Figure 27. Trade studies to resolve architectures, requirements, and functional & design solutions

Cost Estimating

In any Systems Engineering 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.

34. Systems Engineering Fundamentals, DSMC, Jan 2001

Risk Management

Risk management is an important and often critical activity for DoD systems acquisition. It is the act or practice of managing risk. The Risk Management Guide for DOD Acquisition[35] defines risk as a measure of the inability to achieve overall program objectives within defined cost, schedule, and technical constraints and has two components: (1) the probability of failing to achieve a particular outcome and (2) the consequences of failing to achieve that outcome. For processes, risk is a measure of the difference between actual performance of a process and the known best practice for performing that process.

The acquisition process itself is designed, to a large degree, to allow managers to control events, or their consequences, that might adversely affect a program. In the past, many managers viewed risk as something to be avoided and required that any program that had risk areas be subjected to review and oversight. This attitude has changed. DoD decision makers recognize that risk is inherent in programs, and a goal of DoD acquisition is to study future program events, identify potential risks, and take measures to control them and ensure favorable outcomes.

There are many approaches that can be adopted for risk management. Any approach selected must be tailored for each specific project. Considerations on defining a risk management program include the acquisition strategy, program cost and schedule constraints, technology maturity, anticipated hazards, and many others. It is recognized that risk management is a program management responsibility. However, engineers are essential contributors to a successful risk management program. Some of the most frequently employed techniques include quantitative risk assessment, probabilistic risk analysis, fault tree analysis, and failure mode and effect analysis.

The GAO provides the following criteria necessary for good risk management[36]:

- Planned procedures – risk management is planned and systematic
- Prospective assessment – current and potential future problems are considered
- Explicit attention to technical risk
- Documentation – all aspects are recorded and data maintained
- Continuous process throughout acquisition

Successful risk management programs generally have the following characteristics[37]:

- Feasible, stable, and well understood user requirements and threats
- A close relationship with user, industry, and other appropriate participants
- A planned and structured risk management process, integral to the acquisition process
- Continual reassessment of program risks
- A defined set of success criteria for performance, schedule, and cost
- Metrics for monitoring effectiveness of risk reduction strategies
- Effective test and evaluation program

---

35 RISK MANAGEMENT GUIDE FOR DOD ACQUISITION, Fifth Edition, Department of Defense Acquisition University, June 200333

36 Technical Risk Assessment: The Current Status of DoD Efforts." Government Accounting Office, GAO/PEMD-86-5. Washington, D.C.: Government Accounting Office, 1986.

37 RISK MANAGEMENT GUIDE FOR DOD ACQUISITION, Fifth Edition, Department of Defense Acquisition University, June 2003

- Documentation

Program guidelines that ensure that management programs possess the above characteristics include:

- Assess program risk using a structured process
- Identify early and manage intensively those design parameter which affect cost, capability and readiness
- Use technology demonstrations/models/simulations and prototypes to reduce risk. (See GAO/NSIAD-99-162, page 68, for definitions of Technology Readiness Levels (TRLs) that can be used to assess the remaining risk.)
- Use test and evaluation as a means of quantifying the results of the risk handling process
- Include industry and user participation in risk management
- Establish a series of risk assessment review to evaluate the effectiveness of risk handling against clearly defined success criteria
- Establish the means and format to communicate risk information and to train participants in risk management including the program office.
- Obtain risk management buy-in at all appropriate levels of management

The Risk Management Guide for DOD Acquisition[38] describes a risk management program comprised of risk planning, risk assessment, risk-handling, risks monitoring, and documenting the overall risk management program.

Risk planning is the process of developing and documenting an organized, comprehensive, and interactive strategy and methods for identifying and tracking risk areas, developing risk handling plans, performing continuous risk assessments to determine how risks have changed, and assigning adequate resources.

Risk assessment is the process of identifying and analyzing program areas and critical technical process risks to increase the probability/likelihood of meeting cost, schedule, and performance objectives. Risk identification is the process of examining the program areas and each critical technical process to identify and document the associated risk. Risk analysis is the process of examining each identified risk area or process to refine the description of the risk, isolating the cause, and determining the effects. It includes risk rating and prioritization in which risk events are defined in terms of their probability of occurrence, severity of consequence/impact, and relationship to other risk areas or processes.

Risk handling is the process that identifies, evaluates, selects, and implements options in order to set risk at 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 associated cost and schedule. The most appropriate strategy is selected from these handling options. For purposes of the Guide, risk handling is an all-encompassing term whereas risk mitigation is one subset of risk handling.

Risk monitoring is the process that systematically tracks and evaluates the performance of risk-handling actions against established metrics throughout the acquisition process and develops further risk-handling options, as appropriate.

---

38 RISK MANAGEMENT GUIDE FOR DOD ACQUISITION, Fifth Edition, Department of Defense Defense Acquisition University, June 2003

Risk documentation is recording, maintaining, and reporting assessments, handling analysis and plans, and monitoring results. It includes all plans, reports for the PM and decision authorities, and reporting forms that may be internal to the PMO.

Example Approach to Implementing Risk Management

An initial task would be to prepare a risk management plan. The Risk Management Guide for DoD Acquisitions cited above, provides an example risk management plan for an acquisition program. An outline of a risk management plan is included in Appendix C3, Example Risk Management Plan Outline.

The risk management plan identifies assessment approaches and methods to be used. SMC does have a few tools that could be of benefit to establish and maintain a risk management program. See Chapter 5, What are the Systems Engineer's Tools? for further discussion on risk management tools.

Risk Assessment Matrix

A technical risk may be understood to be the probability and severity of loss from exposure to a hazard. If so, we would first start with the determination of quantitative or qualitative measures to ascertain the level of risk associated with a specific hazard. By combining the probability of occurrence with consequence, a matrix is can be created where intersecting rows and columns define a Risk Assessment Matrix. The Risk Assessment Matrix[*] forms the basis for judging both the acceptability of a risk and the management level at which the decision on acceptability will be made. See Figure 28.

Risk Identification and Characterization Techniques

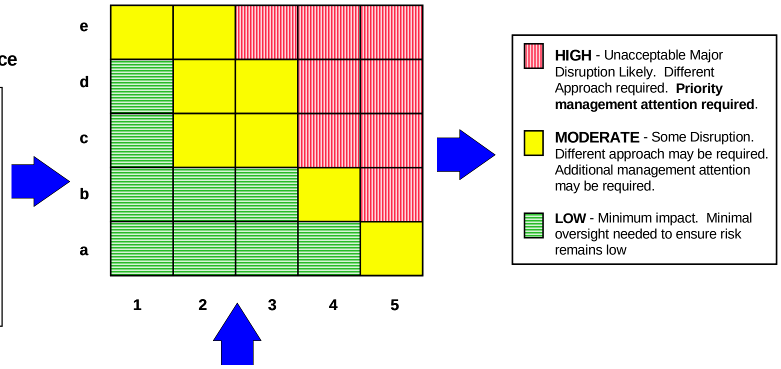
There are a number of disciplined processes for identifying, characterizing, and monitoring program risks. Characterization techniques for characterizing and monitoring risks are used to determine level of risk and consequence to the system. Of course, if a risk is characterized as moderate or high, a risk mitigation plan must be developed and implemented. Progress towards reducing risk is often monitored by a Risk Management Board or other technical management forum.

One particular characterization method relies on being able to assign numerical scale factors to hardware and software attributes such as complexity, maturity, and dependency to define the level of risk posed to the system. In addition, values are assigned to consequence attributes such as performance, schedule, and cost. Through a relative simple mathematical technique, an overall risk factor is calculated. Weighting factors may also be determined and applied (possibly using the Delphi method). When this approach is being used to compare risks between choices during a trade study, the weights can be equalized. Using an arbitrary scheme that can be modified by each program office, low, moderate, and high risk is assigned based on the range of the risk factor. A key to using this approach successfully is the ability to clearly define a quantitative set of characteristics for each attribute factor. This approach works best at the configured item level. Attribute definitions can be changed or modified as appropriate for each system.

* Note: The risk assessment matrix can be as simple as 3X3 or as large as 5X5. Furthermore, which blocks in the risk matrix are low, medium or high is a matter of discretion.

Probability of Occurrence

Level	Likelihood the Risk will Happen
a	Remote
b	Unlikely
c	Likely
d	Highly Likely
e	Near Certainty



Consequence: Given the Risk is Realized, What is the Magnitude of Impact

Level	Technical	Schedule	Cost	Impact on Others
1	Minimal Or No Impact	Minimal Or No Impact	No Impact	No Impact
2	Acceptable With Some Reduction In Margin	Additional Resources Required; Able To Meet Need Dates	Minor cost growth absorbable within budget	Some Impact
3	Acceptable With Significant Reduction In Margin	Minor Slip In Key Milestone; Not Able To Meet Need Dates	Cost growth exceeds budget, Mgmt. Reserves available	Moderate Impact
4	Acceptable with No Remaining Margin	Major Slip In Key Milestone Or Critical Path Impacted	Cost growth exceeds budget. Mgmt reserve is inadequate.	Major Impact
5	Unacceptable	Can't Achieve Key Team Or Major Program Milestone	Cost growth greatly exceeds budget. Large funding increase necessary	Significant Impact

Figure 28. A risk assessment approach

Risk Management Summary

How does a risk management approach work successfully?

- Over-all program management responsibility. Senior project management must be involved to properly allocate priorities and to get performer buy-in. If the program manager does not think risk management is his/her job, the risk management efforts will not be very effective.
- Lead and Train. The program manager and system engineer lead the process -- They train the team.
- Celebrate the risk reduction victories.
- Establish a risk-reducing culture on the project.
- Integrate risk management. Do not treat risk management as an add-on or parallel activity -- it is an integral part of the project activities.

Other good sources on Risk Management include:

- RISK MANAGEMENT GUIDE FOR DOD ACQUISITION, Fifth Edition, Department of Defense Acquisition University, June 2003,
- Defense Acquisition Deskbook – see http://deskbook.dau.mil/jsp/default.jsp,
- AFMC Pamphlet 63-101, Risk Management,
- SMC Risk Management Critical Process Assessment Tool (CPAT) – see http://ax.losangeles.af.mil/axe/axmp/CPAT/cpat.html,

- Air Force Pamphlet 91-215, Operational Risk Management (ORM) Guidelines and Tools, and
- DoD 4245.7-M, Transition From Development To Production.

Interface Management

Interface management is a Systems Engineering 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 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.

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. An outline of a typical interface control plan follows. For more information, refer to POH 7.7.4 at www.smc.sparta.com/golive/site16.

Interface Control Plan Outline

- Purpose – to establish the policies and procedures, and define the organizations and responsibilities necessary to achieve GPS interface control and management.
- Applicable Documents
- Definitions
    - Interface Control Action Sheet - a form used to document and communicate to affected participants and the relevant government agencies, the plan of action and associated schedules required to prepare or revise ICDS and/or resolve interface problems.
    - Interface Control Contractor Agency - The participating contractor or agency having leadership responsibility in initiating, defining, negotiating and maintaining Interface Control Documents/Drawings
    - Interface Control Document/Drawing - A document to establish, define and control the detailed interface design definition between two or more systems/segments/CIs.
    - Interface Control Steering Group - Top level management organization established to provide coordination, direction and control of interface activities at the Program Manager level
    - Interface Control Working Group - Technical body established to ensure accomplishment of the planning, scheduling, and execution of all interface activities.
- Organization - Interface control participants and hierarchy. Government agency(s), prime contractors, subcontractors, …
- Interface Types
    - Physical interfaces - Define the physical envelopes

  - Functional interfaces -- Define the performance requirements
  - Primary interfaces - A primary interface exists between two separately deliverable items (referred to as Configuration Items/CIs) when the mutual boundary area is not controlled by a single specification, when the interface is with systems outside the project (external interfaces), or at the discretion of the cognizant Interface Manager
  - Secondary interfaces -- A secondary interface is an interface that is defined by a single specification.
- Interface Management Structure
  - Interface Control Steering Group
  - Interface Management Group
  - Interface Control Working Group
  - Responsibilities
  - Interface Control Steering Group Responsibilities
  - Interface Management Group Responsibilities
  - ICWG Responsibilities
  - ICWG Functional Responsibilities
- Documentation
  - Interface Control Document/Drawing (ICD)
  - ICD/Specification Relationships
  - ICD Change/Revision
  - Interface Control Action Sheets
- ICD Development and Approval Process
- ICD Change Control Process
- Problem Resolution Process

In summary, the 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 documents (ICDs), or the like. 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.

Change Management/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 systems engineering 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.

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.

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.

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 which 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.

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.

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 should be managed as a part of the systems engineering 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.

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 is shown in Figure 29. 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. Descriptions of the reviews are found in SMCI 63-1201. For more information, refer to POH 9.3.2.7 and 9.3.2.9 at www.smc.sparta.com/golive/site16.

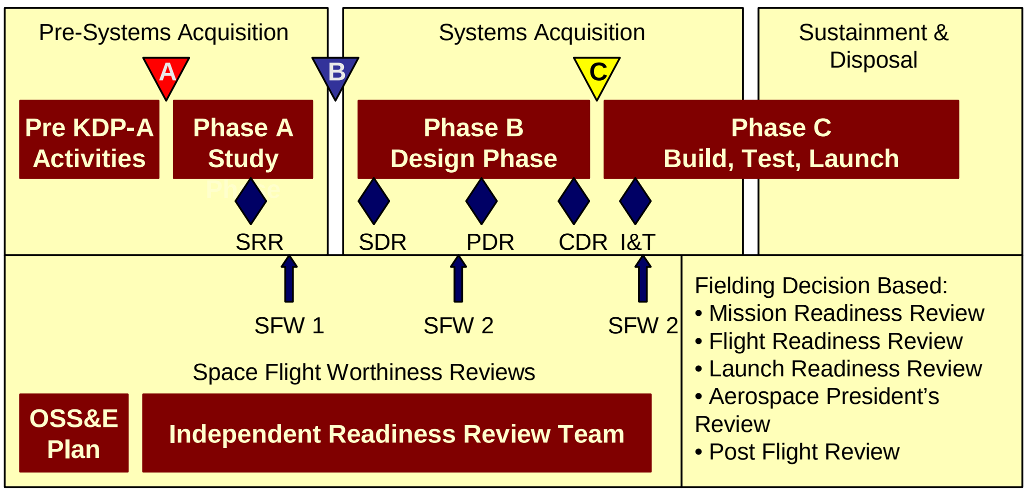


Figure 29. OSS&E assurance process

Chapter 5
What are the System Engineer's Tools?

Overview of Engineering Tools Usage by Acquisition Phase

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 systems engineering 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.

As systems definition and development commences, we usually continue to make use of the modeling and architecture tools used in the previous phase and significantly add to our modeling tool suite to support analyses and conclude technical/design solutions. During systems definition and development, we now put much more emphasis on requirements development, requirements and design analysis and validations, cost modeling and analysis, and certainly program/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 Diminishing Manufacturing Sources (DMS), configuration control/management, and possibly some of the M&S and analytical tools that would support system modifications.

Modeling & Simulation (M&S) Tools

The Chairman of the Joint Chiefs Of Staff Instruction, CJCSI 3010.02A, 15 April 2001 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 are used that accurately capture current and future Joint and Service capabilities, doctrine, and tactics.

DoD 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. This policy also establishes and defined the roles of the Defense Modeling & Simulation Office (DMSO). For more information, refer to POH Chapter 10 at www.smc.sparta.com/golive/site16.

Defense Modeling & Simulation Office (DMSO)

The DMSO serves as the official representative for M&S in DoD. This office coordinates with other federal agencies on M&S activities and serves as the focal point with the DoD. They are currently establishing a defense-wide capability to identify, and coordinate Joint Modeling & Simulation (M&S) activities and requirements.

Figure 30 represents the DoD Modeling & Simulation community. Obviously, M&S applications extend well beyond missions. Modeling and simulation tools have become important tools in the design process as a whole. Simulation has also increased in importance as a system integration and software verification tool. M&S are also used as highly effective training tools.

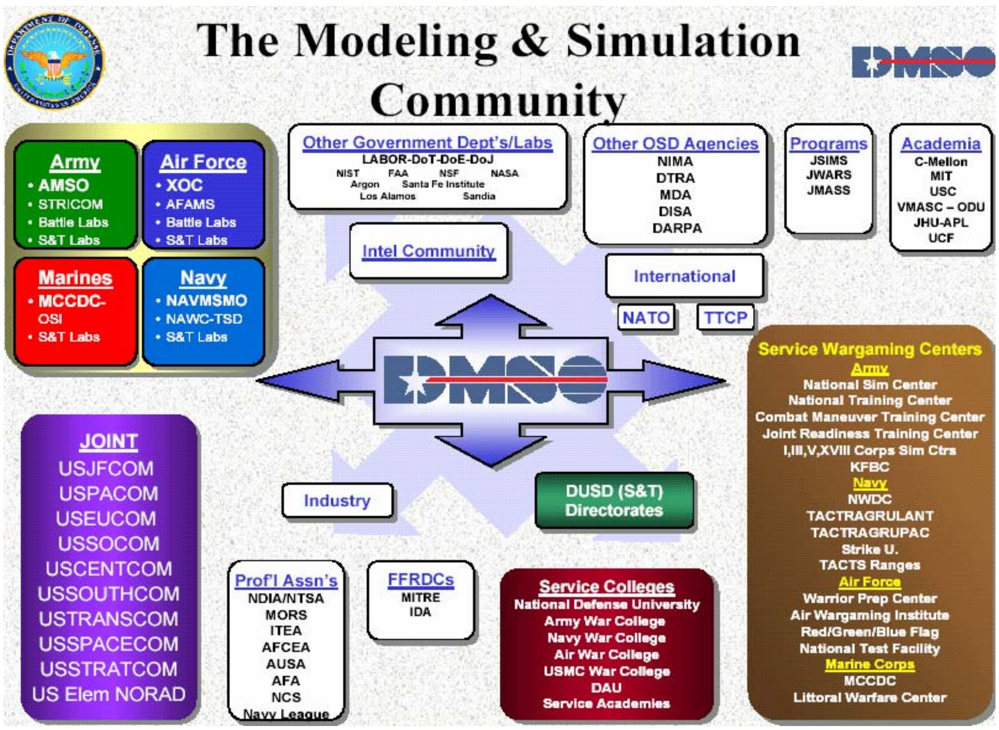


Figure 30. The DoD modeling & simulation community

High Level Architecture (HLA)

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.

HLA Federate Compliance Testing

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. For more information regarding HLA compliance testing see https://www.dmso.mil/public/transition/hla/compliancetesting

The Interim Defense Acquisition Guidebook, October 30, 2002, addresses Simulation-Based Acquisition (SBA) and Modeling and Simulation. SBA is the robust and interactive use of M&S throughout the product life cycle. This Guidebook counsels planning for SBA/M&S and making necessary investments early in the acquisition life cycle. We are also provided further guidance to use SBA and M&S during system design, system T&E, and system modification and upgrade.

- 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 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.

Describe, in the acquisition strategy, the planned implementation of SBA/M&S throughout program development, including during engineering, manufacturing, and design trade studies; and in developmental, operational and live fire testing applications.

Concept or System Architecture Tools

Architecting graphical representations to capture complex knowledge have become common practice to help define our weapons systems. The DoD's C4ISR Architecture Framework is one of many methods to capture an understanding or view of how a military force may be organized for particular mission or operational scenario. The principal objective of the C4ISR architecture framework is to define a coordinated approach for DoD architecture development, integration, and presentation. The framework is intended to ensure that architecture descriptions can be compared and relate across organizational boundaries. In February, 1998, the DoD Architectural Coordination Council mandated the use of this framework for all C4ISR architecture descriptions. For more information on the C4ISR Architecture Framework, see Appendix C11.

Many architecture tools are now available to support the development, assessment, evolution, or presentation of architecture representations. The International Council on Systems Engineering (INCOSE) provides an on-line survey service for 10 common systems engineering architecture tools at http://www.incose.org/tools. Common features of an architecture tool may include:

- Supports engineering modeling topology and notation. Particular models for system architecture analysis may include Functional Flow Block Diagrams (FFBDs), Physical Architecture Diagrams (PADs), Hierarchy Interface Diagrams (HIDs), Data Flow Diagrams (DFDs), as well as the C4ISR architecture suite of models.
- Provides bi-directional tracing between requirements, functions, processes, behavior and architectural components. Queries identify all affected components, items and products associated with any change in the architecture via the cross-reference links.
- Data Dictionary and component definitions are related to a defined architecture.

- Description of inputs, outputs and process are related to each architecture element.
- Relates operational concepts to architectural elements.
- Allows cost estimation through a spreadsheet feature and maps costs to architectural elements.
- Supports Multiple System Views. Provides architecture views from functional and object oriented (OO) perspectives. Examples: WBS, functional , physical, data flow, state diagrams
- Produces a view of interface connectivity.
- Supports various physical architectures.
- Supports various types (i.e. technology applications) of architectures: Mechanical, electrical, chemical, Information etc.

Analytical Tools

There are thousands of analytical tools that are used during the course of a space or satellite systems development effort -- too many to discuss here. Analysis categories commonly discussed at SMC include mission/architecture analysis, requirements analysis, design analysis, and systems analysis. For more information, refer to POH Chapter 10 at www.smc.sparta.com/golive/site16.

Architecture Analysis

There are a number of different types of tools that are used to define and select architectures. The most common type of architecture tool is that which is used to develop graphic representations or physical/functional/behavioral models of a system. Some of these tools also provide the capability to capture relationships between elements of the architectures as well as the attributes (e.g., weight, cost, size) of each element. For time-critical elements of an architecture, timeline analysis is performed using tools contained either the architecture software of other scheduling applications. Some more sophisticated architecture tools also provide dynamic modeling capabilities and may allow the user to define simulation environments.

For complex systems, architecture trades tools are often specifically and uniquely developed for a system to perform rigorous comparative performance, cost, schedule, and risk analyses between alternative architectures. Architecture alternatives tools analyses compare system architectures to determine the best value solution(s). Viable architectures are defined technically/parametrically and operationally and prescribed for each architecture. In addition, outputs of associated tools such as Cost As Independent Variable (CAIV) sensitivity analyses, reliability models, risk analyses and others are also candidate inputs for comparative analyses. These tools are often unwieldy but necessary to support architecture down selection of complex systems where there are thousands of parameters and many architectures to assess and compare.

If there are just a handful of architectures to be considered, a few tools are available on the market to assess alternative architectures. One example is the Architecture Tradeoff Analysis Method (ATAM) developed by Software Engineering Institute (SEI). This method is an architecture evaluation technique. The input to the ATAM consists of a system or product line architecture and the perspectives of stakeholders involved with that system or product line. The ATAM relies on generating use-case and change-case scenarios to assess the architecture. For more information on this method, see relevant documents published by the SEI[41] Modeling and Simulation.

Modeling and Simulation

Models and simulations allow you to study the 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 as time and understanding increase. The first step in modeling is identifying inputs that can be manipulated and determining what outputs result for the process or product under study. Then examine 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, your model is ready.

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 Further Reading section are good starting points.

Scenarios

Scenarios are often used in conjunction with models and simulations. A scenario describes expected situations in which the system might operate. Applying these situations to a simulation will allow you to see the system's response and change or augment the system to improve it. Using Monte Carlo techniques and multiple runs, it is possible to simulate closely the expected environment in which the candidate system will operate.

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 time-sequence 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 operations concept 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. A single chart is too confining for comprehensive information. Several charts typically show the overall operations and the details for each major operation. The information is then available for derivation of further requirements.

---

41 http://www.sei.cmu.edu

Requirements & Design Analysis Tools

For systems engineers, requirements analysis is a systematic process to define all of the constraints imposed by the operating environment. 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. Requirements analysis 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. Requirements analysis links with the Functional Analysis and Allocation to form the Requirements Loop of the System Engineering Process.

Tools used for the Missions and Environments requirements analysis often include many of the modeling and simulation tools mentioned above. Performance analyses utilize modeling and simulation tools as well as many analytical tools to determine the acceptable range of performance expected from a given design. Often statistical tools are employed to analyze and determine performance parameters.

As we proceed to the design phase, there is a need to define and validate design choices so analyses tools dominate. Design analysis tools are used to perform structural and thermal stress analyses, electromagnetic interference and compatibility analyses, worst case analysis, tolerance analyses, failure analyses – to name a few. Engineering subject matter experts use these tools to provide their specialized engineering contributions to the evolution and validation of the design. A systems engineer rarely has the opportunity to use this category of tools and assist in the design process. However, there are circumstances where the systems engineer will be expected to understand the limitations of a particular tool and assess the output or products of these analytical tools. For example, high risk aspects of a design may warrant independent assessments. Alternately system failures impacting mission capabilities often involve Government teams to assist in assessing design limitations.

A discussion of some of the tools available to the System Engineer and an introduction to their use is provided in this section. The analysis tools and techniques introduced in this section are listed in Table 8. The treatment here is not exhaustive. What is presented is a broad brush description of selected tools/techniques and their general application. The summary descriptions provided are meant only to permit you to assess which tools might be useful and to allow you to ask enough intelligent questions to get the detail you need. In no way will these paragraphs make you an expert in the application of these tools.

Table 8. Available systems engineering analysis tools

TOOL/TECHNIQUE	USE
Correlation Charts	A means of identifying the relationships between technical factors such as design features and system requirements
Value System Design	A technique for quantifying objectives and developing measures of utility
Functional Analysis Tools	Means of representing sequential functions and their interrelationships to assist in defining system solutions
Quality Function Deployment	A methodology for decomposing top-level Quality requirements
Pugh's Controlled Convergence	A peer process for optimizing system design
Models and Simulations	Software, hardware or combinations of both that allow elements of a system and its intended environment to be exercised in the laboratory
Scenarios	Sample situations representative of those to be encountered by the system, used to analyze the response of candidate configurations

TOOL/TECHNIQUE	USE
Joint Application Design	A technique for gathering inputs from all disciplines in a joint effort to establish the application design
Non-Customer Interactive Analysis	Techniques for finding available data on related system design and application
Allocations, Traceability & Decomposition	A methodology to establish a relational database that also aids in requirements allocation and decomposition
Baselining	Tools to document configurations as the basis to control design efforts and manage changes

Correlation Charts

Correlation charts are a graphic means of displaying interrelationships between various analytical factors (e.g., requirements and features). The three types discussed here are the cross-correlation chart, the self-interaction matrix, and the N x N chart.

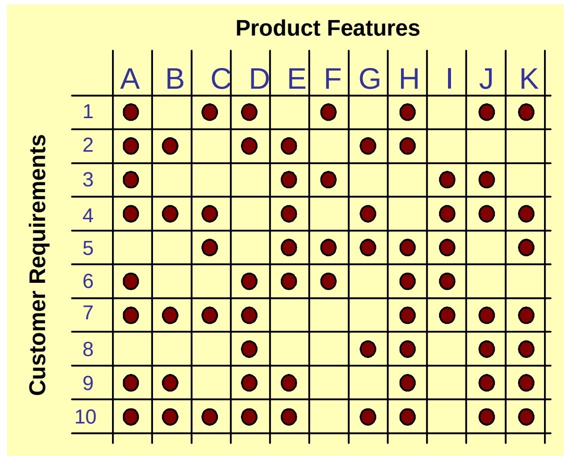


Figure 31. Cross correlation Charts–products features checklists

Cross-Correlation Chart: Figure 31 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 unneeded features are not included without being addressed. In Figure 31, 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 how important Requirement 8 is 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.

Self-Interaction Matrix: Figure 32 is a typical self-interaction matrix. 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 33, Requirement 1 affects or is affected by Requirements 2, 3, 5, 7, and 9. On the other hand, Requirement 4 interacts only with Requirement 9. 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.

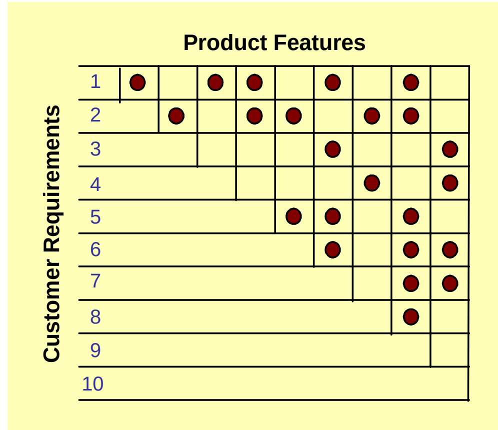


Figure 32. Self-interaction charts–showing which factors affect others

N x N Charts: These charts show both interfaces and relationships. Figure 33 is an example of an N x N chart used to show functional flow. The four functions represented form the diagonal of the chart. The block atop the second column shows that Function 1 feeds Function 2. Similarly, the blocks in the third and fourth column of the second row show that Function 2 feeds both Functions 3 and 4. The first block in the second row shows that Function 2 also feeds back to Function 1. Completing the picture, Function 3 feeds Function 4, and Function 4 feeds back to both Functions 1 and 2.

Function 1	F1 — F2		
F1 — F2	Function 2	F2 — F3	F2 — F4
		Function 3	F3 — F4
F1 — F4	F2 — F4		Function 4

Figure 33. Standard NXN (4X4) chart

In Figure 34, we have a more graphic representation of the interrelationships. Notice that the diagram shows two complete feedback loops — between Function 1 and Function 2, and between Function 4 and Function 2 (the feedback between Function 4 and Function 1 does not constitute a loop since there is no direct connection from Function 1 to Function 4). The analyst can see that Function 2 is complex since it is the intersection of two feedback loops. This will warn him to be extra careful in the design of Function 2 or to consider other interfacing that might eliminate this complexity. This type of chart is excellent to represent the states and modes of a system. See Appendix C10, States & Modes.

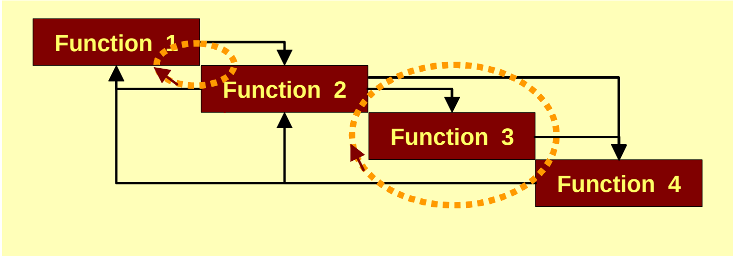


Figure 34. NxN showing graphical interelationships

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 five areas that define what is desired of the product: objectives; objective measures; criteria and weighting; and utilities.

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 its 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. Therefore, be sure 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." Systems Engineering seeks the greatest possible "worth" at an acceptable cost. A measure of effectiveness has these characteristics:

- Relates to performance.
- Simple to state.
- Complete.
- States any time dependency.
- States any environmental conditions.
- Can be measured quantitatively (if required, may be measured statistically or as a probability).
- 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 these factors:

- Performance: the probability that the product will perform its mission.
- Availability: the probability that a product is ready for use when needed.
- Dependability: the probability that a product behaves reliably in use.
- 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: 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 are continuous. They 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 are:

- The customer.
- Quality Function Deployment charts.
- Functions or behaviors.
- Measures of effectiveness.
- Measures of performance.
- Contractual costs.
- Contractual schedules.
- Manufacturing.
- Product Support.
- Project and organization objectives.
- Other considerations.

Weighting: 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.

You should be aware of some of the concerns with weighting methods. 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. If you depend entirely on the rating method you ignore the inherent uncertainties. Scoring should always be challenged, and recursion often occurs as the program matures.

Table 9 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 9. Criteria weighting–an example of comparison using weighted criteria

		Alternatives					
		1		2		3	
Criteria	**Wt**	**Score**	**Wt'd Score**	**Score**	**Wt'd Score**	**Score**	**Wt'd 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 Systems Engineering technique or tool, it is necessary to understand the underlying principles that contribute to Value System Design results. In the example in Figure 20, 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 curves are one means of checking sensitivity.

Utilities: 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. In the examples show in Figure 35, utility ranges from 0-5. Calculating loss is one way to plot a utility. In Figure 35 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 your own experience…do they seem reasonable?

Functional Analysis

Functional Analysis is one of the major Systems Engineering activities/processes. Two important benefits of Functional Analysis are that it discourages single-point solutions, and it aids in identifying the desired actions that become lower-level functions/requirements. Design teams typically include experts in the product field. Their knowledge makes for a better design. The drawback to that approach 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

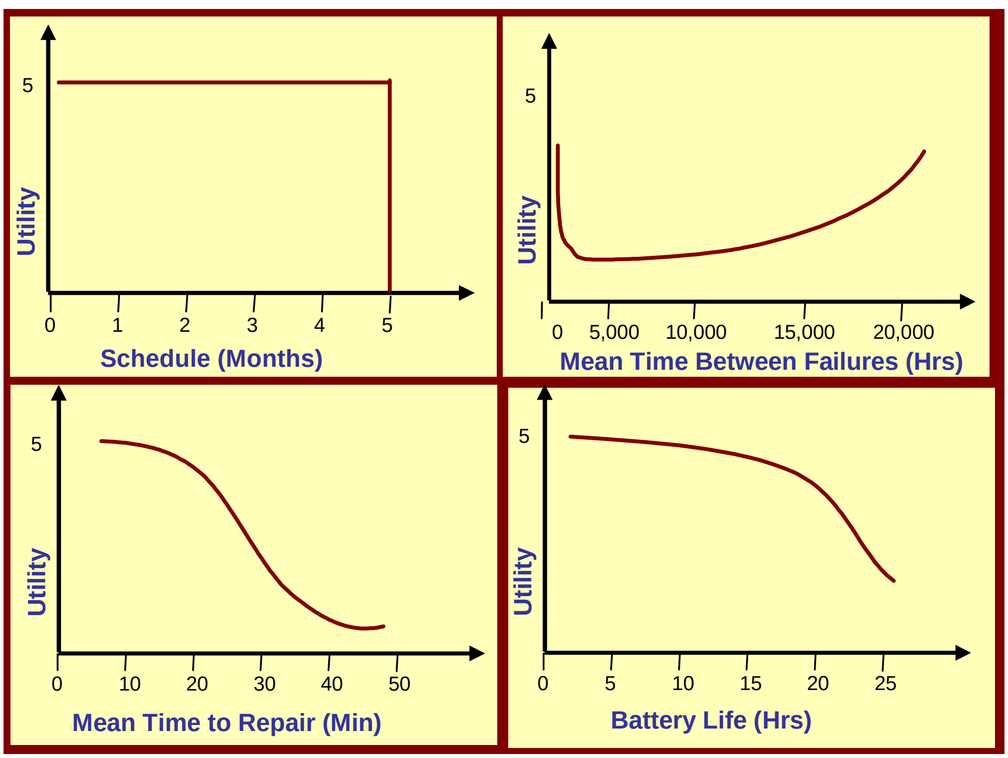


Figure 35. Utility curves–providing insight into criteria sensitivity

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. For more information and example approaches to performing functional analyses, see Appendix C5 Functional Analysis Techniques.

Function Analysis Limits: Unfortunately, function analysis by itself does not adequately describe a product completely. Function analysis does not describe system limitations, complete information flow, performance, or environments. However, it is a significant and essential tool is systems engineering activities. One method of relating these attributes to functions is the Quality Function Deployment (QFD) tool.

Quality Function Deployment

Quality Function Deployment (QFD) is an excellent tool for both planning and requirements flowdown. It combines elements of the cross-correlation chart and the self-interaction matrix. QFD is also useful in decomposing requirements to lower levels of the system. It integrates many of the systems engineering activities and tools. Interestingly, Quality Function Deployment began in Japan about the same time that J. Douglas Hill and John Warfield published a paper called "Unified Program Planning" in 1972 that describes linking correlation and self-correlation matrices. QFD might be based in systems engineering, but it integrates the planning and flowdown beautifully. It provides information including answers to:

- What is important to the customer?
- How can it be provided?
- What relationships are there between the "WHATs needed" and "how accomplished?"
- How much must be provided by the "HOWs" to satisfy the customer?

The most popular QFD tool (Figure 21) utilizes a series of connected correlation matrices to graphically represent interrelationships for analyzing requirements and allocating them to system elements. The graphic is called the "House of Quality" because the self-correlation matrix at the top resembles a roof. Individual areas within the graphic are called "rooms." The core of the house is a cross-correlation matrix which shows the relationship of the driving requirements (the WHATs) to the implementing requirements (the HOWs).

At the top-product level, the WHATs are taken directly from the customer. Information such as "must work a long time without breaking" is organized into categories. An importance rating is assigned to each demanded quality. Prioritizing is one of the most important activities in Quality Function Deployment. In identifying and weighting top-level WHATs it is imperative to ensure that they reflect the customer's/user's viewpoint and not internal biases. Failure to do so results in products that everyone in the project thinks are great, but may not serve user's needs. Beware of the "Edsel Effect." When you begin to develop lower level HOWs, internal customers (e.g., Manufacturing, Quality, Test, etc.) may be able to contribute to the method of implementation, but not at the top level.

With the WHATs organized and rated, the next step is to describe the HOWs. The HOWs are allocated and derived requirements. At the top-product level, the HOWs describe the product features and characteristics. The WHATs and HOWs are linked by a cross-correlation matrix. Initially there may be no one-for-one relationship between the WHATs and HOWs. The matrix allows you to see unfulfilled customer demands and also features that are expensive yet do not serve the customer. Eventually, there should be a HOW for each WHAT to ensure that all customer requirements are met.

The HOWs have their own self-correlation matrix at the roof of the house. It identifies how requirements might reinforce, oppose, or not affect each other. The HOWs are given target values for "how much." Engineering can then do a competitive assessment on the "HOW MUCH" against benchmarks. If the value of a HOW MUCH is initially unknown, record the measure but leave the value open until it can be established.

Figure 36 illustrates the organization of a sample Quality Function Deployment chart for an automobile. Charts should be kept small, 30 x 30 or less. Use the Pareto 80/20 rule (80% of the total requirements are reflected in 20% of the possible factors). Don't ask customers about things they don't know, but be sure to capture all relevant information. An incomplete chart does more harm than good.

In relating the WHATs and HOWs, the following symbols can be used to indicate the strength of the relationship:

⦿	Strong	Nominally valued at 9
○	Medium	Nominally valued at 3
△	Weak	Nominally valued at 1
Blank	None	Nominally valued at 0

The nominal value is an arbitrary weighting to allow comparison of features' worth.

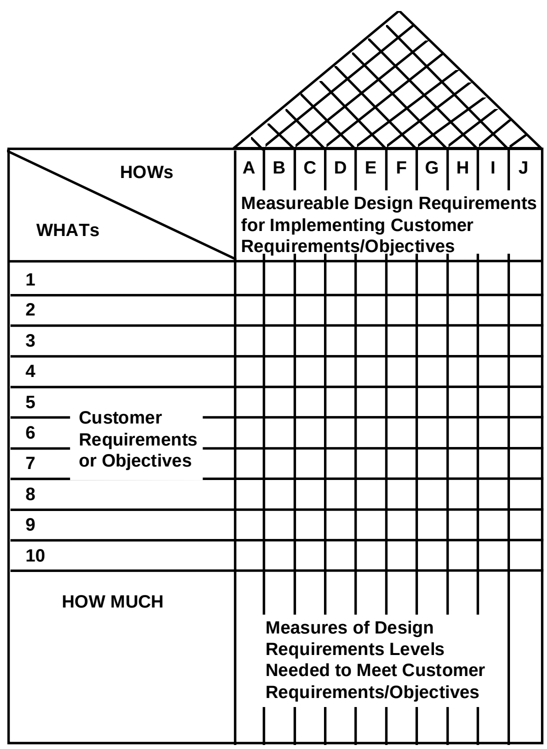


Figure 36. QFD representation–house of quality

Figure 37 has two new rooms. Relative Importance allows each WHAT to be assigned a value between one and five indicating how important it is to achieving the customer's perceived need. When this weighting is multiplied by the strength of the relationship to each HOW and then summed, the result recorded in the Weighted Importance lets you determine the contribution of each HOW to the overall satisfaction of the customer. In the sample chart, Manufacturing Hours and MTBF are the two principal drivers. This exercise shows how important it is to be talking to the right customer. Another group may consider comfort and luxury as the most important requirements, which would change the ratings and even cause replacement of some of the requirements.

Figure 37 also adds symbols to the HOWs' self-correlation chart in the roof. These are:

- ◉ Strongly support each other
- ○ Support each other
- X Adversely affect each other
- ⋇ Strongly oppose each other
- Blank Do not affect each other

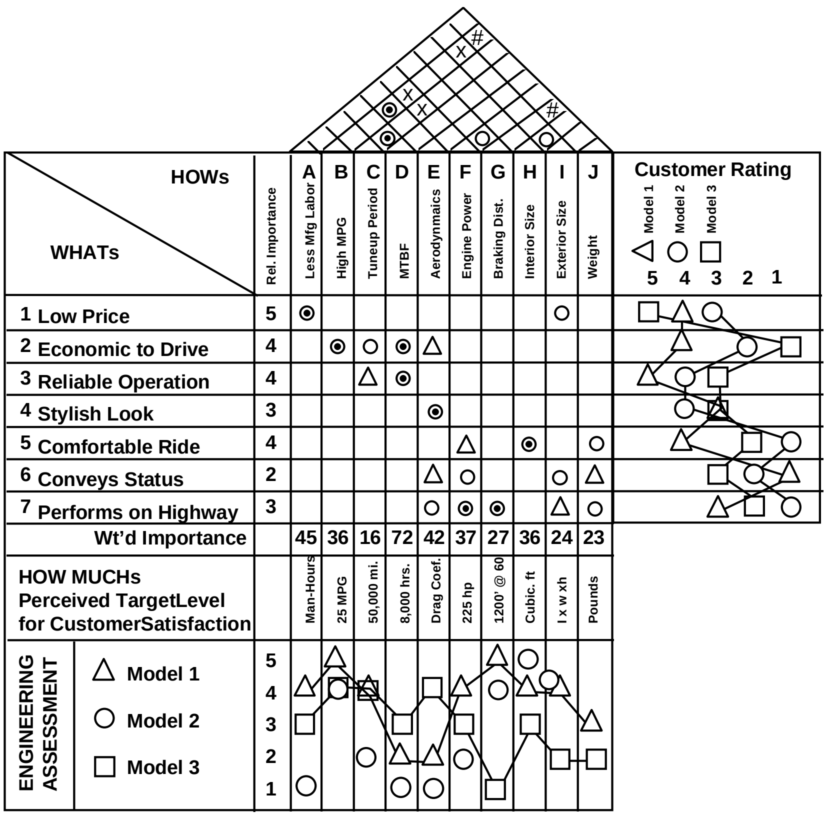


Figure 37. A full house of quality–added rooms for greater QFD definition

Efforts should be made to eliminate or reduce HOWs that strongly oppose. Such relationships might be used to direct trade studies and research.

Other rooms added in Figure 37 show engineering assessment of how well candidate approaches meet HOW goals and also how well candidates meet customer requirements (WHATs).

At the next lower hierarchical level, the WHATs come from the higher-level HOWs and HOW MUCH (Figure 38). The requirements flow down in this manner. Quality is deployed throughout the system from the voice of the customer speaking through marketing, engineering, manufacturing, and supporting organizations.

You can purchase software that organizes and prints the charts. There is a standard symbology for relationship and correlation's. If you are a first-time user, your goal might be to just get through the first time. Mastery of the chart technique takes practice.

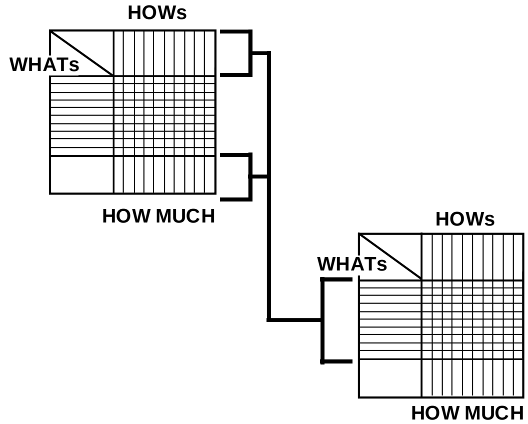


Figure 38. House of quality–new rooms

Pugh's Controlled Convergence

Evaluating alternatives requires a common means of measure. You must compare on a basis of equivalent standards. In addition to the weight scoring method, you can evaluate alternatives by the Pugh controlled convergence method[42].

Stuart Pugh of Great Britain developed a technique of selecting the best alternative by controlled convergence. In a sense, you are describing a benchmark and then improving on it. In the process of evaluating alternatives, you also generate new ones.

Pugh's controlled convergence method involves team effort. Pugh's experience is that the method makes it difficult for strong-willed people to push their own ideas for irrational reasons. The peer process is both analytic and synthetic in that both selection and creativity happen. Pugh believes that a disciplined approach leads to improvements in the product development.

The process is recursive, going through several phases to improve the initial concepts. A synopsis of the steps is:

1. Outline each alternative concept approach to the same level of detail.
2. Make a concept evaluation and comparison matrix (Figure 39) and enter approaches in the matrix.
3. Choose the criteria for the selection evaluation.
4. Choose a benchmark from the alternatives.
5. Comparing the alternatives to the benchmark, sticking to one

Criteria \ Concepts	Concept #1	Concept #2	Concept #3	
A	+	–	+	
B	+	–	–	
C	+	–	+	
D	S	S	S	
Total +	3	0	2	
Total –	0	3	1	

Figure 39. Concept evaluation and comparison matrix

42 Pugh, Stuart. Total Design: Integrated Methods for Successful Produce Engineering. Wokingham, England: Addison-Wesley, 1990

criterion at a time. Record an evaluation for each criterion/concept pair as follows:

+	decidedly better
-	decidedly worse
S	about the same

6. Abstain from modifying alternatives during the comparison.
7. Add pluses and minuses for each alternative.
8. Look at the negatives of the strongest alternatives. Can they be changed into pluses? Do not change the existing alternatives on the matrix, but add those modified as new additions to the matrix.
9. Look at the weakest alternatives. Can they be saved? If not, delete them from the matrix.
10. Look at all alternates for direction in improving the best, not worrying about numerical scores.
11. Repeating the steps until the design converges to a single acceptable and optimum solution.

Joint Application Design

Joint Application Design (JAD) is a common effort performed by the system users and system designers. It centers about a structured workshop called the JAD session. The workshop has a detailed agenda, a moderator/leader, and a scribe who records the agreed-upon requirements. The beauty is in the short time it takes to arrive at requirements, agreed to by the user/customer, and recorded in real time!

Credit for the JAD concept goes to Chuck Morris of IBM who started with it about 1977. In 1980, IBM Canada adapted and refined the tool. JADs have since spread outside IBM through training courses and are now used for all types of applications, including the original management information systems. JAD tasks include:

- Project definition:
  - Interviewing users.
  - Creating the participation list.
- Research:
  - Interviewing designers.
  - Learning about the system.
- Preparation:
  - Preparing the Working Document.
  - Preparing the session script.
  - Scheduling the meeting.
- JAD session
- Final Document:
  - Reviewing and updating the draft.

- Getting signatures on the document.

Using neutral, trained moderators and scribes works best. The key is preparation. For the meeting to be focused, the designers must have a good idea of the requirements for which they are looking. JAD sessions are an excellent way to converge diverse groups to an agreed specification or set of requirements. They can shorten the development time of a product dramatically by forcing all the key players into one room without disturbances

Non-customer Interactive Analysis

Not all requirements analysis is customer interactive. Other sources of requirements include:

- Literature research.
- Computerized databases.
- Trade journals.
- Trade shows.
- Market research.
- User characteristics databases (for example, anthropometrics).
- Forecasting.
- Modeling.

Requirements Definition/Traceability/Decomposition Tools

One of the most important tasks of the Systems Engineer is 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 the design/production chain is going to question the need for a particularly sticky requirement that he would just as soon not have to meet. He may be right! But even if he is, unless you know how the requirement originated you can't tell feel safe in granting relief unless you can determine its origin. Then too, he may be wrong!! 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 alleviate this problem.

Such tools usually employ relational databases. SMC has developed RDAV (Requirements Development and Validation) to more effectively perform requirements definition and change management. This is a Government owned tool developed by SMC, LAAFB, CA. 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.

Risk Analysis and Optimization

According to DoD 5000.2-R, The PM must identify the risk areas of the program and integrate risk management within overall program management. Systems Engineering evaluates the risk, or potential loss, of selecting an alternative as a solution. Even if a solution is the best technically, if the possible drawbacks cannot be accepted, the alternative must be discarded or modified. The need for risk analysis is not confined to the beginning of a project, but it is a continuing effort. The process of risk management is an organized method of identifying and measuring risk and developing, selecting, and managing options for handling these risks. The types of risk include, but are not limited to, schedule, cost, technical feasibility, threat, risk of technical obsolescence, security, software management, dependencies between a new program and other programs, and risk of creating a monopoly for future procurements.

Likely the systems engineer/risk manager will recognize the benefits of using risk management tools to assist in evaluating and managing program risks. SMC has developed a risk management tool that the systems engineer might consider: Program Supportability Management (PSM). The PSM, Shown in Figure 40, was initially developed for Integrated Logistics Support (ILS) risk management. However, it can be used effectively to manage any sort of risk. PSM works with milestones and allows users to assign probability and consequence values (as per the Risk Management Guide for DoD Acquisition) to each milestone event. These values can be combined, summarized and reported. The sample 5 x 5 Bubble Chart shown below gives a good overview of program health and allows managers to "drill" down into reports to examine the individual events that are most at risk.

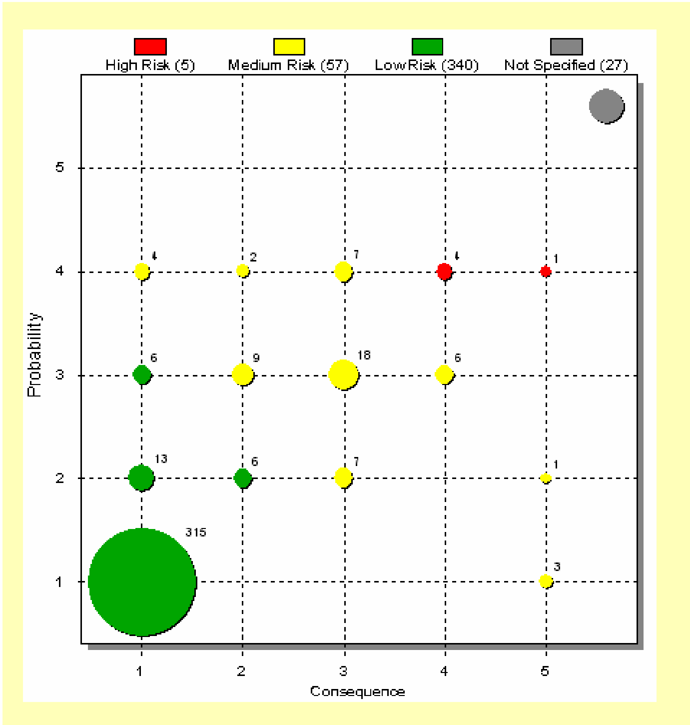


Figure 40. PSM bubble chart

The Requirements Design Analysis and Validation Tool (RDAV) tool also provides several risk management methods for both work breakdown structures (WBS) and requirements. Each of these methods allows one or more engineers to assign risk values to individual requirements or individual WBS elements, then allows these risk values to be combined for easy and clear reporting. RDAV has "Manage by Exception" features that allow summary charts to be "drilled down" to examine specific problem areas. Specific risk management methods supported by RDAV include:

- Requirements Risk Management. Probability and consequence values (as per the Risk Management Guide for DoD Acquisition) are assigned to each requirement. These values can be combined into reports and displays. The available charts give a good overview of program health and allows managers to "drill" down into reports to examine the individual requirements that are most at risk.
- Requirements Maturation. A well written requirements document has requirements that have been carefully examined and reviewed. RDAV has a Requirements Assessment

Matrix (RAM) feature that allows each requirement to be examined and rated by up to 12 different criteria. Each result is reported in the Requirements Assessment Matrix report. This report gives a clear assessment of problem requirements or disciplines that need more engineering attention. The baseline set of 12 criteria are included in Appendix C15. Requirements Evaluation and Acceptability Criteria.

- Structured Evaluation. Once a program has progressed to the point where is has a work breakdown structure (WBS) RDAV supports any sort of structured evaluation of all or a portion of that WBS. To initiate an evaluation one needs to decide:
  - The list of criteria by which each WBS element will be judged
  - The subsystem that is going to be evaluated (examples could be launch vehicle, spacecraft, spacecraft instrument package…)
  - How far down the WBS the evaluation should look.
- Once these three things have been decided, RDAV can build a complete evaluation matrix with work management features to help ensure all the evaluation work gets done. The results can be charted and reported in several formats. RDAV has been initialized to support Operational Safety Suitability and Effectiveness (OSS&E) evaluations.

Robust Design Versus 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.

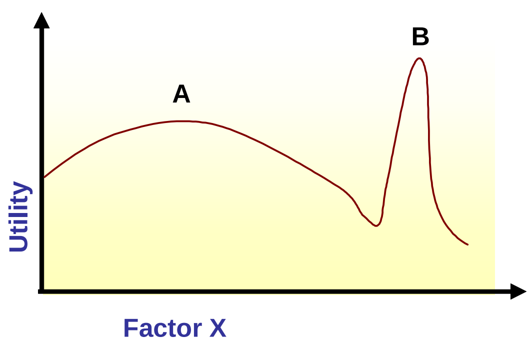


Figure 41. Robust design may be better than optimum

Optimal design is not always the best solution. Figure 41 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.

Analyzing Sensitivity

Analyzing sensitivity means the 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.

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 42. 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 42-b), you can extract 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.

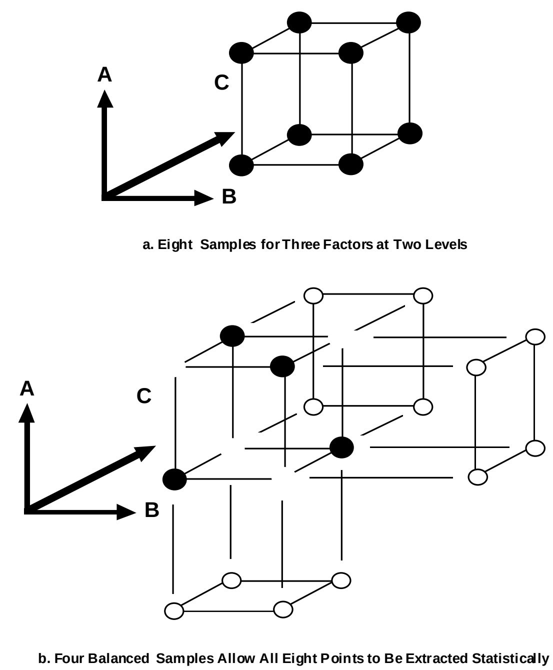


Figure 42. Balanced Experiments can Reduce Experimentation Costs and Schedule

Optimization Using the 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.

Systems Engineering minimizes loses 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 for the product the steps 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 Systems Engineering tool for cost reduction and increased customer satisfaction.

Requirements Change Management Tools

In the past, systems engineers generated much paper documentation to keep track of requirements and manage changes to requirements of complex space and launch systems. With the advent of the PC, system engineers began to use databases, spreadsheets, and other common office application software to perform requirements development and change management functions. These tools improved the efficiency to perform these activities, but still provided a restricted environment to a requirements development and change management environment. Hence, requirements management tools were developed to support multi-user collaborative environments, provide data exchange capability between other common and specialized tools, and make use of other computer technology improvements.

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. The International Council on Systems Engineering (INCOSE) provides an on-line survey service for 21 requirements management tools at http://www.incose.org/tools. Common features of a requirements management tool may include:

- Ability to capture and identify requirements – document enrichment / analysis, document change / comparison analysis, automatic parsing of requirements, semi-automatic and manual requirement identification, requirement classification.
- Ability to capture system element structure
- Provides traceability/requirements flow-down capability -- requirements derivation, allocation of performance requirements to system elements, bi-directional requirement linking to system elements, capture of allocation rationale, accountability, test/validation, criticality, issues, etc.
- Perform traceability analysis -- identify inconsistencies, visibility into existing links from source to implementation--i.e. follow the links, verification of requirements
- Perform configuration management tasks such as baseline/version control, track history of requirement changes
- Provide documents and other output media -- specification output, quality and consistency checking, status reporting.
- Interfaces with other selected engineering and office tools
- Provide sufficient system environment -- single user/multiple concurrent users, multiple platforms/operating systems, resource requirements
- Adequate User Interfaces
- Adequate support and maintenance – warranty, network license policy, maintenance and upgrade policy, on-line help
- Adequate Training.

System Analysis and Control Tools

System Analysis and Control is the welding that holds all the other Systems Engineering Process activities together, the steering wheel that gives them direction, and the map that shows where the process is going and where it has been. It is the activity that spans the whole life of the program. The 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 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.

Hence, the control/management tools assist the System Engineer in planning, tracking and measuring progress along the Systems Engineering process. Typical control/management tools include plans and schedules. e.g., the Integrated Master Plan (IMP), Integrated Master Schedule (IMS), and the System Engineering Management Plan (SEMP) and progress assessment techniques, e.g., technical performance measures (TPMS) and earned value measurements.

Given the description of the scope of this activity, you can see that tools are essential to effectively and efficiently perform systems analysis and control. Initially a program will select project management tools to perform planning and scheduling activities, tools to track actions and issues. There is also often a need to use tools to identify and assess program and technical risks and assess program progress via metrics collection and trend assessments. Data management tools are used to manage data deliverables from the program contractors. Configuration management tools are used to manage documentation baselines and changes to the baseline. For more information, refer to POH Chapter 10 at www.smc.sparta.com/golive/site16.

Process Capability Models

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. SMC is currently developing a process capability framework and performing appraisals on select Programs to better understand our strengths and weaknesses in regards to processes. To get the latest information, guidance, and tools of SMC engineering and acquisition processes, contact SCM/AXE.

There are a number of process capability models that have come into use over the last 30 years and there is much written on this subject. Since Carnegie Mellon's Software Engineering Institute (SEI) is currently contributing to defining the SMC process capability framework and performing appraisals, we will limit this overview to SEI's Capability Maturity Model – Integrated (CMMI™).

Software Engineering Institute Capability Maturity Model

You have likely heard of SEI's CMMI™ unless you are relatively new to the world of software development. A short discussion on this subject follows for those who are not familiar with CMMI. 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 five levels of maturity associated with this model.

Level 1. Initial Process

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. Repeatable Process

Basic project controls are in place to repeat project successes. An organization has in place, predefined software development procedures. No environment or commitment for process improvements. Edward Yourdon43 suggests the following processes for software: software Planning, software cost estimating, configuration management, and management commitment.

Level 3. Defined Processes

Organization wide software development processes are standardized. An Engineering Process Group is in place. Yourdon provides then following 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. Managed Process

This level emphasizes detailed quantitative methods to measure product and process quality. In other words, an emphasis is placed on quality to identify and correct deficiencies.

Level 5. Optimized Process

The organization continually invests in process automation and improvements. This level is measured quantitatively in terms of innovative process adaptations and new technologies. A rigorous defect causal analysis and prevention program is in place.

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. Marian Myerson [3] provides more discussion on several modified CMMs which do take into consideration personnel capability and performance.

The Carnegie Mellon Software Engineering Institute latest CMMI now combines 3 source models:

- Capability Maturity Model for Software (SW-CMM) v2.0 draft C,
- Electronic Industries Alliance Interim Standard (EIA/IS) 731, and
- Integrated Product Development Capability Maturity Model (IPD-CMM) v0.98.

There are 4 categories of CMMI Process Areas, which include Process Management, Project Management, and Engineering Support. Within each process area, goals and practices are defined as reflected in Figure 43.

---

43 Yourdon, Edward, Decline & Fall of the American Programmer, Englewood Cliffs, New Jersey, 1993

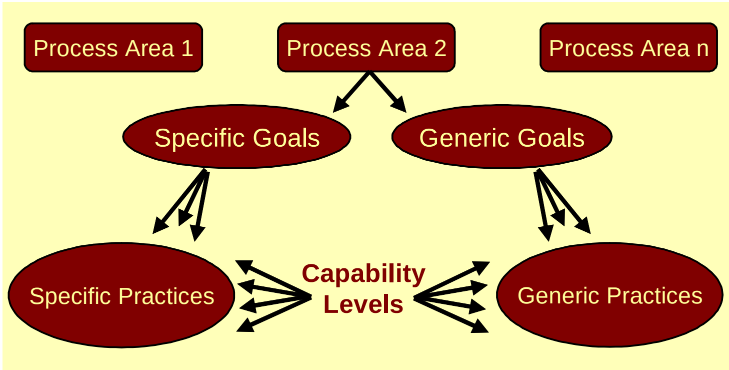


Figure 43. CMMI™ model components

Program/Project Management Tools

Though project management tools are briefly discussed above in the System Analysis and Control section, we will discuss them further here.

Project management tools abound in the marketplace. It is interesting to note that some of the project management application software that is available is user friendly and easy to use but are not necessarily the most expensive. In fact, the more expensive project management tools are full of features that would not commonly be of use for most SMC projects. Some characteristics of a project management tool follow:

- Process Management – Assists to develop project plans, schedules, and initiate best practices.
- Resource Management – Assists the Project Officer to plan and manage staff and material/facilities/equipment resources.
- Collaboration – Allows team members to receive assignments, communicate project status, and possibly provides for an electronic work environment for meetings, reviewing documents.
- Performance Measurement -- Project performance metrics such as Earned Value may be tracked and available for all stakeholders to review.

Most SMC programs and projects have not adopted integrated project management tools. Typically, a suite of tools are selected and used. The Comprehensive Cost and Requirement System (CCaR) is the tool of choice for financial management. The System Evaluation and Estimation of Resources (SEER) is often used for parametric cost estimating though there are other cost estimating tools that are used as well. For development of a project WBS and associations with scheduling of project tasks/activities, MS project is the most commonly used. However, Primavera or other more sophisticated (and costly tools) are sometimes used when more extensive program planning and scheduling capability is desired or provide stronger interface with other SMC project stakeholders.

SMC Project Officers will most likely to be constrained to use CCaR to establish and maintain their funding line(s). CCaR is a graphical financial management system designed to give Government Agencies a greater flexibility and control over their budgets. Some of the CCaR characteristics are provided below.

- Provides electronic coordination with e-mail interface to speed approval processing time.
- Allows for the creation of multiple coordination cycles. Coordination cycles are fully customizable.
- Tasks may be prioritized. Ensures most important efforts are given funding consideration first.
- Provides instant feedback on fundability of effort upon approval.
- Basis of Estimate submissions are mandatory before release for approval.
- CCaR allows attachment of Excel spreadsheet to document calculations.
- Estimate sufficiency is documented during the coordination process.
- CCaR supports creation of both Obligation and Expenditure forecasts.
- Forecasts are event-driven.
- CCaR allows easy tracking of Contracts. Tracks changes by Modification with focus on CLIN funding, Obligation amount, tracking of contract by ACRN/CLIN.
- Automatically updates budget obligations and expenditures.

For more information, refer to POH Chapter 10 at www.smc.sparta.com/golive/site16.

Tools Summary

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 maybe 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.

Chapter 6
What are the Companion Disciplines to Systems Engineering?

Because of the scope and complexities of our modern weapons systems, it is important from the outset to guard against unintended consequences such as unsafe operation, high failure rates, or electromagnetic interference (EMI). As a result, systems engineering must also provide for the integration of specialists in safety, reliability, component/design engineering, and many other discipline areas to help define and allocate the requirements, evolve and complete the design, and verify that the design satisfies the requirements. This chapter provides an overview of some of the functional specialties that systems engineers play a major role to integrate their activities and contributions. Of course the systems engineer's overall objective is to make certain that no critical omission has been made that will render the system less effective than it is intended to be. As with the preceding chapter on tools, it is not meant to be exhaustive. What is intended is to give you a feel for the disciplines and what you might expect from their practitioners.

Design

The SMC Systems Engineers may not often have the opportunity to work directly with the Design engineers. However, the relationship between the designers and the SMC systems engineer is much closer that some may realize.

Space systems development contractor Systems Engineers work hand-in-hand with designers. Following development of a good set of requirements and several reasonable system architectures, designers are well suited to assess how the requirements might be implemented and the risk associated with each approach. The systems engineers in this environment must be flexible and allow innovation. Brainstorming can be effective in stimulating both the systems engineers and the designers. As the designers begin identifying plausible technical solutions, requirements might come into sharper focus, requiring further refinement or modification. Trade studies and analyses using the tools of the previous chapter help select viable candidates and establish firmer requirements. The SMC systems engineer also plays into this process as the requirements evolve and the trade space on many are further investigated. Such requirements are usually recorded in System Requirements Documents (SRDs) or system specifications that initially may have only "TBDs" (To Be Determined), but which fill up as the design matures.

In modern systems, software is as important as hardware, and is recognized as a significant risk aspect of systems development. Usually greater system performance versatility and flexibility is realized if certain system functions are implemented through programmable processing. For this reason, IPTs ensure that the team includes a software design representative when system functions such as critical and complex sequencing of events, autonomous operations, autonomous fault management and other functions are required.

The Systems Engineer must be ever watchful of design for design's sake. Market analyses may uncover commercial products fully capable (or reasonably so) of fulfilling some system functional needs with little or no modification. Only in rare instances requiring unusual performance do engineers design their own power supplies or RF plumbing. But complete subsystems, such as receivers, may easily be adapted to the intended use with great savings in time and money. The use of COTS (Commercial Off The Shelf) equipment is becoming more popular in this era of declining funds for development. COTS often provides savings not only in development, but also in support areas such as data, training, provisioning support equipment, etc. Similar savings may be derived through use of NDI (Non-Developmental Items). These are system elements developed on other programs which can be used in original or modified form in the proposed system. As with COTS, NDI avoids many of the costs and headaches associated with a new design. Systems Engineers should actively search for COTS and NDI. The savings realized may eventually save the program itself! There is a caution that must be stated with regards to use of COTS and NDI. COTS items must be carefully evaluated to establish that they can really satisfy the requirements. Environments, life cycles, and overall reliability, for example, may not be met. Consider an evaluation similar to that used for "qualification by similarity" before baselining COTS or NDI.

Design Engineers are involved in most programs nearly as long as the Systems Engineer. Starting soon after concepts are first identified, they contribute throughout development and into the production phase. After deployment, designers are called upon to provide fixes for field problems and modifications as changing needs, environments, or threats surface. During this period, it is the Systems Engineer's responsibility to assess the system effects of any change, maintain tight configuration control, ensure that proper consideration is given to other disciplines affected, and oversee the introduction of the change.

The relationship between Systems Engineering and Design is close and generally well understood. Many Systems Engineers have extensive prior design experience and hence are conversant with both areas. For this reason the interface will not be belabored here.

Research & Technologies

Systems Engineers may interface with Research & Technologies organizations as users and/or patrons. As the requirements are understood, Research may be asked if there is anything in the pipeline that might provide advantages over present technology in accomplishing the required capabilities and functions. Assistance with the research & technologies experts often expands to literature searches, trade or scientific journals, trade or industry shows and seminars. In addition the research experts may be knowledgeable of work conducted elsewhere that might provide complete solutions, or at least clues to some of the pressing systems challenges.

On the other hand, Systems Engineering may commission a research project to determine the feasibility of a critical component or process. In commissioning research it is imperative that the requirements are clearly defined and a timetable is agreed upon for the results. Also, the requests must be practical in terms of performance and schedule. Also, consider the possibility of later infusion of updated technology. Often we design around an area requiring advanced technology and then incorporate the new research product later in the development, in subsequent production, or even retrofitting in the field. There are levels of risk associated with using new technology. Obviously the most risky level is depending on technology being developed by another program that has yet to begin. The least risky approach is for a given program to assume the development responsibility. It is very important that budget and schedule be coordinated with key program milestones along the way.

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. Appendix C9 contains summary descriptions of the TRLs.

Manufacturing & Producibility

One of the major goals of IPTs is to develop products that can be efficiently manufactured. For this reason it is essential to have early manufacturing representation on the team. Manufacturing can identify cost, schedule and production difficulties that can aid in the trade offs of requirements in the Requirements Loop and candidate approaches in the Design Loop. Interaction with Systems and Design Engineering can result in minor changes in system/subsystem/unit design that have major impact on the cost and ease of production. The roots of the Manufacturing Plan should be in IPT participation and the plan should grow in concert with the system design.

Often those things which enhance the producibility of a product also have beneficial impact on testing, reliability and support but not always. 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 even operation. Achieving balanced system design requires that the other disciplines in the IPT be recognized as important contributors to the finalization of manufacturing decisions. Manufacturing involvement grows from early design through production. They are also involved in spares manufacture, in modifications and in producing retrofit assemblies and kits.

Reliability & Maintainability

Many times you will see Reliability lumped with Maintainability (i.e., R&M). While these disciplines are related, interactive and often performed by the same personnel, their perspective is different. Reliability is directed toward assuring that the given design attains the longest possible continued operation (high Mean Time Between Failures — MTBF) and operating life. Maintainability is directed toward achieving the reliability inherent in the design through servicing and maintenance, and efficiently restoring the system to operation should failures occur.

Engineers working in the R&M field deal with a number of Reliability and Availability terms and concepts with which the Systems Engineer must be conversant. Reliability is the probability that a product will perform without failure over a stated period of time and under a given set of conditions. The inherent Availability (AI) of a product is a measure of the designed-in probability that the product is ready for mission use. It is based on the reliability of the product, reduced by factors related to the time required for maintenance actions (servicing, preventive maintenance, troubleshooting and failure repair). Operational Availability (AO) is AI further reduced by factors related to down times caused by such items as administrative delays (e.g., not having the right part or person available to complete a maintenance action) or longer than expected mission times. Inherent and operational dependability are similar terms used to measure the ability of a system to complete its mission once it starts. In space systems, dependability usually applies to the space element while availability and dependability can both apply to the ground element. AI and DI are essentially within the control of the Systems Engineers and Reliability and Maintainability engineers. However, they and the ILS engineers must work closely with the customer/user to assure that the AO/DO achieved is as close as possible to the inherent Availability/Dependability (AI/ DI). Appendix C6 provides an example of how a system engineer can apply these principles to a real world problem.

Reliability

Reliability and Availability/Dependability goals are usually specified in the contract or the user/customer requirements. Reliability engineers can review candidate approaches and give some indication to the SE of the relative chances of each candidate meeting the reliability goals. As requirements are firmed, Reliability can comment on the feasibility of requirements, techniques (redundancy, fault tolerance, HiRel, etc.) that must be employed to meet them, and the methodology and cost involved in verifying achievement through tests and demonstrations. This kind of information is essential to the SE in selecting viable system candidates. Consequently, Reliability should be involved as approaches are formulated and functional analyses are performed in the Requirements Loop. Their involvement increases in the detailed design phase and decreases as the system enters production. After deployment, Reliability monitors field reports to ascertain the need for changes to improve system reliability and/or fix areas where unexpected reliability problems occur.

Designing a reliable space-based system requires use of proven techniques and engineering discipline. They include the following:

- use of redundancy at the unit level,
- fault detection, isolation, and correction at the unit level,
- rigorous thermal control of electronic units,
- selection of electronic piece parts which are resistant to; degradation in the expected radiation environment to be encountered, and latch-up due to single event upsets,
- adequate de-rating of electronic piece parts for electrical stresses and radiation environments encountered,
- systematic approach to evaluating the design for potentially mission-catastrophic single point failure modes,
- adequate margins for wear out items and consumables, and
- adequate margins for structural and thermal loads.

There are standard reliability analysis tools and techniques available. Some examples include:

- reliability models, analyses, and predictions. The foundation of a reliability model is the reliability block diagram (RBD). It is a top down symbolic logic model generated in the success domain. Simple RBDs are constructed of series, parallel, and any combinations of series and parallel elements. Blocks may depict events or elements in a system. By applying appropriate probabilistic success functions to each block, an overall value for system success, over a defined time period, can be calculated. Mil-Hdbk-217 describes analytical approaches that may be used to evaluate system designs. It includes methods for applying electrical and thermal stress to electrical, electronics, and electro-mechanical (EEE) parts to further refine reliability models. Failure data in the handbook corresponds with recent statistics for EEE parts. Failure data for mechanical parts are more difficult to obtain. A potential source is the non-electronic parts reliability database (NRPD-25) collected and maintained by the Reliability Analysis Center in Rome, NY. Note: reliability models using Mil-Hdbk-217 data usually result in conservative predictions of mean mission duration.
- failure modes, effects, and criticality analysis (FMECA) as defined in Mil-Std-1629 – The FMECA process is a disciplined approach to identifying the failure modes of a system. It is a bottoms up tabular technique that explores the ways or modes in which each system element can fail and assesses the consequences of these failures. The FMECA also addresses the criticality and risk of each failure. Countermeasures can be defined and consequent reduction in risk can be evaluated. FMECA is a valuable tool

for cost and benefit studies, and to implement effective risk mitigation and countermeasures. Of particular interest are those mission catastrophic modes which my be the result of a single failure in the system. For each single point failure mode resulting in serious consequences to the system, a critical item control plan should be developed. The implementation of this plan should mitigate that failure mode.

- fault tree analysis (FTA) – It is a top down symbolic logic model generated in the failure domain. This modeling technique traces the failure pathways from a predetermined undesirable condition or event (top event) of a system to failures or faults that could act as causal agents. FTA includes generating a fault tree. It is very useful in graphically depicting the aggregate of failure modes for a system. It is also very helpful in identifying significant cut-sets and path sets. A cut set is any group of initiators that will, if they all occur, cause the top event to occur. A path set is a group of fault tree initiators, if none of them occur, will guarantee the top event cannot occur. It is particularly useful for high-energy systems (i.e., potential high severity events) to ensure that an ensemble of countermeasures adequately suppresses the probability of mishap. An FTA is a powerful diagnostic tool for analysis of complex systems and is used as an aid for design improvement.
- event tree analysis (ETA) – It is a bottoms up symbolic logic model generated in both the success and failure domains. This modeling technique explores system responses to an initiating challenge and enables assessment of the probability of an unfavorable or favorable outcome. The system challenge may be a failure or fault, an undesirable event, or a normal system operating command. The event tree presents all plausible system operating alternative paths from the initiating event. The ETA is particularly useful for analyzing command start or stop protective devices, emergency response systems, and engineered safety features.

Maintainability

Maintainability Engineers need a working understanding of Reliability concepts because they must build on the Reliability results in identifying Maintainability needs and approaches. Maintainability must work to satisfy the Availability and Dependability requirements. Prime among the factors contributing to Availability/Dependability is the MTBF, which establishes the frequency of need for corrective maintenance. Once a failure has occurred, Availability/Dependability is dictated by the amount of time necessary to return the system to operation (Mean Time To Restore Function — MTTRF). This in turn is affected by the time required to isolate the problem and to repair, switch to a backup, or replace the defective component(s). Rapid isolation is enhanced by the manner in which functional interfaces are drawn, by the inclusion of test ports for insertion and measurement of signals, and by the use of self-diagnostic or Built-In-Test (BIT). Some factors that reduce MTTRF also have a negative effect on MTBF (BIT usually adds components and numerous interfaces may increase connections, both tend to reduce Reliability). Such lower Reliability normally places additional stress on meeting Maintainability goals. The Systems Engineer must be aware of these tradeoffs and strive for a balance that approaches both Reliability and Maintainability targets.

Maintainability also has major interaction with Logistics and Support and in many cases may be handled under the umbrella of Integrated Logistics Support (ILS). Maintainability decisions greatly affect other ILS functions and likewise, some ILS constraints (expected deployment, isolation of locations, maintenance echelons, training of available personnel, etc.) may steer Maintainability approaches. Some common goals, such as modularity and BIT, may be mutually supportive. Others, such as commonality, the need for special tools or equipment, etc., may be disparate. Often the SE is called upon to make judgments as to which approach provides the proper blend of these goals.

Maintainability Engineering starts with a general review of requirements to ascertain there are no “show stoppers.” The Maintainability effort increases during the transition from the Requirements Loop to the Design Loop begins to decrease through the development effort, and usually ends after Maintainability goals have been demonstrated.

- The System Maintenance Concept and Maintenance Plan
- Designing Maintainable Space-Based Systems
- Maintainability Analysis Tools and Techniques.

Mass Properties

How much does it weigh? Always the big question. Just as in life, it is always easier to add weight than to reduce it. Over the life of a system’s development cycle, system engineering must carefully manage the weight budget and other mass properties of the system. Probably most critical is the throw weight of the space vehicle. But also important is the weight of transportable and mobile elements of the system. Will it fit into that C5B? Will all the equipment fit into the ISO container?

Weight estimates must be established early in the development cycle. For a new space element, it is wise to plan on having a 25 percent weight contingency at PDR and 15 percent by CDR. This is in addition to any contingency held by the program office for future capability growth. Of course these can be adjusted depending on the maturity of the hardware to be used by the system. Weight is a parameter that should be managed using a program level metric.

For the space element moments of inertia and center of mass are also important properties to be understood as the design matures, although a metric is usually not needed.

Environments & Survivability

AF Space Command and US Stratcomm place high importance on protection of space systems. The US warfighter is highly dependant on space systems to successfully complete his mission. Imagine a battlefield without the capability for early attack warning, protected communications, Intelligence, Surveillance, and Reconnaissance (ISR), GPS navigation, or weather prediction and reporting. This implies the need to design USAF space systems to operate under extreme space and terrestrial weather and in a weapons environment.

The natural environment of space is demanding. During launch the satellite must survive acceleration, vibration, acoustics, depressurization, thermal, radiated RF emissions, and separation shock. While on orbit temperature, geomagnetic radiation, solar flare particles, galactic cosmic rays, and orbital debris play together to degrade satellite performance and lifetime. During wartime, weapons effects become a significant driver. A high altitude nuclear detonation contributes significantly to the total radiation dose received by the satellite. In fact a yield in the 10s of kilotons can essentially use up the entire radiation lifetime of a low Earth orbiting satellite in the matter of several months. X-rays influence from a single device can destroy the electronics of any satellite in line of sight in a matter of seconds. Scintillation in the atmosphere, although not life threatening, can block communications for extended periods of time. There are many other man-made threats to space assets including: high and low energy lasers; kinetic energy kill vehicles; and ground and air based RF jammers to name a few.

The natural environment on the ground can be equally challenging. If a ground element is based in a fixed facility at Schriever AFB or Buckley AFB, the system deals with normal local weather conditions through normal environment management systems. In addition, buildings and antennas need to be designed to survive thunderstorms, wind storms, ice storms, snow, attack by wildlife, fungus, fog, blowing sand and dirt, lightning, and seismic events. If the facility is being attack during wartime, it may have to endure long enough to switch control over to an alternative facility or to survivable mobile elements. This implies protection and countermeasures against attack by Special Forces or terrorists, and airborne systems that may deliver nuclear or conventional, biological, or chemical weapons. If the facility is located OCONUS, attack by ground forces and local agitators is a consideration. Mobile, survivable ground elements may require protection measures from all types of threats depending on basing.

There are numerous mil-stds defining the threats resulting from natural and space weather environments. The “System Threat Assessment Report” (STAR) is developed by the National Aerospace Intelligence Center (NAIC) for each system to be fielded. This document includes the definitive sets of manmade threats to a space system. The system engineer must be familiar with this information and be prepared to make decisions on countering threats to the system under development. It is too costly, however, to have a countermeasure for every threat. It is systems engineering job to perform a threat evaluation and CAIV study to determine reasonable, cost effective countermeasures. An approach to threat evaluation and CAIV is outlined in C.10. Other approaches are certainly feasible.

Environmental, Health and Safety

The USAF takes seriously its legal obligations concerning Environmental, Health and Safety issues. It has developed and implemented for policy and use a documented instruction to ensure that all new, proposed and/or revised versions of space and missile weapons systems have been scrutinized and subjected to technical analysis for potential environmental impacts.

The Air Force Instruction (AFI) 32-7061 as promulgated by 32 CFR 989 complies with the National Environmental Policy Act (NEPA) and Executive Order (E.O.) 12114 and is the mandatory instruction to assess environmental impacts of new, proposed and/or revised space systems. This instruction is known as the Environmental Impact Analysis Process EIAP.This NEPA analysis process utilizes a systematic, interdisciplinary approach that is common to all good systems engineering decision making.

The EIAP begins with a Description Of Proposed Actions and Alternatives (DOPAA) usually prepared by the System Program Office (Program Office) in conjunction with the prime contractor included are provisions for an initializing document, AF Form 813 (Request For Environmental Impact Analysis) which in some cases may lead to a Categorical Exclusion (CATEX), or an Environmental Assessment (EA) or an Environmental Impact Statement (EIS) that includes a Record Of Decision (ROD) and may require a Mitigation Plan (MP). These requirements are to ameliorate or control/eliminate identified potential environmental impacts.

The EIAP process identifies the Program Office as the proponent for all environmental actions related to , proposed, new, and/or revised space systems. The Program Offices are encouraged to use Acquisition Civil and Environmental Engineering and Acquisition Safety and Health functions for support to facilitate meeting their NEPA requirements.

The EIAP analyzes air quality including installation compatible use zones facilities, water resources, safety and occupational health, hazardous materials and waste, biological resources, cultural resources, geology and soils, and socioeconomic issues including environmental justice. Analyses may include non ionizing/ionizing radiation, de-orbiting debris, noise and sound.

All Draft Acquisition Environmental Documentation must be coordinated through members of the Space and Missile Systems Center (SMC) Environmental Protection Committee for their review and comments. EA's and EIS's must be submitted to the public for review. All final documentation, CATEXs, EAs and EISs must be approved by the Chairperson of the SMC EPC and routed for further USAF action as prescribed by 32 CFR 989.

All prime contractors including sub-contractors are responsible for complying with all Federal, State and Local environmental laws and have the responsibility to furnish timely, revised and new changes effecting Hazardous Materials usage and possibly leading to Pollution Prevention

Recent changes to Space Acquisition Policy have mandated a Programmatic Environmental Safety and Health Evaluations (PESHE) for all Space Acquisitions identified under NSS-03-01 . The process for this task has been established through the development of a PESHE Guide, Milestone KDPA,B and C Check Lists, and chartering PESHE Working Groups (WG) that encompasses a NEPA Schedule and other programmatic ESH risks , all essential items for a PESHE document development. This work is highlighted by the development of a PESHE document that is integrated with ESH risk analysis .Experience has shown that PESHEWG are most effective when co-chaired by a Program Office member and a member of the Civil and Environmental Engineering staff to provide a sense of ownership for this document. An interface with AX Systems Engineering is essential during the concept and development stages of proposed space systems to raise issues of hazardous material use leading to possible pollution, safety and occupational health issues. As space weapons systems are developed, provisions to identify and assess environmental costs are evaluated along with cumulative environmental effects identified across all SMC programs. PESHE guides, charters, checklists risk analyses, NEPA requirements and the final PESHE product are all 'Living Documents' and need to be reviewed and updated on an as-scheduled /as-needed basis throughout the life time of the program.

Human Engineering–Personnel Subsystems

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 understanding of the intelligence, experience and training of the expected operators. The SE must include such analysis in candidate system selection and development. If you require an operator who is less than four feet tall, has three arms and no regard for bodily functions, your chances of widespread acceptance of the system are nil. Human engineer experts should have a chance to review and comment on requirements to identify any potential problem areas, however, their expertise is not regularly needed until specific designs begin to emerge. They are particularly helpful in the layout and arrangement of controls. They should also look at maintenance functions to ensure they are workable.

Training

Closely allied to Personnel Subsystems is the Training activity. Early system tests require a cadre of trained operators, so consideration of training must begin soon after PDR. What must be decided is the kinds of personnel required, types of training to be used and the need for any training equipment. In fact, some training equipment, such as mock-ups and simulators, may even be an integral part of the testing itself to tie down proposed operating procedures and control layout. As the system advances to Operational Test and Evaluation (OT&E) training requirements increase. Many of the operators and maintenance personnel who conduct these tests have little or no prior contact with the system, and must be brought quickly up to speed.

This requires prior planning of training and the training of trainers who can pass on the requisite information to the troops who will be doing the work. Training planning for OT&E and beyond should start about the time of CDR. Methodology should be established—lectures, computer based instruction (CBI), workshops, briefings, and demonstrations. Required resources must be identified—personnel (instructors and students), data (manuals, drawings, workbooks, and interactive computer programs), equipment (familiarization trainers, simulators, mock-ups, complete systems, and support equipment), and facilities (classrooms, labs, and computers). It is the System Engineer's responsibility to blend these requirements and activities with all the other activities clamoring for recognition and resources. Shot changing Training is short sighted. Unless your system works autonomously, you're going to need the cooperation of a knowledgeable user to accomplish the system's mission and maintain a satisfied customer.

Quality Assurance

Quality Assurance really does have multi-discipline implications. Often, there are overlaps in the assigned responsibilities of the quality experts. The Quality Assurance Manager often operates as a policeman to ensure that all contractually imposed specifications, standards, processes, and other design requirements are met. The Quality Manager also assists to establish internal conformance levels and criterion of acceptability for processes and products.

The focus of the Quality Control (QC) Specialist is typically to monitor activities and processes to ensure conformance levels and criterion of acceptability are being met. Quality Control also often manages quality processes such as the internal engineering change release system, calibration system, inspection systems and other quality related processes. Quality Control also acts as an in-process and final check of workmanship, test and overall production functions.

The Quality Engineer (QE) might be assigned to perform the quality related roles defined above. However, the QE expert engineering to determine and assess process variability as well as implement the use of statistical quality control and related QE/QC techniques. The Quality Engineer and Systems Engineer should investigate the use of these techniques as much as possible.

The QE also provides technical advisement on product and process improvements, supports root cause of anomalies, non-conformances, or test failure investigations, and provides recommendations on dispositions of non-conformances.

To generalize, Quality typically understands the 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, Quality assures that planning and design are carried out properly in the production system. Quality may also identify slight changes in the design of individual candidate approaches that could make their job easier and less costly. As software has attained a greater importance in modern systems, so to has the need for Software Quality Engineering and Control. The Systems Engineer 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 program from its inception through final disposal. Obviously then, Systems 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 Systems Engineer should confer with the user/customer to ascertain that they are willing to assume the costs involved in reaching certain goals. For more information, refer to POH 9.3.1.6 and 9.3.1.7 at www.smc.sparta.com/golive/site16.

Integrated Logistics Support

Logistics and Support, or more properly Integrated Logistics Support (ILS), contains ten elements which are mini-disciplines in their own right. These elements are:

Maintenance Planning (MP): the determination of what maintenance operations are required and the organizational level at which they will be performed.

Manpower and Personnel (M&P): the numbers of personnel and kinds of training required at each level to support the maintenance planning.

Supply Support (SS): provisioning and the development of data to support provisioning.

Support Equipment (SE): planning, design and development of equipment to test, handle and service the system in the field.

Technical Data (TD): 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 (T&TS): 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 (CRS): planning and support of efforts to maintain and upgrade fielded system software/hardware.

Facilities (FA): plan and implement the modification or upgrade of existing facilities, or the development of new facilities to support the system.

Packaging, Handling, Storage & Transportation (PHS&T): 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 (DI): 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.

In the past there was a tendency not to address logistics issues until the developed system was about ready to be deployed. After all, "why worry about how you are going to support it if you're not yet sure it will work?" In this era of limited resources, we have come to recognize that if we must expend nearly everything supporting systems already in the field, there will not be much left over for the new starts necessary to keep us competitive. Cost of ownership has tilted heavily toward support. ILS involvement in early design decisions has greatly reduced support costs and facilitated some of the recent initiatives invoking greater reliance on Commercial Off-the Shelf (COTS) items, Non-Development Items (NDI), and joint usage.

ILS personnel should be involved from the earliest requirements analyses through development, production, deployment and continuing operations. Because of their long-range viewpoint, logisticians tend to keep their options open. This characteristic is extremely helpful to the SE in identifying and avoiding potential problem areas.

We have not touched on many of the other functional discipline areas the systems engineer must interface with and integrate into the overall development and design efforts. The list can be quite exhaustive: Structural Engineers; Thermal Engineers; EMI experts; Test Engineers, Parts, Materials & Process Engineers; Corrosion Specialists; System Security; Operators; Users; and many more.

Chapter 7
Validation and Verification

Validation and Verification are important to the designer, the systems engineer, the program manager, and the customer. The V&V efforts provide direct evidence of progress towards ultimately meeting the customer's requirements. The V&V results 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 will satisfy his operational needs.

Table 10 lists some of the considerations involved in Validation/Verification planning. As to be expected, those associated with Validation tend to be oriented toward analysis, while those associated with Verification are oriented toward test. Planning should be documented in an integrated plan that identifies what will be validated and/or verified, the method(s) to be employed, and a schedule of events.

Table 10. Validation and verification considerations

Type	Description	Comment
Inspection	Examination by the senses (sight, sound, smell, taste, or touch) to determine requirements compliance.	Might use gauges or simple measures. - Some Physical Characteristics.
Analysis	Technical evaluation of data using logic or mathematics to determine compliance with requirements.	Used in Verification when given attribute is impossible or difficult/costly to test. Commonly used to extend test results beyond range of test.
Demon-stration	Un-instrumented test — compliance determined by observation (e.g., maintenance task performance time).	Used when compliance with requirement does not require measurement of a parameter. - Some aspects of Maintainability.
Test	Using procedures and test/measuring equipment to verify compliance with requirements.	Most recognized method of Verification; used also to support Validation analyses.
Process Control	Process control values accepted as evidence of requirements compliance. Process factors known, measured, and held to predetermined targets.	Use growing. Used to show dependability/ consistency of process results. Cannot be used to show that a system/component design complies with requirements.

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 requirements 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.

Table 11. Validation and verification control considerations

Validation	Verification
Analyses properly identified and defined prior to start	Document preparation properly supervised and approved.
Analysis results documented and cataloged for traceability	Documents are under configuration control.
Analysis results disseminated to design/ specialty disciplines	Non-conformance identified and analyzed.
Design decisions traceable to associated analyses	Measuring/test equipment calibrated to traceable standard.

Table 11 lists some of the considerations involved in Validation/Verification 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 particular designs 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. Systems Engineering is often the instigator and curator of the database/network combination.

Validation and Verification Methods

The five methods normally employed in Validation/Verification to establish compliance with requirements are listed in Table 12. Analysis is the primary method used in Validation while the others are used primarily in Verification. However, some testing is done to support Validation efforts, and occasionally Verification is accomplished by analysis where testing is difficult or prohibitively expensive, where expected operational environment cannot be created (all-out missile attack), or where testing costs can be effectively reduced because similar systems have been previously tested or have a history of use (compliance by similarity).

Inspections may be used to show compliance with some Physical Characteristics (size, weight, color), and along with Process Controls, may be used Quality and Manufacturing personnel to ensure/measure quality in production.

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 some of the “Ilities,” —Maintainability, Safety, Human Engineering, etc. The SE needs to know of them, but the special province and the major focus of the SE must be on analysis and test. Analysis has been discussed at length throughout this manual. Following are a few words about testing.

Testing

Testing increases confidence in meeting customer requirements and is part of overall risk reduction. Testing is of two types: a) developmental tests; and b) qualification/acceptance 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.

Validation/Verification testing of performance surfaces:

- Designs and design changes that fail to meet requirements.
- Manufacturing defects.
- Component failure or non-conformance.

Types of tests include:

- Burn-in and stress screening.
- Environmental testing.
- Variable and Go/No Go testing.
- Hierarchical level testing.
- Production assessment.
- Destructive and nondestructive testing.

Burn-In Tests are meant to get components past their infant mortality stage. By weeding out failures in this manner, the remaining test samples exhibit a higher level of reliability. Often burn-in is combined with temperature, vibration and vacuum stressing of the samples. Temperature cycling stresses the product to allow identification, replacement, or even redesign, of components that are particularly sensitive to thermal effects. Random vibration causes loose screws and parts to work free. Vacuum reduces outgassing of finishes that would otherwise contribute to contaminating surfaces in space. Such screening finds:

- Parts failure.
- Manufacturing defects.
- Marginal design.

Environmental Testing simulates the expected operating environment. In design proofing, the product may be subjected to levels greater than expected to prove design margins and as insurance that it can handle overstress conditions should they be encountered. Environments typically tested include:

- Atmospheric pressure or vacuum
- Temperature
- Solar radiation
- Rain
- Humidity
- Fungus
- Corrosive Atmosphere(s) (Salt fog)

- Sand and dust
- Explosive atmosphere
- Water immersion
- Acceleration
- Vibration
- Acoustic noise
- Shock
- Icing and freezing rain
- Electromagnetic Radiation

Variable testing records the actual value of the measurement.

Go/No Go compares the measured value against predetermined limits and determines whether or not the item is acceptable.

Hierarchical Level Testing refers to the evaluation performed at varying levels of assembly. As stated previously, it is more economical to surface problems at the lowest possible level. However, some problems that might not appear until elements are aggregated at higher levels. Such problems include tolerance build-up, race conditions, sneak paths, and stored energy hazards. For example, paralleling relays without isolation diodes will cause "chattering" relays because of stored charge in the relay coils. Hierarchical testing is especially important in software development programs.

Production Assessment Testing is done on sample products drawn periodically from production. This is an on-going verification of the production process. An example is verification of weight when the product is under configuration control. Production assessment is a check on processes and parts that might change over time, and otherwise go undetected.

Destructive Tests are performed to determine the stress level that causes the item to fail, and renders the test object unfit for its intended use. These tests must be done as samples, or nothing would be left to ship. Destructive tests are done on objects such as fuses, flash bulbs, and metallic materials.

Test and Evaluation

Test and evaluation is an adjunct of Validation. It provides confidence that the product will work before it is assembled. It identifies areas of risk for elimination or reduction during the product's development. It is also a validation of the Systems Engineering process. Test and evaluation generates information and knowledge on the developing product. It is deliberate and rational. System engineering compares and evaluates results of testing against the requirements. Test and evaluation includes physical testing, modeling and simulations, experiments, and analyses. "Test" means the actual testing of the product and components. "Evaluation" is the review and analysis of the information. The distilled information allows system engineering to:

- Define requirements.
- Manage the system engineering process.
- Identify risk
- Discover new alternatives.

- Improve product robustness.
- Find constraints.
- Decide the allocation of resources.

For more information, refer to POH Primer, Acq. Process at www.smc.sparta.com/golive/site16

Design for Testing

Efficient test and evaluation demands design for testing during product development. Systems Engineering must, in its design, address the need to:

- Collect data during the development process.
- Enable easy measurement, including:
    - Partitioning
    - Controllability
    - Observability
- Enable rapid and accurate assessment of the information.

Integrating Test and Evaluation

Test and evaluation must be integrated with the rest of the Systems Engineering effort. Documented decisions for test and evaluation are called the Test and Evaluation Master Plan (TEMP). The testing program in the TEMP must be consistent with the overall program management plan. The test program in the TEMP must provide the technical performance measurements required for review, audits, and risk management. Other documents integrated with the TEMP include the:

- 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 also. For example, supportability can and must be measured.

Reducing Integration and Test Time

In this era of cost competition and short schedules, reducing integration and test time has major benefits. Of all the considerations listed in Table 12, careful attention to the first two will provide maximum return. 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. It will also make you a hero to your test engineering, manufacturing, and quality associates. 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.

Table 12. Considerations for reducing integration and test time

Easily verifiable requirements.
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.

Chapter 8
Summary

Not Commandments. Not Rules. Not even Guidelines. Just 12 undeniable facts of Systems Engineering:

1. **It ain't over 'til it's over.** Systems Engineering is not a once-through-the-process-and-forget-it routine. It is a continuous, evolving, ever-different, program-tailored course that starts at program inception and progresses to product disposal after useful life is expended.

2. **There's no such thing as a stupid question.** Encourage your associates to question anything they don't comprehend. If they can't understand, they can't implement your ideas. Or they may implement them incorrectly. You need to rephrase your ideas in a way that all associates understand. Then too, occasionally a question brings up something you overlooked, and the stupid question saves you from disaster!

3. **Everybody's a QA man.** The product will be better if all are focused on product and process quality. Encourage everyone involved in the process to be on the lookout for potential problems. You can't be everywhere at once, and sometimes someone else's perspective uncovers items that may never occur to you.

4. **There's got to be an easier way.** This is the essence of all engineering. Be ever mindful of the power of innovation and open to the great revelation that leads you to the better mousetrap.

5. **There's no easy way to do anything.** Ah, a contradiction to the previous fact. What this says is there's no substitute for hard work and beware of treacherous shortcuts.

6. **Humans are best species on the planet to invent and use tools. Be human!** This is an admonition to make maximum use of available tools and look for ways to adapt them to the present use.

7. **We're all in this together.** The practice of Systems Engineering is an interdisciplinary process. The development of superior products requires that all specialties have timely knowledge of all design decisions and a chance to air their views.

8. **Listen to your instincts.** We've become so dependent on computers that we have a tendency to accept their outputs without challenge. Don't get so wound up in the process that you don't occasionally step back and look where you're going and where you've been. Also, weigh things in light of your experience and listen to your intuition. If something doesn't look right, it may not be!

9. **There's probably an upper limit to the number of times you should check your results, but you'll never reach it in any practical problem.** Check your inputs. Check your outputs. Check your checks. It's incredible how persistent some errors are. You may exorcise them out of one version of the program and find someone using a previous version. That slipped decimal point will come back to haunt you 'til they give you the gold watch.

10. **A good Systems Engineer is humble.** Don't think you have all the answers. If you do, you'll just end up with the same system you designed last time. Be open to suggestions.

11. **Yesterday's solutions may not be the answer, but it's the best place to start.** Don't get pulled into that "We did it this way last time 'syndrome'. On the other hand, the wheel is a

pretty basic device that has worked well for some time now and probably needs little re-engineering. Spend your energy where it will provide the most return. You usually have to have something that works before you can make something that works better.

12. **The good Systems Engineer knows when to kick it out the door.** There will always be a new device on the horizon that will give you 3 db more; or a new technique in development that will speed processing. But if it's not needed now to make your product meet requirements, don't hold off deployment to chase that extra bit of performance. If the product as is meets today's need, it should be in the customer's hands. Add the new item when the need arises. Besides, you may learn more from a few weeks of field experience than you might get in years of experiment and test.

Congratulations if you have gotten this far!! But don't think you've got this Systems Engineering thing completely in hand. This booklet was not intended as the last word on all things related to Systems Engineering. What we hoped to do was provide some background for those who are encountering Systems Engineering for the first time, or provide a reprise of the latest thinking for those who have been away from it for a while. We hoped we have peaked your interest to the extent that you seek additional information, and with the aid of some practicing professionals, implement the SE principles in your programs. The suggested additional readings in the Bibliography would be a good place to start in gathering more information. Your friendly librarian will also help you find suitable books, articles, and journals that might help and interest you. One of the main purposes of this booklet is to aid you in forming the right questions in your search for additional knowledge.

INCOSE

The International Council on Systems Engineering (INCOSE) is an organization formed to develop and enhance multi-disciplinary system development under the title of Systems Engineering. INCOSE is the one of the only professional associations dedicated entirely to systems engineering. INCOSE currently has more than a dozen working groups covering issues such as best practices, policy review, process description, tools, etc. INCOSE has national meetings annually with professional papers and other information of interest to systems engineers. INCOSE was created to:

- Foster the definition, understanding and practice of world class systems engineering in industry, academia, and government,
- Provide a focal point for dissemination of systems engineering knowledge,
- Promote collaboration in systems engineering education and research, and
- Assure the existence of professional standards for integrity in the practice of systems engineering.

So Many Interfaces, So Little Time

After reaching this point you're probably wondering how you'll ever be able to meet and deal with all these people on a daily basis. Fortunately, the problem is fairly bounded. While it's essentially true that the Systems Engineer has to interface with the world, he doesn't have to do it all the time and all at once. He will have a close long-term relationship with the designers, Quality and the logisticians, but they will get together to make interim decisions and then each will go off to perform the analysis, synthesis and design work necessary for the next set of decisions. Interfaces with the others are on a similar basis, but over a shorter period of time.

The most important point for the SE to understand is that each of these disciplines has a specific contribution to make and successful projects properly blend these inputs.

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).

Alternative Systems Review (ASR): A formal technical review, usually conducted early in the acquisition life cycle of a system or evolutionary increment or spiral, of (1) support to the Capabilities Need process, (2) an assessment of selected concept(s) relative to effectiveness in the intended environment, potential for growth, affordability, timeliness, and risk, and (3) the risks for the preferred system concept(s) that should be addressed during subsequent phases.

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. Addressed in DoDI 5000.2, NSSAP 03-01, and CJCSI 3170.01C. 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 systems engineering 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. verb–To formally approve a baseline.

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. See CJCSI 3170.01C and CJCSM 3170.01 for more detail.

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.

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.

computer software unit: A subdivision of a computer software component.

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.

Contract Work Breakdown Structure (CWBS): Work Breakdown Structure (WBS) prepared by the developer to capture all work planned under the contract or subcontract and that is accepted by the customer.

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 Analysis Requirements Document (CARD): The description of the salient programmatic and technical features of the program and the system it is to provide that is used by the teams preparing cost or schedule analyses or cost estimates. See DoDI 5000.2, 12 May 2003, Sections E6.1 and E6.2, DoD 5000.4-M, especially Chapter 1, or the NSSAP 03-01, especially Appendix 4 (AP4) in the draft of 15 Nov 2002.

cost engineering: The art of analyzing and estimating the cost of a design solution and relating those costs to the requirements.

cost goals, cost constraints, or cost requirements: The financial objectives or thresholds for the program or contract and their allocation to items. Often expressed in terms of development, design-to-cost (DTC), unit production cost (UPC), operations and support (O&S), and life cycle cost (LCC) thresholds, targets, or goals. Cost goals and requirements are a reflection that fiscal constraints are a reality in defense acquisition.

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.

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), Design-to-Cost: 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.

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.

external interface: A design constraint imposed on a system by another system or facility.

Family of Systems (FoS): A set or arrangement of independent systems that can be arranged or interconnected in various ways to provide different capabilities. The mix of systems can be tailored to provide desired capabilities, dependent on the situation. An example of an FoS would be an anti-submarine warfare FoS consisting of submarines, surface ships, aircraft, static and mobile sensor systems (some of which may be in space in the future), and space communications systems. Although these systems can independently provide militarily useful capabilities, in collaboration they can more fully satisfy a more complex and challenging capability: to detect, localize, track, and engage submarines.

Follow-On Operational Test and Evaluation (FOT&E): See "Operational Test & Evaluation (OT&E)."

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.

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 systems engineering 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.

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 criteria established by DoD 5000.2-R 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.

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 profile: The time-phased projection, usually in graphical form, of the values for a technical parameter.

planned value: The predicted value of a technical parameter at the planned time of measurement based on the planned profile.

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.

REQUIREMENT REFERENCE: A higher level requirement or an analysis, test, or other justification for a requirement, requirement allocation, or other architectural element. Abbreviated Req. Ref.

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.

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.

significant accomplishment criteria: Specific, measurable conditions that must be satisfactorily demonstrated before a significant accomplishment listed in an Integrated Master Plan (IMP) is complete and before work dependent on the accomplishment can proceed.

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 Development Plan (SDP): A management plan for the software development activities on a contract, usually prepared by the developer.

software, software product: See computer software.

solution, solution set: Products (including processes) and corresponding personnel manpower, skill levels, and specialized training that satisfy all requirements and balance performance, cost, schedule, and risk.

spares, spare parts: Maintenance replacements for replaceable parts, components, or assemblies in deployed items of equipment.

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.

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.

support function: Tasks to be performed to provide support for operations, maintenance, and training. The tasks include the acquisition and supply of spares, depot level maintenance, and the acquisition and maintenance of the facilities and selection and training of personnel to carry out the support function.

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 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 Design Review:

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 Manual (TM): Instructions for the deployment, operation, maintenance, training, support, and disposal of weapon systems, weapon system items, and support equipment. Technical Orders (TOs) that meet this definition may also be classified as Technical Manuals.

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. See Figure 1.7 of AFMC Instruction 63-XXX for more detail.

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.

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.

To Be Determined (TBD): When used in a Government controlled requirements document or Interface Control Drawing, an item that has not been determined and for which a determination is to be recommended by the Contractor (by a System Engineering or Integrated Product Team in which the Government participates) for final Government approval.

To Be Resolved (TBR): When used in a Government controlled requirements document or Interface Control Drawing, an item that is preliminary and for which a final resolution is be recommended by the Contractor (by a System Engineering or Integrated Product Team in which the Government participates) for final Government approval.

To Be Supplied (TBS): When used in a Government controlled requirements document or Interface Control Drawing, an item that has not been determined and for which a determination is to be formally supplied by the Government to the Contractor (though it may be studied by the System Engineering or Integrated Product Teams on which both Contractor and Government personnel participate).

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.

Work Breakdown Structure (WBS): A product-oriented hierarchical tree composed of the hardware, software, services (including cross-product tasks such as systems engineering), 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: many terms are defined in Appendix A.

ACAT	Acquisition Category
	Advance Concept Technology Demonstration
ADM	Acquisition Decision Memorandum
	Air Force Material Command
AFSCN	Air Force Satellite Control Network
APB	Acquisition Program Baseline
ASR	Alternative Systems Review
B	(1) Section of an RFP or model contract that specifies supplies or services and prices/costs
	(2) Blue evaluation ranking
BCD	Baseline Concept Description
BPPBS	Biennial Planning, Programming, and Budgeting System
C4I	command, control, communications, computers, and intelligence
C4ISE	command, control, communications, computers, intelligence, surveillance, and reconnaissance
C/SCS	Cost/Schedule Control System
C/SSR	Cost/Schedule Summary Report
CAID	Clear Accountability in Design
CAM	Cost Account Manager
CARD	Cost Analysis Requirements Description
CCA	Critical Capability Area
CDD	Capability Development Document
CDR	Critical Design Review
CDRL	Contract Data Requirements List
CE	Concept Exploration (Phase 0)
CE&D	Concept Exploration and Definition
CFSR	Contract Funds Status Report
CI	Configuration Item
CJCS	Chairman of the Joint Chiefs of Staff
CLIN	Contract Line Item Number

COTS	Commercial off the Shelf
CPAT	Critical Process Assessment Tool
CPD	Capability Production Document
CPI	Critical Program Information
CPR	Cost Performance Report
CRD	Capstone Requirements Document
CSOW	Contract Statement of Work
CWBS	Contract Work Breakdown Structure
DAB	Defense Acquisition Board
DAD	Defense Acquisition Deskbook
DEM/VAL	Demonstration and Validation (Phase I)
DIA	Defense Intelligence Agency
DID	Data Item Description
DoD	Department of Defense
DOT&E	Director, Operational Test & Evaluation
DOTMLPF	doctrine, organization, training, materiel, leadership and education, personnel and facilities
DPML	Deputy Program Manager for Logistics
DSAB	Defense Space Acquisition Board
DT&E	Development Test and Evaluation
DTC	Design to Cost (See also DTUPC, UPC)
DTUPC	Design to Unit Production Cost (See also DTC, UPC)
EA	evolutionary acquisition
EBB	Electronic Bulletin Board
ECP	Engineering Change Proposal
EDMs	engineering development models
EELV	Evolved Expendable Launch Vehicle
EMD	Engineering and Manufacturing Development (Phase II)
EVMS	Earned value management system
F	Section or an RFP or model contract that specifies delivery schedules
FCA	Functional Configuration Audit
FFBD	Functional Flow Block Diagram
FFP	Firm Fixed Price

FOC	Full Operational Capability
FoS	Family of Systems
FOT&E	Follow-On Operational Test and Evaluation
FRD	Functional Requirements Document
FRP	Full Rate Production
G	Green evaluation ranking
H	(1) Section or an RFP or model contract that specifies special contract requirements or provisions
	(2) High Risk
HSI	human systems integration
I	Section or an RFP or model contract that specifies contract clauses
ICA	Independent Cost Assessment
ICD	(1) Initial Capability Document
	(2) Interface Control Document
ICE	Independent Cost Estimate
ICWG	Interface Control Working Group
ILS	Integrated Logistics Support
IMP	Integrated Master Plan
IMS	Integrated Master Schedule
IOC	Initial Operational Capability
IOT&E	Initial Operational Test and Evaluation
IPA	Integrated Program Assessment
IPD	Integrated Product Development -- see IPPD
IPPD	Integrated Product and Process Development
IPT	Integrated Product Team
IRS	Interface Requirements Specification
ITAMP	Integrated Task and Management (or Master) Plan (ITAMP)
ITO	Instructions to the Offerors
J	Section of an RFP or model contract which lists attachments
JCIDS	Joint Capabilities Integration and Development System
JROC	Joint Requirements Oversight Council
JROCM	JROC memorandum
JTA	Joint Technical Architecture

KDP	Key Decision Point
KPP	key performance parameter
L	(1) Section of an RFP that includes the Proposal Preparation Instructions
	(2) Low Risk
LAAFB	Los Angeles Air Force Base
LCC	Life Cycle Cost
LFT&E	Live-fire Test & Evaluation
LOE	Level Of Effort
LRIP	Low-Rate Initial Production
LRU	Line Replaceable Unit
LSA	Logistics Support Analysis
M	(1) Section of an RFP that includes the evaluation criteria and factors
	(2) Moderate Risk
MDA	Milestone Decision Authority
MDAP	Major Defense Acquisition Program
MIL-Spec	Military Specification
MIL-STD	Military Standard
MIS	Management Information System
MNS	Mission Need Statement
MS	Milestone
MSSRP	Military Specifications and Standards Reform Program
MTBF	Mean Time Between Failure
NBC	Nuclear, Biological, and Chemical
NBCC	Nuclear, Biological, and Chemical Contamination
NDI	Non-Developmental Item
NSS	National Security System, National Security Space
NSSAP	National Security Space Acquisition Process
O&S	Operations and Support
OA	Operational Architecture (as in OA View)
OIPT	Overarching Integrated Product Team
ORD	Operational Requirements Document
OSD	Office of the Secretary of Defense
OTA	Operational Test Authority

OT&E	Operational Test and Evaluation (IOT&E and/or FOT&E)
PCA	Physical Configuration Audit
PCO	Procuring Contracting Officer
PDR	Preliminary Design Review
PESHE	Programmatic Environmental, Safety, and occupational Health Evaluation
PM	Program Manager
POH	Project Officer's Handbook
PPI	Proposal Preparation Instructions
PPBE	Planning, Programming, and Budgeting Execution process
PPBS	Planning, Programming, and Budgeting System
PWBS	Program or Project Work Breakdown Structure (WBS)
R	Red evaluation ranking
RAA	Responsibility, Authority, and Accountability
RCM	Requirements Correlation Matrix
RFP	Request for Proposal
SA	System Architecture (as in SA View)
SAF	Secretary of the Air Force
SDCE	Software Development Capability Evaluation -- see AFMC Pamphlet 63-103, Volumes 1 and 2
SDD	System Development and Demonstration
SDP	Software Development Plan
SDR	System Design Review
SEIT	System Engineering & Integration Team
SEMP	Systems Engineering Management Plan
SERD	Support Equipment Requirements Data (SERD)
SFR	System Functional Review
SMC	Space and Missile Systems Center
SOO	Statement of (Government) Objectives
SoS	System of Systems
SOW	Statement of Work
SPD	System Performance Document
Program Office	System Program Office

SRD	System Requirements Document
SRR	System Requirements Review
SRU	Shop Replaceable Unit
SSA	Source Selection Authority
SSS	System/Subsystem Specification
STAR	System Threat Assessment Report
SVR	System Verification Review
T&E	Test & Evaluation
TA	Technical Architecture (as in TA View)
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)
TDP	Technical Data Package
TDS	Technology Development Strategy
TEMP	Test and Evaluation Master Plan
TM	Technical Manual
TO	Technical Order
TPM	Technical Performance Measure
TRD	Technical Requirements Document
UPC	Unit Production Cost (See also DTC, DTUPC)
USD(AT&L)	Under Secretary of Defense (Acquisition, Technology, and Logistics)
USD(C)	Under Secretary of Defense (Comptroller)
USecAF	Under Secretary of the Air Force
WBS	Work Breakdown Structure (see also CWBS and PWBS)
Y	Yellow evaluation ranking

Appendix C–Templates and Examples

Appendix C contains templates, and some methodologies, that will hopefully provide a good starting point to perform common systems engineering tasks. Of course, the systems engineer must be mindful of his/her program unique requirements before selecting an approach to initiate a task.

Appendix C1–A Sample SEMP Outline

Title Page

Systems Engineering Management Plan

System Name or Identifier

Table of Contents

Scope

Purpose of the System

Summary and Purpose of SEMP

Relation to other plans and schedules such as the Integrated Master Plan (IMP), Integrated Master Schedule (IMS), and Earned Value Management System (EVMS)

The following statement: "This SEMP is the plan for the complete, integrated technical effort. Nothing herein shall relieve the Contractor of meeting the requirements of the Contract."

Applicable Documents

Government Documents to include contractual requirements documents or specifications

Non-government Documents to include any applicable from independent standards organizations

Corporate Documents

Systems Engineering Process and Responsibilities for its Implementation

Description of the Contractor's systems engineering process activities to be accomplished during the contract to include the iterative nature of the process application in the form of narratives, supplemented as appropriate by graphical presentations, detailing the contractor's processes and procedures for completing the systems engineering effort

Requirements Analysis

Functional Analysis and Allocation

Synthesis

Systems Analysis and Control to include Control and Manage to include trade studies, cost-effectiveness analyses

Risk Management

Configuration Management

Interface Management

Data Management

Technical Performance Measurements (TPMs) – initial list, criteria for changing the list, update schedule, responsibility for monitoring, and relationship to risk management

Technical Reviews and Audits

Description of products and results

Decision Database – describe development, implementation, life-cycle accessibility, and life-cycle maintenance including how traceability of the information will be accomplished

Specifications (or equivalent) and configuration baselines – describe development, measures of completeness, verifiability, traceability, and how and when controlled

Verification Planning – planning for verifying all requirements to include identification, configuration control, and maintenance of accuracy/precision of all verification tools

Organizational responsibilities, authority, and means of accountability for implementing the process under the Contract

Work authorization – methods for opening work packages under the EVMS, closure, and authorization of changes

Subcontractor technical effort – description of the level of subcontractor participation in the technical effort as well as the role of systems engineering in subcontractor and vendor selection and management

Transitioning Critical Technologies

Criteria for assessing and transitioning technologies

Evolutionary/spiral acquisition strategies

Integration of the Systems Engineering Activities

How management plans and schedules (such as the IMP and IMS) and the EVMS will be used to plan, organize, direct, monitor and control the systems engineering activities

Systems Engineering Tools

Approach and process for system integration and test

Additional Systems Engineering Activities

Notes

Glossary of terms used in the SEMP

Appendices – each appendix shall be referenced in the main body of the SEMP where the data would otherwise have been provided.

Appendix C2– "Tailored" WBS for a Launch & Satellite System

Level 1	Level 2	Level 3
Space System		
	Launch Vehicle	
		Stage I
		Stage II . . . n (as required)
		Strap-on boosters (as required)
		Fairing (shroud)
		Guidance and Control
		Integration, Assembly, Test & Checkout
	Space Vehicle	
		Spacecraft (bus)
		Payload (I . . . n)
		Orbit injector/dispenser
		Integration, Assembly, Test, and Checkout
	Ground Command, Control, Comm, and Mission Equipment	
		Telemetry, Tracking and Control
		External Communications
		Data Processing Equipment
		Auxiliary Equipment
		Facilities (Control, Communications, Mission)
		Integration, Assembly, Test and Checkout
	Systems Engineering/Program Mgt	(See Definitions below)
	System Test and Evaluation	Development Test and Evaluation
		Operational Test and Evaluation
		Mock-ups
		Test and Evaluation Support
		Test Facilities
	Training	
		Courseware
		Equipment
		Services
		Facilities
	Data	(See Definitions below)
	Peculiar Support Equipment	
		Test and Measurement Equipment
		Support and Handling Equipment
	Operational/Site Activation	
		System Assembly, Installation, and Checkout
		Contractor Technical Support
		Site Construction
		(See Definitions below for others)
	Flight Operations and Services	
		Assembly, Mate, and Checkout
		Mission Control
		Telemetry, Tracking, and Control
		Launch Equipment
	Storage	
		Planning and Preparation
		Storage
		Removal and Transportation
	Initial Spares	(See Definitions below)

WBS Definitions

Space System

The complex of equipment (hardware/software), data, services, and facilities required to attain and/or maintain an operational capability in space. This operational capability requires the ability to develop, deliver, and maintain mission payload(s) in specific orbit, which further requires the ability to place, operate, and recover manned and unmanned space systems.

Includes:

- launch vehicles, orbital transfer vehicles, payload fairings (shrouds), space vehicles, communications, command and control facilities and equipment, and any mission equipment or other items necessary to provide an operational capability in space.

Launch Vehicle

The primary means for providing initial thrust to place a space vehicle into its operational environment. The launch vehicle is the prime propulsion portion of the complete flyaway (not to include the orbital transfer vehicle and space vehicle). The launch vehicle may be single-stage or multiple-stage configuration.

Includes:

- the structure, propulsion, guidance and control, and all other installed equipment integral to the launch vehicle as an entity within itself,
- the design, development, and production of complete units (i.e., the prototype or operationally configured units which satisfy the requirements of their applicable specification, regardless of end use), and
- Sub-elements to the launch vehicle.

Note: All effort directly associated with the remaining level 3 WBS elements and the integration, assembly, test and checkout of these elements into the launch vehicle is excluded.

Stage I

The launch vehicle stage which provides initial lift-off propulsion for the complete launch vehicle (flyaway) and cargo.

Includes, for example:

- structure, propulsion, controls, instrumentation, and all other installed subsystem equipment integral to Stage 1 as an entity, and
- design, development, production, and assembly efforts to provide Stage I as an entity.

Excludes:

- strap-on units.

Note: All effort directly associated with the remaining level 3 WBS elements and the integration, assembly, test and checkout of these elements into the launch vehicle is excluded.

Stage II…n (as required)

The second and subsequent launch vehicle stages (if applicable) used to place a space vehicle into its operational environment.

Includes, for example:

- propulsion following separation of the first stage and subsequent stages (if applicable),
- structure, propulsion, controls, instrumentation, separation subsystems, and all other installed subsystem equipment integral to the stage as an entity, and
- design, development, production, and assembly efforts to provide each individual stage as an entity.

Excludes:

- strap-on units.

Note: All effort directly associated with the remaining level 3 WBS elements and the integration, assembly, test and checkout of these elements into the launch vehicle is excluded.

Strap-On Boosters (as required)

Solid or liquid propulsion assemblies that provide additional thrust or propellant to assist the launch vehicle in placing a spacecraft into its operational orbit if strap-on units are employed.

Includes, for example:

- complete set of strap-on units -- case, nozzle, igniter, tanks, mounting structure, cordage, etc., and
- design, development, production, and assembly efforts to provide the strap-on units as an entity.

Note: All effort directly associated with the remaining level 3 WBS elements and the integration, assembly, test and checkout of these elements into the launch vehicle is excluded.

Payload Fairing (Shroud)

The protective covering and equipment mated to the launch vehicle which protects the cargo (i.e., orbital transfer vehicle or space vehicle/orbital transfer vehicle combination) prior to and during the launch vehicle ascent phase.

Includes, for example:

- structure – the shroud structure, mechanisms and hinges,
- instrumentation – the hardware and software required to measure the environment and loads being experienced by the shroud during the ascent phase until shroud separation and deployment,
- separation subsystem – the sequencers, ordnance, and other necessary mechanisms to assure a successful shroud separation from the launch vehicle and cargo,
- power system – the necessary generation, storage, and distribution of electrical power and signals, hydraulic power, and any other power required by the shroud,
- thermal control systems – thermal paint, insulation, heat shield tiles, or any other active or passive means necessary to maintain appropriate temperature of the shroud and mission equipment within it, and
- integration, assembly, test and checkout.

Note: All effort directly associated with the remaining level 3 WBS elements and the integration, assembly, test and checkout of these elements into the launch vehicle is excluded.

Guidance and Control

The means (hardware/software) for generating or receiving guidance intelligence, conditioning the intelligence to produce control signals, and generating appropriate control forces.

Controllers may interface with the structure by actuating moveable aero surfaces or with the propulsion system to produce control reaction forces or may independently produce reaction forces for control.

If the design is such that electronics are packaged into a single rack or housing as an assembly, this rack or housing will be considered part of the guidance and control system.

Includes, for example:

- guidance intelligence system, computer, sensing elements, etc.

Note: All effort directly associated with the remaining level 3 WBS elements and the integration, assembly, test and checkout of these elements into the launch vehicle is excluded.

Integration, Assembly, Test, and Checkout.

In those instances in which an integration, assembly, test, and checkout element is used (Appendices A through G), this element includes all effort of technical and functional activities associated with the design, development, and production of mating surfaces, structures, equipment, parts, materials, and software required to assemble the level 3 equipment (hardware/software) elements into a level 2 mission equipment (hardware/software) as a whole and not directly part of any other individual level 3 element.

Includes:

- the development of engineering layouts, determination of overall design characteristics, and determination of requirements of design review,
- the set up, conduct, and review of testing assembled components or subsystems prior to installation,
- the detailed production design, producibility engineering planning (PEP), and manufacturing process capability, including the process design development and demonstration effort to achieve compatibility with engineering requirements and the ability to produce economically and consistent quality,
- inspection activities related to receiving, factory and vendor liaison,
- design maintenance effort,
- quality planning and control,
- tooling (initial production facilities, factory support equipment) including planning, design, and fabrication,
- administrative engineering,
- the joining or mating and final assembly of level 3 equipment elements to form a complete prime mission equipment when the effort is performed at the manufacturing facility,
- integration of software (including loading and verification of firmware), and
- conduct of production acceptance testing.

Excludes:

- all systems engineering/program management and system test and evaluation which are associated with the overall system.

Note: When an integration, assembly, test, and checkout element is utilized at lower levels of the contract work breakdown structure, it will be summarized into the next higher level equipment (hardware/software) work breakdown structure element and should never be summarized directly into a level 3 integration, assembly, test, and checkout element.

Space Vehicle

The satellite.

Includes:

- the structure, propulsion, thermal control, power and power conditioning, and all other installed equipment integral to the space vehicle as an entity within itself
- the design, development, and production of complete units (i.e., the prototype or operationally configured units which satisfy the requirements of their applicable specification, regardless of end use)
- Sub-elements to the space vehicle

Note: All effort directly associated with the remaining level 3 WBS elements and the integration, assembly, test and checkout of these elements into the space vehicle is excluded.

Spacecraft

The principal operating element of the space vehicle which serves as a housing or platform for carrying a payload and other mission-oriented equipments into space.

Includes, for example:

- structure, power, attitude determination and control, and other equipments characteristic of spacecraft, and
- all design, development, production, and assembly efforts to provide the spacecraft as an entity.

Payload

The equipment provided for special purposes in addition to the normal equipment integral to the spacecraft or reentry vehicle.

Includes, for example:

- experimental equipment placed on board the vehicle and flight crew equipment (space suits, life support, and safety equipment), and
- communications, displays and instrumentation, telemetry equipment and other equipments specifically to collect data for future planning and projection purposes.

Note: All effort directly associated with the remaining level 3 WBS elements and the integration, assembly, test and checkout of these elements into the space vehicle is excluded.

Orbit Injector/Dispenser

The function of placing orbiting objects in the planned orbital path.

Includes, for example:

- structure, propulsion, instrumentation and stage interface, separation subsystem, and other equipment necessary for integration with other level 3 elements.

Note: All effort directly associated with the remaining level 3 WBS elements and the integration, assembly, test and checkout of these elements into the space vehicle is excluded.

Integration, Assembly, Test, and Checkout

The integration, assembly, test, and checkout element includes all efforts as identified above to provide a complete space vehicle.

Ground Command, Control, Communications, and Mission Equipment

The ground hardware/software equipment used for communicating between control and tracking facilities, monitoring the health and status of space vehicles, commanding the space vehicle's hardware, and adjusting the space vehicle's orbit as required for space vehicle health or mission purpose. Two configurations for the ground command, control, communications and mission equipment are the parabolic dish-based antenna system and the phased array-based antenna system.

If a ground site has multiple antenna configurations, each will have its own separate command and control equipment, communications equipment, data processing equipment and test equipment.

Includes:

- the design, development, and production of complete units -- (i.e., prototype or operationally configured units which satisfy the requirements of their applicable specifications, regardless of end use), and
- sub-elements to the ground command, control, communications, and mission equipment.

Telemetry, Tracking and Control

The hardware/software elements that facilitate launch decisions and command and control of the aerospace vehicle.

Includes, for example:

- supplementary means for guidance of those aerospace vehicles not having completely self-contained guidance and control and means to command destruct, and
- control and check-out consoles, data displays, and mission records.

External Communications

The hardware and software components that allow the ground station to communicate with any external data link or source such as telephone (analog) lines, digital data lines, or nonsatellite radio receivers. While the terrestrial data lines may connect to radio of other satellite communications stations, the external communications subsystem ends where these links physically connect to the secure communications, modulation/demodulation (modem) or coder/decoder equipment.

Data Processing Equipment

The hardware and software components that provide the activities and means to condition data generated at the launch site or aboard the space vehicle, or data received from associated systems to accommodate the needs of command and control or mission data processing.

Includes, for example:

- central processing unit (computer), peripheral equipment, and the software required to operate the data processing equipment.

Auxiliary Equipment

The general purpose/multi-usage ground equipment utilized to support the various operational capabilities of the command and launch equipments.

Includes, for example:

- power generators, power distribution systems, environmental control, cabling, malfunction detection, fire prevention, security systems, and other common-usage items not applicable to specific elements of the ground based equipment.

Facilities (Control, Communications, Mission)

The special construction necessary to accomplish ground system objectives.

Includes, for example:

- modification or rehabilitation of existing facilities used to accomplish ground system objectives.

Excludes:

- installed operational ground equipment, and
- the brick and mortar-type facilities identified as industrial facilities – see Operational/Site Activation.

Integration, Assembly, Test, and Checkout

The integration, assembly, test, and checkout element includes all efforts as identified above to provide a complete ground system.

Systems Engineering/Program Management

The systems engineering and technical control as well as the business management of particular systems and programs. Systems engineering/program management elements to be reported and their levels will be specified by the requiring activity.

Includes:

- the overall planning, directing, and controlling of the definition, development, and production of a system or program including supportability and acquisition logistics, e.g., maintenance support, facilities, personnel, training, testing, and activation of a system.

Excludes:

- systems engineering/program management effort that can be associated specifically with the equipment (hardware/software) element.

Systems Engineering

The technical and management efforts of directing and controlling a totally integrated engineering effort of a system or program.

Includes, but not limited to:

- effort to define the system and the integrated planning and control of the technical program efforts of design engineering, specialty engineering, production engineering, and integrated test planning,
- effort to transform an operational need or statement of deficiency into a description of system requirements and a preferred system configuration,
- technical planning and control effort for planning, monitoring, measuring, evaluating, directing, and replanning the management of the technical program, and
- (all programs, where applicable) value engineering, configuration management, human factors, maintainability, reliability, survivability/vulnerability, system safety, environmental protection, standardization, system analysis, logistic support analysis, etc.

Excludes:

- actual design engineering and the production engineering directly related to the WBS element with which it is associated.

Examples of systems engineering efforts are:

1) System definition, overall system design, design integrity analysis, system optimization, system/cost effectiveness analysis, and intra-system and inter-system compatibility assurance, etc.; the integration and balancing of reliability, maintainability, producibility, safety, human health, environmental protection, and survivability; security requirements, configuration management and configuration control; quality assurance program, value engineering, preparation of equipment and component performance specifications, design of test and demonstration plans; determination of software development or software test facility/environment requirements.

2) Preparation of the Systems Engineering Management Plan (SEMP), specification tree, program risk analysis, system planning, decision control process, technical performance measurement, technical reviews, subcontractor and vendor reviews, work authorization, and technical documentation control.

3) Reliability engineering -- the engineering process and series of tasks required to examine the probability of a device or system performing its mission adequately for the period of time intended under the operating conditions expected to be encountered.

4) Maintainability engineering -- the engineering process and series of tasks required to measure the ability of an item or system to be retained in or restored to a specified condition of readiness, skill levels, etc., using prescribed procedures and resources at specific levels of maintenance and repair.

5) Human factors engineering -- the engineering process and the series of tasks required to define, as a comprehensive technical and engineering effort, the integration of doctrine, manpower, and personnel integration, materiel development, operational effectiveness, human characteristics, skill capabilities, training, manning implication, and other related elements into a comprehensive effort.

6) Supportability analyses -- an integral part of the systems engineering process beginning at program initiation and continuing throughout program development. Supportability analyses form the basis for related design requirements included in the system specification and for subsequent decisions concerning how to most cost effectively support the system

over its entire life cycle. Programs allow contractors the maximum flexibility in proposing the most appropriate supportability analyses.

Program Management

The business and administrative planning, organizing, directing, coordinating, controlling, and approval actions designated to accomplish overall program objectives which are not associated with specific hardware elements and are not included in systems engineering.

Includes for example:

- cost, schedule, performance measurement management, warranty administration, contract management, data management, vendor liaison, subcontract management, etc.,
- support element management, defined as the logistics tasks management effort and technical control, and the business management of the support elements. The logistics management function encompasses the support evaluation and supportability assurance required to produce an affordable and supportable defense materiel system, and
- planning and management of all the functions of logistics. Examples are:
    - maintenance support planning and support facilities planning; other support requirements determination; support equipment; supply support; packaging, handling, storage, and transportation; provisioning requirements determination and planning; training system requirements determination; computer resource determination; organizational, intermediate, and depot maintenance determination management; and data management.

System Test and Evaluation

The use of prototype, production, or specifically fabricated hardware/software to obtain or validate engineering data on the performance of the system during the development phase (normally funded from RDT&E) of the program.

Includes:

- detailed planning, conduct, support, data reduction and reports (excluding the Contract Data Requirements List data) from such testing, and all hardware/software items which are consumed or planned to be consumed in the conduct of such testing, and
- all effort associated with the design and production of models, specimens, fixtures, and instrumentation in support of the system level test program.

Note: Test articles which are complete units (i.e., functionally configured as required by specifications) are excluded from this work breakdown structure element.

Excludes:

- all formal and informal testing up through the subsystem level which can be associated with the hardware/software element, and
- acceptance testing.

Note: These excluded efforts are to be included with the appropriate hardware or software elements.

Development Test and Evaluation

This effort is planned, conducted and monitored by the developing agency of the DoD component. It includes test and evaluation conducted to:

- demonstrate that the engineering design and development process is complete,
- demonstrate that the design risks have been minimized,
- demonstrate that the system will meet specifications,
- estimate the system's military utility when introduced,
- determine whether the engineering design is supportable (practical, maintainable, safe, etc.) for operational use,
- provide test data with which to examine and evaluate trade-offs against specification requirements, life cycle cost, and schedule, and
- perform the logistics testing efforts to evaluate the achievement of supportability goals, the adequacy of the support package for the system, (e.g., deliverable maintenance tools, test equipment, technical publications, maintenance instructions, and personnel skills and training requirements, etc.).

Includes, for example:

- all contractor in-house effort,
- all programs (where applicable), models, tests and associated simulations such as wind tunnel, static, drop, and fatigue; integration ground tests; test bed aircraft and associated support; qualification test and evaluation, development flight test, test instrumentation, environmental tests, ballistics, radiological, range and accuracy demonstrations, test facility operations, test equipment (including its support equipment), chase and calibrated pacer aircraft and support thereto, and logistics testing, and
- avionics integration test composed of the following:
    - test bench/laboratory, including design, acquisition, and installation of basic computers and test equipments which will provide an ability to simulate in the laboratory the operational environment of the avionics system/subsystem
    - air vehicle equipment, consisting of the avionics and/or other air vehicle subsystem modules which are required by the bench/lab or flying test bed in order to provide a compatible airframe avionics system/subsystem for evaluation purposes
    - flying test bed, including requirements analysis, design of modifications, lease or purchase of test bed aircraft, modification of aircraft, installation of avionics equipment and instrumentation, and checkout of an existing aircraft used essentially as a flying avionics laboratory
    - avionics test program, consisting of the effort required to develop test plans/procedures, conduct tests, and analyze hardware and software test results to verify the avionics equipments' operational capability and compatibility as an integrated air vehicle subsystem
    - software, referring to the effort required to design, code, de-bug, and document software programs necessary to direct the avionics integration test.

Operational Test and Evaluation

The test and evaluation conducted by agencies other than the developing command to assess the prospective system's military utility, operational effectiveness, operational suitability, logistics supportability (including compatibility, inter-operability, reliability, maintainability, logistic requirements, etc.), cost of ownership, and need for any modifications.

Includes, for example:

- initial operational test and evaluation conducted during the development of a weapon system,
- such tests as system demonstration, flight tests, sea trials, mobility demonstrations, on-orbit tests, spin demonstration, stability tests, qualification operational test and evaluation, etc., and support thereto, required to prove the operational capability of the deliverable system,
- contractor support (e.g., technical assistance, maintenance, labor, material, etc.) consumed during this phase of testing, and
- logistics testing efforts to evaluate the achievement of supportability goals and the adequacy of the support for the system (e.g., deliverable maintenance tools, test equipment, technical publications, maintenance instructions, personnel skills and training requirements, and software support facility/environment elements).

Mock-Ups

The design engineering and production of system or subsystem mock-ups which have special contractual or engineering significance, or which are not required solely for the conduct of one of the above elements of testing.

Test and Evaluation Support

The support elements necessary to operate and maintain, during test and evaluation, systems and subsystems which are not consumed during the testing phase and are not allocated to a specific phase of testing.

Includes, for example:

- repairable spares, repair of repairable, repair parts, warehousing and distribution of spares and repair parts, test and support equipment, test bed vehicles, drones, surveillance aircraft, tracking vessels, contractor technical support, etc.

Excludes:

- operational and maintenance personnel, consumables, special fixtures, special instrumentation, etc., which are utilized and/or consumed in a single element of testing and which should be included under that element of testing.

Test Facilities

The special test facilities required for performance of the various developmental tests necessary to prove the design and reliability of the system or subsystem.

Includes, for example:

- test tank test fixtures, propulsion test fixtures, white rooms, test chambers, etc.

Excludes:

- brick and mortar-type facilities identified as industrial facilities.

Training

Deliverable training services, devices, accessories, aids, equipment, and parts used to facilitate instruction through which personnel will learn to operate and maintain the system with maximum efficiency.

Includes:

- all effort associated with the design, development, and production of deliverable training equipment as well as the execution of training services.

Excludes:

- overall planning, management, and task analysis function inherent in the WBS element Systems Engineering/Program Management.

Courseware

Distinctive deliverable end items of training courses, assigned by either a contractor or military service, required to meet specific training objectives.

Includes, for example:

- operational training courses, maintenance training courses, and other training courses.

Excludes:

- training equipment.

Equipment

Distinctive deliverable end items of training equipment, assigned by either a contractor or military service, required to meet specific training objectives.

Includes, for example:

- operational trainers, maintenance trainers, and other items such as cutaways, mock-ups, and models.

Excludes:

- training courseware.

Services

Deliverable services, accessories, and aids necessary to accomplish the objectives of training.

Includes:

- training course materials; contractor-conducted training (in-plant and service training); and the materials and curriculum required to design, execute, and produce a contractor developed training program, and
- materiel, courses, and associated documentation (primarily the computer software, courses and training aids).

Excludes:

- deliverable training data associated with the WBS element Support Data.

Facilities

The special construction necessary to accomplish training objectives.

Includes, for example:

- modification or rehabilitation of existing facilities used to accomplish training objectives.

Excludes:

- installed equipment used to acquaint the trainee with the system or establish trainee proficiency, and
- the brick and mortar-type facilities identified as industrial facilities.

Data

The deliverable data required to be listed on a Contract Data Requirements List, DD Form 1423.

Includes:

- only such effort that can be reduced or avoided if the data item is eliminated,
- (government-peculiar data) acquiring, writing, assembling, reproducing, packaging and shipping the data, and
- transforming into government format, reproducing and shipping data identical to that used by the contractor but in a different format.

Technical Publications

Technical data, providing instructions for installation, operation, maintenance, training, and support, formatted into a technical manual. Data may be presented in any form (regardless of the form or method of recording). Technical orders that meet the criteria of this definition may also be classified as technical manuals.

Includes, for example:

- operation and maintenance instructions, parts lists or parts breakdown, and related technical information or procedures exclusive of administrative procedures, and
- data item descriptions set forth in categories selected from the Acquisition Management Systems and Data Requirements Control List (DoD 5010.12-L).

Engineering Data

Recorded scientific or technical information (regardless of the form or method of recording) including computer software documentation. Engineering data defines and documents an engineering design or product configuration (sufficient to allow duplication of the original items) and is used to support production, engineering and logistics activities.

Includes, for example:

- all final plans, procedures, reports, and documentation pertaining to systems, subsystems, computer and computer resource programs, component engineering, operational testing, human factors, reliability, availability, and maintainability, and other engineering analysis, etc., and
- Technical data package (reprocurement package) which includes all engineering drawings, associated lists, process descriptions, and other documents defining physical geometry, material composition, and performance procedures.

Excludes:

- computer software or financial, administrative, cost or pricing, or management data or other information incidental to contract administration.

Management Data

The data items necessary for configuration management, cost, schedule, contractual data management, program management, etc., required by the government in accordance with functional categories selected from the DODISS and DoD 5010.12-L.

Includes, for example:

- contractor cost reports, cost performance reports, contract funds status reports, schedules, milestones, networks, integrated support plans, etc.

Support Data

The data items designed to document support planning in accordance with functional categories selected from DoD 5010.12-L.

Includes, for example:

- supply; general maintenance plans and reports; training data; transportation, handling, storage, and packaging information; facilities data; data to support the provisioning process and all other support data; and software supportability planning and software support transition planning documents.

Data Depository

The facility designated to act as custodian to maintain a master engineering specification and establish a drawing depository service for government approved documents that are the property of the U.S. Government. As custodian for the government, the depository, authorized by approved change orders, maintains these master documents at the latest approved revision level. This facility is a distinct entity.

Includes, for example:

- all drafting and clerical effort necessary to maintain documents.

Excludes:

- all similar effort for facility's specification and drawing control system, in support of its engineering and production activities.

Note: When documentation is called for on a given item of data retained in the depository, the charges (if charged as direct) will be to the appropriate data element.

Peculiar Support Equipment

The design, development, and production of those deliverable items and associated software required to support and maintain the system or portions of the system while the system is not directly engaged in the performance of its mission, and which are not common support equipment (See H.3.7 below).

Includes:

- vehicles, equipment, tools, etc., used to fuel, service, transport, hoist, repair, overhaul, assemble, disassemble, test, inspect, or otherwise maintain mission equipment,
- any production of duplicate or modified factory test or tooling equipment delivered to the government for use in maintaining the system. (Factory test and tooling equipment initially used by the contractor in the production process but subsequently delivered to the government will be included as cost of the item produced), and

- any additional equipment or software required to maintain or modify the software portions of the system.

Excludes:

- overall planning, management and task analysis functions inherent in the work breakdown structure element, Systems Engineering/Program Management, and
- common support equipment, presently in the DoD inventory or commercially available, bought by the using command, not by the acquiring command.

Test and Measurement Equipment

The peculiar or unique testing and measurement equipment which allows an operator or maintenance function to evaluate operational conditions of a system or equipment by performing specific diagnostics, screening or quality assurance effort at an organizational, intermediate, or depot level of equipment support.

Includes, for example:

- test measurement and diagnostic equipment, precision measuring equipment, automatic test equipment, manual test equipment, automatic test systems, test program sets, appropriate interconnect devices, automated load modules, taps, and related software, firmware and support hardware (power supply equipment, etc.) used at all levels of maintenance, and
- packages which enable line or shop replaceable units, printed circuit boards, or similar items to be diagnosed using automatic test equipment.

Support and Handling Equipment

The deliverable tools and handling equipment used for support of the mission system.

Includes, for example:

- ground support equipment, vehicular support equipment, powered support equipment, nonpowered support equipment, munitions material handling equipment, materiel handling equipment, and software support equipment (hardware and software).

Common Support Equipment

The items required to support and maintain the system or portions of the system while not directly engaged in the performance of its mission, and which are presently in the DoD inventory for support of other systems.

Includes:

- acquisition of additional quantities of this equipment needed to support the item
    - all efforts required to assure the availability of this equipment to support the item.

Test and Measurement Equipment

The common testing and measurement equipment which allows an operator or maintenance function to evaluate operational conditions of a system or equipment by performing specific diagnostics, screening or quality assurance effort at an organizational, intermediate, or depot level of equipment support.

Includes, for example:

- test measurement and diagnostic equipment, precision measuring equipment, automatic test equipment, manual test equipment, automatic test systems, test program sets, appropriate interconnect devices, automated load modules, taps, and related software, firmware and support hardware (power supply equipment, etc.) used at all levels of maintenance, and
- packages which enable line or shop replaceable units, printed circuit boards, or similar items to be diagnosed using automatic test equipment.

Support and Handling Equipment

The deliverable tools and handling equipment used for support of the mission system.

Includes, for example:

- ground support equipment, vehicular support equipment, powered support equipment, nonpowered support equipment, munitions material handling equipment, materiel handling equipment, and software support equipment (hardware/software).

Operational/Site Activation

The real estate, construction, conversion, utilities, and equipment to provide all facilities required to house, service, and launch prime mission equipment at the organizational and intermediate level.

Includes:

- conversion of site, ship, or vehicle,
- system assembly, checkout, and installation (of mission and support equipment) into site facility or ship to achieve operational status, and
- contractor support in relation to operational/site activation.

System Assembly, Installation, and Checkout on Site

The materials and services involved in the assembly of mission equipment at the site.

Includes, for example:

- installation of mission and support equipment in the operations or support facilities and complete system checkout or shakedown to ensure operational status. (Where appropriate, specify by site, ship or vehicle.)

Contractor Technical Support

The materials and services provided by the contractor related to activation.

Includes, for example:

- repair of repairable, standby services, final turnover, etc.

Site Construction

Real estate, site planning and preparation, construction, and other special-purpose facilities necessary to achieve system operational status.

Includes, for example:

- construction of utilities, roads, and interconnecting cabling.

Site/Ship/Vehicle Conversion

The materials and services required to convert existing sites, ships, or vehicles to accommodate the mission equipment and selected support equipment directly related to the specific system.

Includes, for example:

- operations, support, and other special purpose (e.g., launch) facilities conversion necessary to achieve system operational status. (Where appropriate, specify by site, ship or vehicle.)

Industrial Facilities

The construction, conversion, or expansion of industrial facilities for production, inventory, and contractor depot maintenance required when that service is for the specific system.

Includes:

- equipment acquisition or modernization, where applicable,
- maintenance of these facilities or equipment, and
- industrial facilities for hazardous waste management to satisfy environmental standards.

Construction/Conversion/Expansion

The real estate and preparation of system peculiar industrial facilities for production, inventory, depot maintenance, and other related activities.

Equipment Acquisition or Modernization

The production equipment acquisition, modernization, or transferal of equipment for the particular system. (Pertains to government owned and leased equipment under facilities contract.)

Maintenance (Industrial Facilities)

The maintenance, preservation, and repair of industrial facilities and equipment.

Flight Support Operations and Services

Mate/checkout/launch; mission control; tracking; and command, control and communications (C3); recovery operations and services; and launch site maintenance/refurbishment. This element supports the launch vehicle, orbital transfer vehicle, and/or space vehicle during an operational mission.

Sub-elements to the flight operations and services:

Mate/Checkout/Launch

The preflight operations and services subsequent to production and/or storage, and the actual launch of the complete system and payload.

Includes, for example:

- materials to conduct equipment receiving and checkout at launch site, preflight assembly and checkout, pre/post flight data reduction and analysis, and any prelaunch flight control/mission control planning.

Mission Control

The personnel and materiel required to operate individual mission control centers and to perform ground command and control with the space vehicles.

Includes, for example:

- mission control centers such as Constellation Command Center, Battle Management/Command Control Center (BM/C3), Space Asset Support System Control Center, and Space Transportation Control Center.

Excludes:

- tracking and communications centers (these are included below.)

Tracking and C3

The personnel and materiel required to perform the functions of telemetry, tracking, controlling, and data retrieval for the mission control systems.

Includes, for example:

- mission control systems, on the ground or in space, including Satellite Control Facility; Remote Tracking Station; Tracking, Data, Relay Satellite System; and other ground/space tracking systems.

Excludes:

- initial acquisition of tracking and C^3.

Recovery Operations and Services

The contractor effort and materiel necessary to effect recovery of the space vehicle or other mission equipment.

Includes:

- the launch site recovery forces, reentry site recovery forces, logistics support to recovery forces, logistics support to the recovery operations, communications, and transportation of recovered equipment to assigned facilities.

Launch Site Maintenance/Refurbishment

The organization, maintenance, and management of launch vehicle facilities and mission equipment, and support at the launch base.

Includes, for example:

- requirements to clean up and refurbish each launch site after each launch.

Storage

Those costs of holding portions of the space system while awaiting use of the system being stored, prepared for storage, or recovered from storage. Periods of holding result from schedule changes and/or technological problems exogenous to the portion of the space system.

Includes:

- Sub-elements to storage.

Planning and Preparation

The planning and preparation costs for storage of all systems/subsystems associated with the launch vehicle, orbital transfer vehicle, and space vehicle equipment.

Includes, for example:

- generation of any storage or maintenance instructions and documents necessary for repairable systems or subsystems.

Storage

The cost incurred while the systems or subsystems of the launch vehicle, orbital transfer vehicle, and space vehicle equipment are in storage.

Transfer and Transportation

The transfer and storage costs incurred when the systems/subsystems of the launch vehicle, orbital transfer vehicle, and space vehicle equipment are moved from one location to another.

Includes, for example:

- costs of relocation necessitated by mission requirements.

Initial Spares and Repair Parts

The deliverable spare components, assemblies and subassemblies used for initial replacement purposes in the materiel system equipment end item.

Includes:

- repairable spares and repair parts required as initial stockage to support and maintain newly fielded systems or subsystems during the initial phase of service, including pipeline and war reserve quantities, at all levels of maintenance and support.

Excludes:

- development test spares and spares provided specifically for use during installation, assembly, and checkout on site. Lower level WBS breakouts should be by subsystem.

Appendix C3 – Example Risk Management Plan Outline.

The Risk Management Guide for DoD Acquisitions also provides an example risk management plan for an acquisition program. An outline of a risk management plan follows:

Background/Purpose, e.g.:

- State Policies That Apply
- Integrate Risk Management Into All Other Project Management Activities
- Support Decision Analysis Process
- Support Resolution Of Technical Issues

Scope

- Project Description
- Risk Management Strategy And Acquisition Strategy
- State Assumptions
- Identify Influencing Requirements & Constraints

Organization And Organizational Responsibilities Of Participants

Definitions

Risk Management Approach.

- Describe Program Risk Management Process To Be Employed; I.E., Risk Planning, Assessment, Handling, Monitoring And Documentation, And A Basic Explanation Of These Components.
- Provide Application Guidance For Each Of The Risk Management Functions Listed Above
- Describe How And How Risks Will Be Tracked

Risk Assessment

- Describe Assessment Procedures
- Describe Risk Identification Reporting Procedures
- Define Information To Be Documented And Reported
- Describe Assessment Techniques And Tools To Be Used

Risk Handling.

- Describe Procedures To Determine And Evaluate Various Risk-Handling Options
- Identify Tools To Assist In Implementing The Risk-Handling Process
- Provide Guidance On The Use Of The Various Handling Options For Specific Risks
- Identify Reporting Requirements

Risk Monitoring

- Describe Procedures To Monitor The Status Of Identified Risk Events
- Provide Criteria For Selection Of Risks To Be Reported On

- Identify The Frequency Of Reporting.
- Provide Guidance On The Selection Of Metrics.

Documentation And Reports

- Provide The Status Of The Risk Program
- Identify The Risks
- Describes The MIS Structure, Rules, And Procedures Used To Document The Results Of The Risk Management Process.

Appendix C4–Risk Identification Trigger List

A set of starting questions for identifying potential program risks have been provided in different functional areas. These questions were derived from the Risk Management Critical Process Assessment Tool (CPAT) developed by SMC/AXD as part of the Military Specifications and Standards Reform Program (MSSRP). They are not meant as all-inclusive, but serve as a starting point of discussion by the management team, a team of experts, or an IPT.

Systems Engineering and Technical Risk Questions

Are the program requirements/objectives clearly defined? Have the stakeholders had opportunities to influence the objectives, requirements and design solution?

Have all system functions been identified and used to derive requirements?

Do design(s) or requirement(s) push the current state-of-the art?

Have vague requirements(s) been implemented in a manner such that a change has the potential to cause large ramifications?

Are the problems/requirements/objectives well understood?

Have the designs/concepts/components been proven in one or more existing system?

Is there adequate margin to meet system performance, reliability, and maintainability requirements?

Is the design easily manufacturable/producible/reworkable?

Are there environmental risks associated with the manufacturing or deployment of the system?

Were there governmental, environmental, safety constraints considered?

Are interfaces clearly defined? External? Internal?

Do the interfaces have clearly defined ownership to ensure adequate attention to details?

Are the external interfaces well defined and stable?

Is there adequate traceability from design decisions back to requirements to ensure the effect of changes can be adequately assessed?

Has the concept for operating the system been adequately defined to ensure the identification of all requirements?

Is there a clearly defined requirement verification plan?

Is there a clearly defined configuration management plan and is it being followed?

Are appropriate lessons learned from prior programs integrated into the design?

Cost Risk Questions

Are budgets adequate to handle the scope of program requirements/objectives?

Are the budgets adequate to handle the level of changes expected to occur?

Are there any state-of-the-art products for which the cost is very soft?

Are there any suppliers whose performance is potentially questionable?

Are there any products where a viable manufacturer must be developed?

Are the manufacturing processes unproven or partially unproven?

Can a single supplier hold the program hostage?

Are there any key suppliers whose financial health is in question?

What areas need to have at least two suppliers?

Are there areas of concern where the potential for delays in development, manufacturing, or demonstration of a product could result in a cascading effect (cost increases) on system costs?

Has the cost of complying with applicable security requirements been included in the budget?

Has the cost of regulatory, statutory, or environmental constraints been included?

Have ground rules and assumptions for cost modeling and cost risk assessments been clearly defined and documented?

Schedule Risk Questions

Does a complete, detailed schedule / IMS exist?

Has a schedule risk analysis been performed?

Are the schedules adequate to meet objective(s)?

Will GFE/GFI be available when needed?

Are there critical lead item concerns?

Has adequate schedule been provided to allow adequate schedule slack?

Are there technical or performance risks that lie on the critical path?

Are there resource limitations, e.g. personnel/staffing, facilities, manufacturing tools, simulators, test equipment, which could impact the critical path?

Is the schedule overly optimistic?

Is the schedule sub-optimal due to fiscal funding limitations and is it sensitive to potential funding changes?

Program Management Risk Questions

Are there risks associated with the teaming allocation of responsibilities?

Does geographical separation among team members potentially impact the program?

Does the program manager have previous management experience as a contractor?

Is the technical skill set in short supply?

Are there adequate resources for a management reserve?

Has pre-contract work been performed?

Is the organizational structure in place?

What controls are in place to manage subcontractors?

Software Risk Questions

Was a detailed operations concept used to derive requirements?

How well are the software requirements defined?

Are the algorithms to be programmed developed?

Is the software reuse? Realistically how much?

What is the interface complexity?

What is the stability of the interfaces?

What is the implementation difficulty?

What is the anticipated code size?

Is the hardware/software integration complex? Extensive?

Is the schedule for software development compressed?

How good is the documentation on reuse software?

Do simulators exist or need to be developed to check out the software?

Do simulators or prototype hardware exist for hardware/software integration?

Can the hardware/software handle the data rate input?

Manufacturing/Producibility Risk Questions

Are the design requirements well defined?

Are the design requirements stable?

Does a prototype exist?

Is the first article the flight article?

Does a manufacturing line exist?

Are there subsystems/components that must be produced in greater quantities than past experience?

What is the production failure rate?

Are there process steps prone to breakage?

Are metrics on some production steps such that the manufactured component is close to the tolerance of acceptability?

Are an adequate number of suppliers available for key components?

Are there integration and test issues?

Are there facility availability issues, particularly if a stressing production rate is required?

Is there slack in the schedule for unexpected problems?

Will the test equipment and special tooling be available when needed?

Systems Integration

Are there components for which the integration is complex or difficult?

What is the difficulty of hardware / software integration?

How well are hardware and software interfaces defined?

Are there prototypes, pathfinders and engineering models available for system integration testing?

Is the CONOPS modeled after existing systems?

Are CONOPS interfaces defined?

How well are space vehicle interfaces defined?

Is there a simulation environment ready to support assembly, integration and test? Is it adequate to the anticipated volume?

Does the ground segment exist, or must it be defined concurrently with the space segment development?

Does the ground segment need to be merged into an existing system? How stable is that design?

Is there a transition plan defined in going from an old system to the new?

Are requirements changing?

What is the potential of funding changes?

What is the impact of funding shortfalls and project stretch out?

Is the customer in the development pipeline, e.g. obtaining frequency allocation, concurrent risk reduction efforts, mandated GFE?

What external factors could impact the program?

Appendix C5–Techniques of Functional Analysis

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 (FFBDs, IDEFs, etc), 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. 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.

Functional Flow Block Diagrams (FFBDs)

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. The figure below shows the typical symbology used in block diagrams. A detailed discussion of the symbology/conventions used follows.

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.

[*] NOTE: FFBDs may also be referred to as functional flow diagrams. Some may even refer to them as Functional Block Diagrams, but that term has alternate interpretations. One common meaning of functional block diagrams refers to diagrams describing the relationships among functional areas (or physical elements) of a system. The relationships/interactions among the prime items of a segment might be shown in this form of a Functional Block diagram. This particular application of the term *Functional Block Diagrams* is also known as *Schematic Block Diagrams* (SBDs)

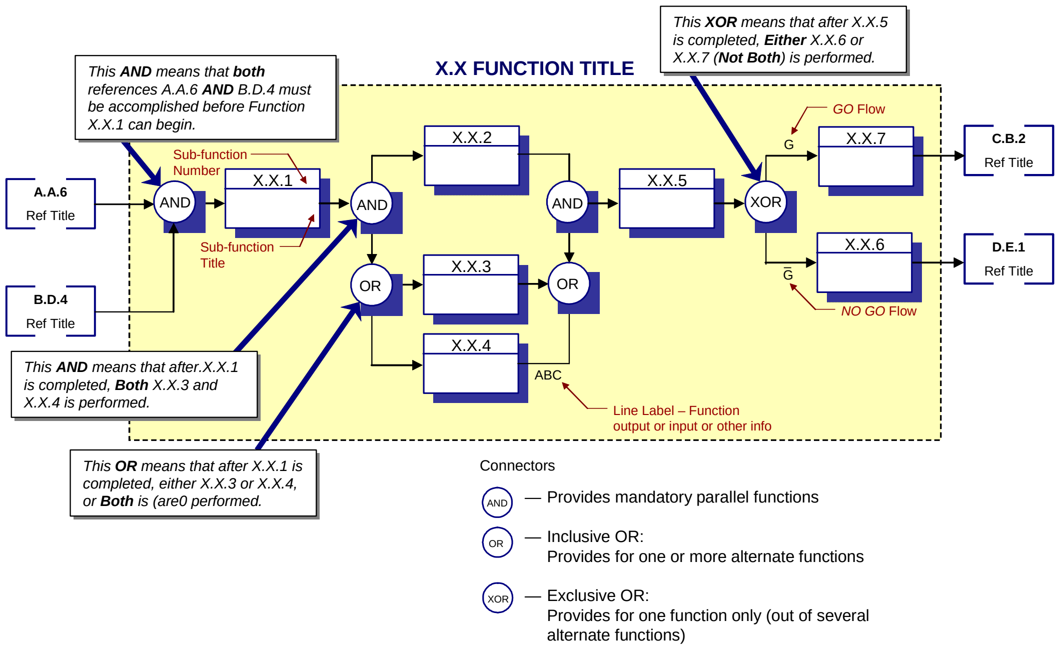


Figure 44. Sample functional flow block diagram (FFBD)–typical symbols used in FFBDs

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.

1. As a Reference Block:

   a.) 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.

   b.) 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,

   c.) 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.

2. 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.

   a.) 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.

   b.) 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.

   c.) 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; 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 C.5-1.

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.

Figure 45 also illustrates the decomposition of functions, producing functional flow block diagrams at succeeding lower levels of the functional architecture. This process provides the systems engineer with a hierarchy of functions that provides the framework for deriving performance requirements that will completely define the system and all its components. At any lower level, the sub-function numbering system carries a reference to the next higher level so that the functional hierarchy is easily discernible.

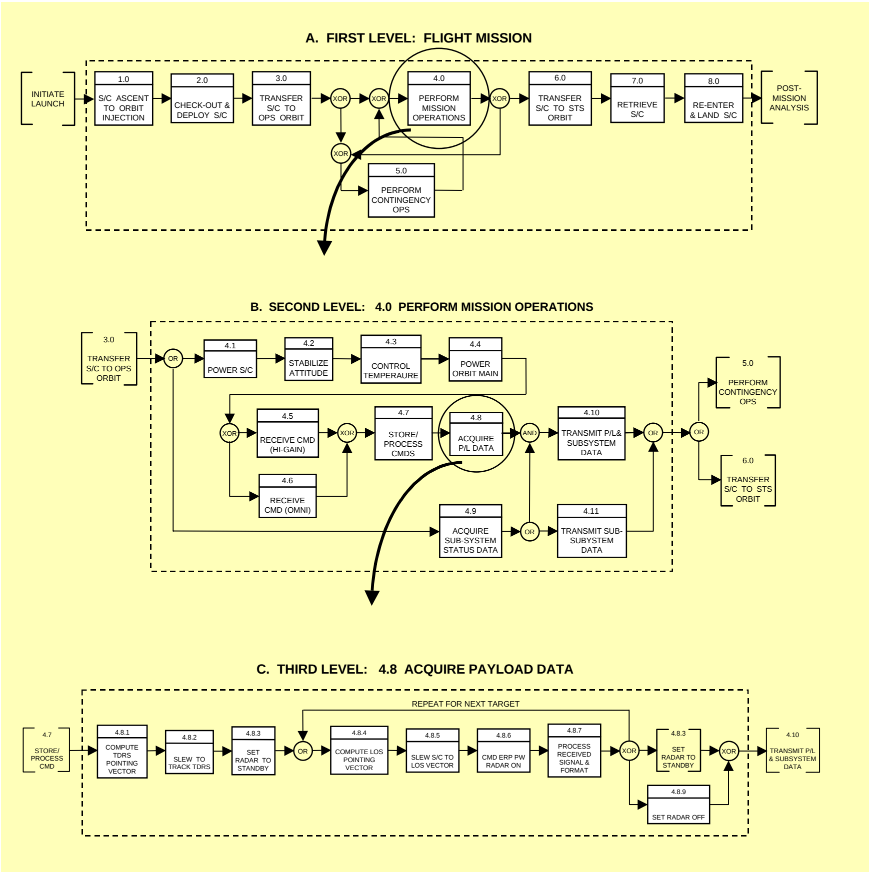


Figure 45. Sample functional flow diagram–showing interrelation of various levels

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 46 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.

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.

Function Analysis Limits – Unfortunately, function analysis by itself does not adequately describe a product. Function analysis does not describe limitations, iteration, information flow, performance, or environments. However, it is a significant and essential tool is systems engineering activities. One method of relating these attributes to functions is the Quality Function Deployment (QFD) tool. See Chapter 4.

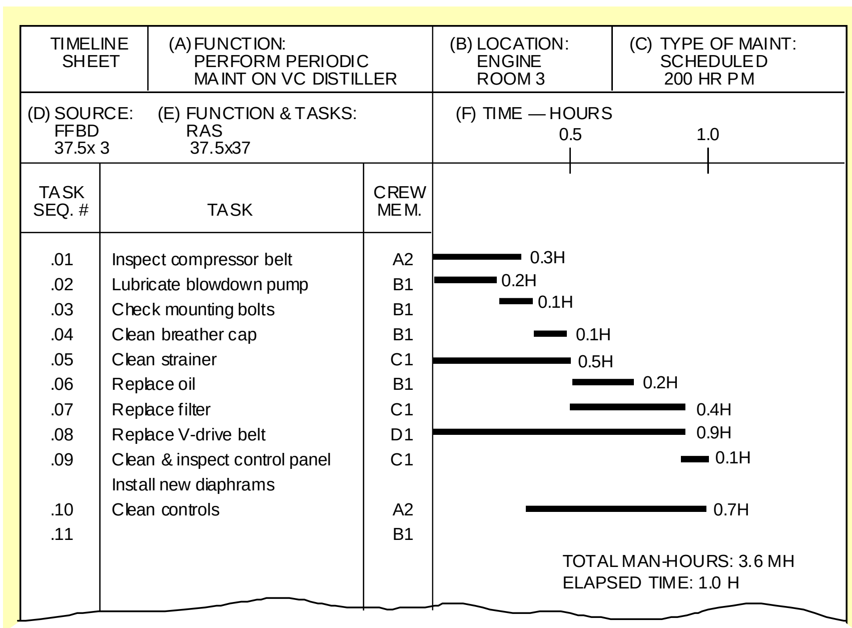


Figure 46. Timeline sheets–show sequence of operational and concurrent action

Appendix C6–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.

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

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

    - o vendor supplied data
    - o comparison with equipment in standard reliability handbooks
    - o engineering estimates

- Establish preliminary estimate of MTTR for each unit considering
    - o minimum sparing to support availability (formal provisioning analysis deferred)
    - o maximum use of commercial maintenance contracts with vendors assumes no logistics or administrative delays for this example.

The figure here 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.

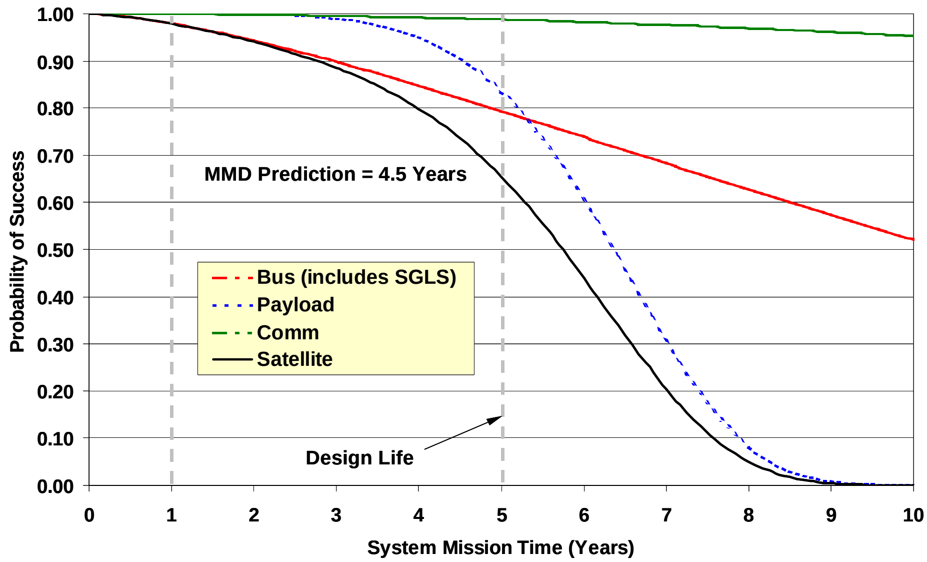


Figure 47. Reliability functions calculated for each major element of satellite

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 above figure)

- 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

The figure below 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.

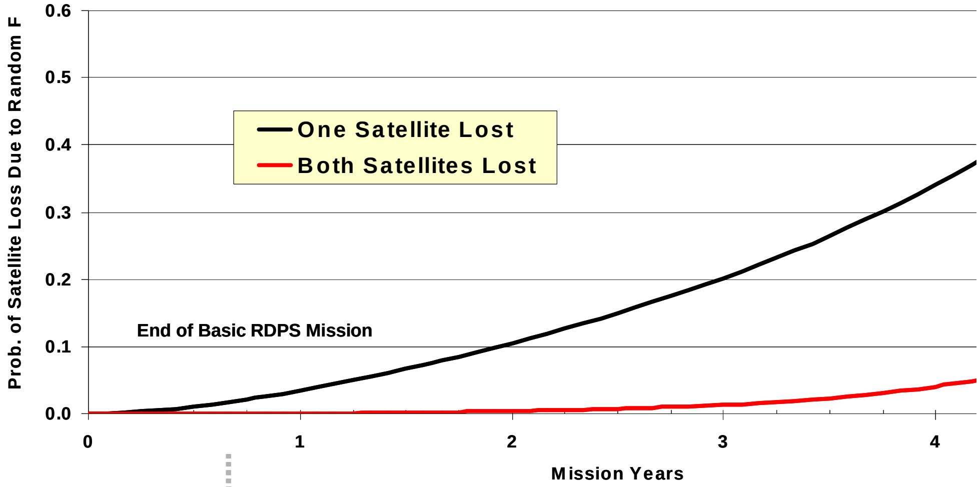


Figure 48. Probability of loss of one or both satellites due to random failure

Table 13 below is an example of a ground segment allocation. This figure 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 14 below is the combined results of space and ground segment dependability. Either ground station can complete the mission without loss of 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. This figure 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 13. 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 14. 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 C7–TPM Examples Using the Hierarchy Methodology

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 systems engineering 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.

TPM Hierarchy Example

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 Manhours 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

Appendix C8–Example Trade Study Outline

Purpose of Study

- Resolve an Issue
- Perform Decision Analysis
- Perform Analysis of Alternatives (Comparative analysis)

Scope of Study

- State level of detail of study
- State Assumptions
- Identify Influencing requirements & constraints

Trade Study Description

- Describe Trade Studies To Be Performed
- The Studies Planned To Make Tradeoffs Among Concepts, User Requirements, System Architectures, Design, Program Schedule, Functional, Performance Requirements, And Life-cycle Costs
- Describe Trade Methodology To Be Selected
- Describe Technical Objectives
- Identify Requirements And Constraints
- Summarize Level of Detail of Analysis

Analytical Approach

- Identify Candidate solutions to be studied/compared
- Measure performance
  - Develop models and measurements of merit
  - Develop values for viable candidates
- Selection Criteria -- risk, performance, and cost are usually at lease three of the factors to be studied
  - Operational Factors (Usability, Ops Skill Levels,…)
  - Reliability
  - Safety
  - Weight
  - Volume
  - Producibility
  - Survivability
  - Other
- Scoring
  - Measures of results to be compared to criteria
  - Weighted reflecting their relative importance in the selection process

- Sensitivity Analysis

Trades Results

- Select User/Operational Concept
- Select System Architecture
- Derive Requirements
    - Performing trade studies to determine alternative functional approaches to meet requirements
    - Alternate Functional Views
    - Requirements Allocations
- Derive Technical/Design Solutions
- Cost Analysis Results
- Risk Analysis Results
- Understand Trade Space

Appendix C9–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 Flight 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 C10–States & Modes

States and Modes 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/expected when. As with other systems engineering terms used in this handbook, definitions and examples for the terms state and mode are provided below and are borrowed from James Martin’s Systems Engineering Guidebook.

State: The condition of a system or subsystem when specific modes or capabilities (or functions) are valid.

Examples of states: Off, Start-up, Ready On, Deployed, Stored, In-Flight, etc.

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. Examples of modes within the Ready state: Normal, Emergency, Surge, Degraded, Reset, etc.

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.

Using States/Modes: The only reason for introducing states and modes into the requirements process and in the resulting specification is as a means to identify different sets of performance requirements for different sets of conditions that will 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.

Common/Shared Requirements: 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.

Mode Transitions: 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.

Appendix C11–C4ISR Architecture Framework

The principal objective of the C4ISR architecture framework is to define a coordinated approach for DoD architecture development, integration, and presentation. The framework is intended to ensure that architecture descriptions can be compared and relate across organizational boundaries. In February, 1998, the DoD Architectural Coordination Council mandated the use of this framework for all C4ISR architecture descriptions. It behooves the system engineer to understand this methodology.

The C4ISR Framework prescribes three views, the operational, systems, and technical views which form steps toward implementing standards in the Joint Technical Architecture to achieve interoperability between systems and forces:[44]

- The Operational Architecture (OA) view is a description of the tasks and activities, operational elements, and information flows required to accomplish or support the warfighter during a military operation. The OA view defines the types of information exchanged, the frequency of exchange, which tasks and activities are supported by the information exchanges, and the nature of information exchanges in sufficient detail to identify specific interoperability requirements. The OA view may be synonymous with the Capabilities/Requirements view discussed above or it may be a subset of the latter that is focused on the needed capabilities for information exchange.
- The Systems Architecture (SA) view is a description of interconnections between systems and intra-connections within a system. For a family or system of systems, the SA view shows how multiple systems link and interoperate. For an individual system, the SA view includes the physical connection, location, and identification of key nodes and specifies system and component performance parameters (e.g., mean time between failure, maintainability, and availability). The SA view also associates physical resources and their performance attributes with the OA view and its requirements via standards defined in the technical architecture (TA).
- The Technical Architecture (TA) view identifies the standards in the Joint Technical Architecture (JTA) that govern system services, interfaces, and relationships for particular systems architecture views and provide the interoperability and other capability needs in particular operational views.

Figure 49 depicts the linkages between views. The framework describes a generic process for describing architectures. The six steps in this generic process are:

- Determine the intended use of the architecture description,
- Determine the scope of the architecture,
- Determine the characteristic to be captured,
- Determine the views and products to be built,
- Build the requisite products, and
- Use the architecture for the intended purpose.

---

44. For more detail, see the *Joint Technical Architecture User Guide and Component JTA Management Plan*, Section 5.1, at http://www.disa.mil/main/jta.html.

An additional reference that provides further insight into this process includes DoD Architecture Framework, Version 1.0: Volume 1, Definitions and Guidelines, Volume 2, Product Descriptions, Volume 3, Appendices.

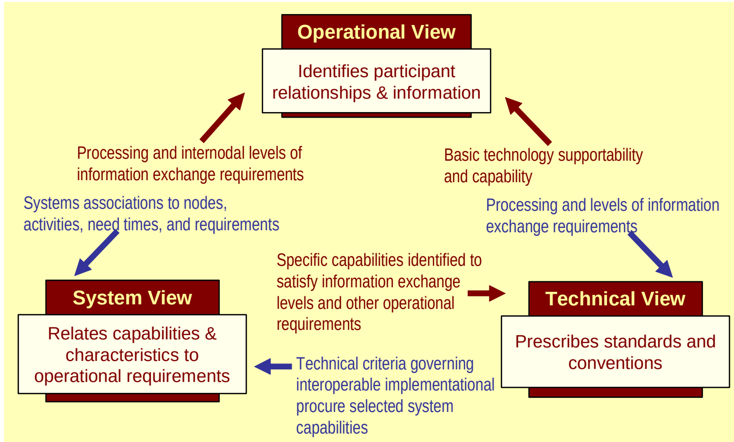


Figure 49. Linkages among the three views

Appendix C12–Example of Threat Evaluation & CAIV Study

The following is an example of how system engineering can perform a top-level assessment of threats to a system. Through the use of models associated with countermeasure implementation, a CAIV analysis can be performed to optimize threat protection and cost to the system.

Top-level generic threats of a space system, Figure 50, are defined in a STAR. A threat risk (TR) model has been developed as a direct function of consequences to the system (Cs) and an inverse function of countermeasure effectiveness (Ecm) and difficulty of an aggressor to impose the threat (Dagg). Likelihood (L) of the threat occurring is used as a weighted average factor when combining threats risks. A life cycle cost model is developed for each threat based on increasing effectiveness of the countermeasure. Table 15 defines scale factors used in the threat risk calculation.

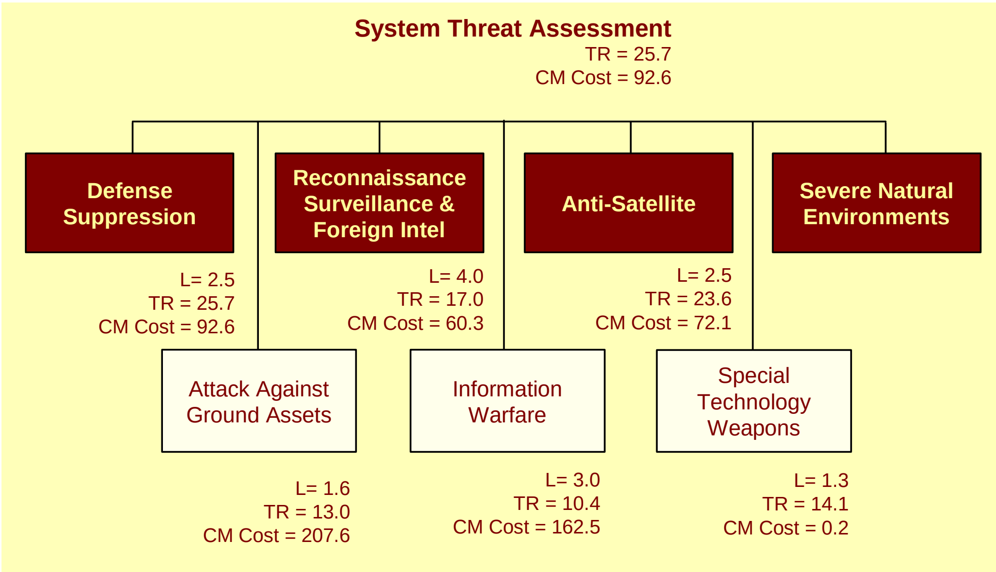


Figure 50. Threat evaluation model

Figure 51 breaks down the generic threats (for example anti-satellite weapons) into specific threats (directed energy, nuclear burst, interceptor) as defined in the STAR. Threat risks and countermeasure costs are determined at the lowest level (i.e., airborne) and rolled up (laser). Likelihood of occurrence (L) provides a weighting factor to combine threats at the same level. The threat risk analysis begins at the lowest level of specific threats (i.e., space, ground, and airborne lasers) and is rolled up to the top level shown in D-1. A value for each threat risk parameter is determined from the definitions in Figure D-2. Effectiveness of countermeasures is defined at the lowest level of threat as are cost models for countermeasures. Results are rolled up to the system level.

Once a basic model is evaluated without countermeasures (Ecm = 1 for all threats) to give a baseline, a CAIV analysis can be performed. Through either linear programming techniques or manual manipulation in a spread sheet, the effectiveness of countermeasures on threat risk can be played off the cost of the countermeasures until a desired result is reached such as minimum threat risk or lowest threat risk for a fixed cost.

Table 15. Scale factors must be defines in a quantitative manner. Definition can be changed depending on the nature of the space system analyzed

Scale Factor	Threat Imposed System Consequence (C_s)	Threat Difficulty for Aggressor (D_{agg}) Without Countermeasures	Effectiveness of Countermeasures (E_{cm})	Likelihood of Occurrence (L) Based on STAR
1	System continues to perform mission uninterrupted; Limited, minor damage (<$1M) which does not impact dependability	Threat capability currently exists in more than one potential enemy nation; Mature technology: Robust aggressor forces with worldwide operational capability; Natural defenses do not exist	No known countermeasure exists or no countermeasures have been implemented in the system against the specific threat; Expect total mission(s) failure	Very Low
2	Causes damage to facilities ($1M - $10M); System still performs its mission without impact to dependability	Threat capability exists in one potential enemy nation; Limited implementation of threat technology; Natural defenses would provide limited protection to minimize damage or compromise	Implemented countermeasures protect the system to the level which allows the primary mission to be completed while sacrificing secondary missions	Low
3	Some key elements of the system are out of commission for more than one month (or >$10M damage); Mission continues with impact to dependability	Threat technology being aggressively pursued by one or more potential enemy nation; current intelligence predicts implementation before 2010; Natural defenses would protect some key elements of the system from major damage or compromise; Moderate aggressor	Implemented countermeasures protect the system to the level which allows the primary and secondary mission(s) to be completed with degraded performance to both	Medium
4	System is partially compromised; Damaged or lost space asset; Some enemy actions can be missed	Threat technology being pursued by one potential enemy nation; current intelligence predicts implementation after 2010; Natural defenses would protect all key elements of the system from major damage or compromise; limited aggressor force with local capab	Implemented countermeasures protect the system to the level which allows the primary mission to be completed with degraded performance to secondary mission(s) only	Medium High
5	System completely compromised; Mission halted; Most or all enemy actions can be completely missed	Threat technology does not exist; limited or no R&D being performed by potential enemy nations; Natural defenses would easily protect the system from any damage or compromise; No identified aggressor force	Implemented countermeasures are 100% effective against the enemy threat; All missions continue uninterrupted; No performance degradation	High

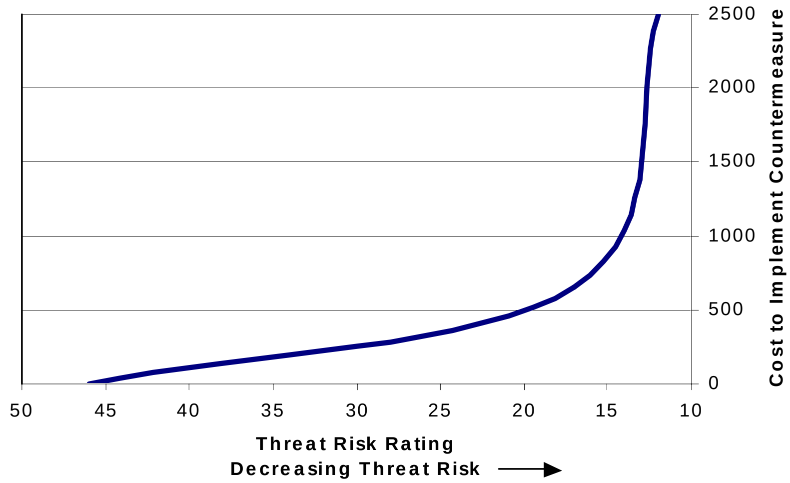


Figure 51. Determine the best manner of investment in countermeasure

Figure 52 depicts an example of a CAIV study done for SBIRS Low. The Study performed for SBIRS Low was to determine the best manner of investment in countermeasure based on information in the NMD STAR and specifics of the system design. A system engineer should allocate requirements such that the cost of countermeasures stays well to the right of the cost cliff. Very little is to be gained by trying to reduce the threat risk below 20 or so in this example.

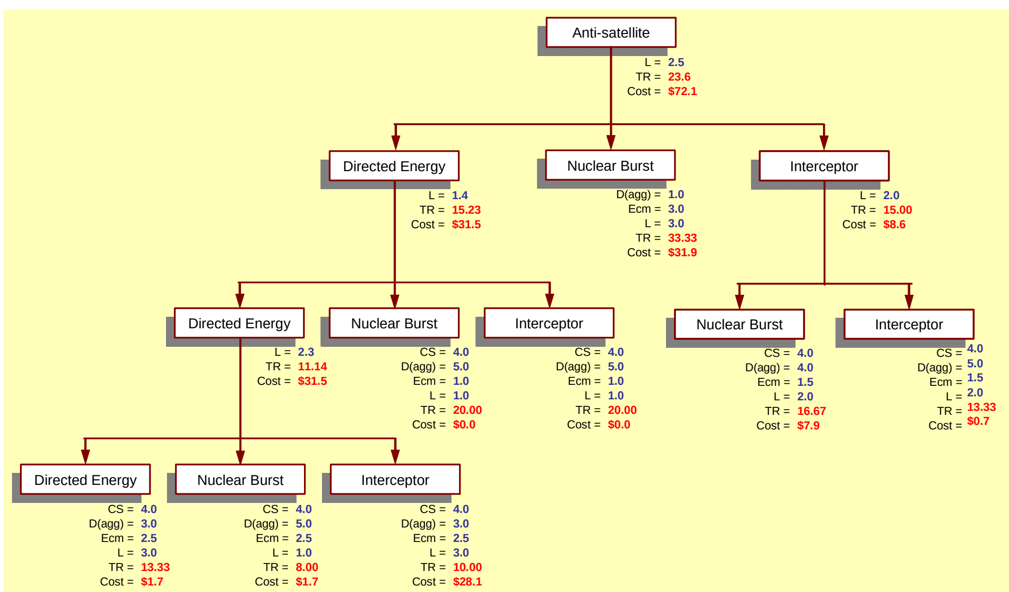


Figure 52. Example of detailed threats from anti-satellite weapons systems as defined in a STAR

Appendix C13–Summary Checklist for an OSS&E Plan

General

The OSS&E process establishes and preserves baselines for operational safety, operational suitability, and operational effectiveness throughout the operational life of a system or end item (including experimental systems). IAW with SMCI, 63-1201, all USAF-developed spacecraft, launch vehicles, and critical ground systems must plan and implement OSS&E and eventually be space flight worthiness certified. The OSS&E plan must specify all critical elements to which the SFW criteria will be applied. Below is a brief checklist for developing and updating an OSS&E plan. Also refer to a much more comprehensive OSS&E guidance[45] However, as OSS&E implementation progresses at SMC, updates to policy, instructions, and guidance will likely occur. Contact SMC/AXE for the latest information.

Stakeholders

The OSS&E plan identifies all stakeholders that have a role in the successful implementation of the SFW certification program and eventual space flight worthiness certification. Each program must perform an assessment to identify stakeholders supporting the SFW certification program and their respective roles. The SFW certification program should include levels or areas of participation from the development and product contractors and well as the appropriate government agencies and organizations (AFSPC, SMC Program Offices, Logistics, Test Centers and System Users) that have a stake in assuring mission success. The stakeholders then document procedures to ensure that all SFW certification program-input requirements are identified, documented, and reviewed; and that all flaws, ambiguities, contradictions, and deficiencies are resolved. (See Section 3.3 of the SMCI 63-1202) SMCI 63-1201 delineates Detachment 8, Detachment 9, and CTF commanders' OSS&E related responsibilities. Additional SFW certification responsibilities need to be addressed in the OSS&E plan. Additional responsibilities may include collection and assessment of selected SFW certification metrics and periodic assessments of the effectiveness of those SFW certification processes taking place at the launch sites.

Each Program Manager will include in the OSS&E Plan how they will verify that each of the SFW criteria has been met. In addition, the Program Manager must detail in their OSS&E Plan, which SFW criteria will be met in real time and which will be verified via past history. Range and Flight Safety criteria must be met and revalidated for each flight.

The Program Manager must certify that spacecraft, launch vehicle, and critical ground systems meet the Space Flight Worthiness criteria and document the method of compliance with these criteria. The Program Manager must also certify his or her system at the time of the Flight Readiness Review.

OSS&E Key Elements

The OSS&E Plan should identify each OSS&E process and define or reference direction, regulations, and instructions, for each. The intent of this section is to describe or reference the engineering development processes and the effective operational, training, supply and

[45] Guidelines for Development of SMC Operational System Safety Suitability and Effectiveness (OSS&E)

Assurance Plans, 12 Jun 03 is available through SMC/AXE maintenance procedures to preserve the system or end item OSS&E characteristics throughout the operational life.

Disciplined Engineering Process

System Design and Qualification

Plan defines a disciplined approach to requirements development, system design and development, qualification and manufacturing to achieve mission objectives.

Operational Risk Management

Plan defines approach to implementation of an operational risk management program to define acceptable risk levels and to manage the impacts of risk items to program activities.

System Safety

Plan identifies and provides approach to eliminate or reduce safety hazards to acceptable levels of risk.

Configuration Management

OSS&E is associated with a specific system or end item configuration. The Plan addresses the process to establish and preserve OSS&E baselines throughout the system's lifecycle.

Test and Evaluation

The plan discusses or references an integrated Test and Evaluation approach or plan in accordance with applicable test and evaluation regulations and directives. The plan or approach should be coordinated with the designated Operational Test and Evaluation activity, as appropriate.

TOs and Technical Data

The Plan addresses any necessary current, validated, and verified TOs and technical data to the Operational Commands and other users. TOs and technical data must clearly identify procedures and requirements necessary to preserve OSS&E baselines. They must identify operational limitations of the system or end item.

Total Ownership Cost

The Plan discusses or references approach to assess and document Total Ownership Cost impacts of any proposed changes to operational use, configuration, maintenance procedures, or part substitutions.

Certifications

Plan addresses approach to certify that spacecraft, launch vehicle, and critical ground systems meet the Space Flight Worthiness criteria established by the SMC/CC. The Plan also addresses any other required certifications supporting OSS&E (e.g., Nuclear Surety and launch site Detachment Flight Certification) prior to system or end item operational use.

Inspections and Maintenance

The Plan identifies development and upkeep of inspections and maintenance procedures to establish OSS&E and prevent its degradation.

Sources of Maintenance and Repair

The Plan addresses approach to ensure that maintenance and repair sources deliver consistently high quality products and services to preserve OSS&E across the full spectrum of operational environments.

Source of Supply

The Plan addresses approach to ensure that sources of supply are capable of producing parts and supplies that preserve the OSS&E baseline across the full spectrum of operational environments.

Training

The Plan addresses approach to develop the necessary system training to enable users to preserve the OSS&E baseline. SMC/AX will develop, maintain, and offer OSS&E training materials for individuals and organizations acquiring space and missile product line systems and end items.

Operations and Maintenance

The Plan addresses approach to develop operations, maintenance, and upgrade processes in conjunction with the user which preserve the OSS&E baseline in accordance with approved Technical Orders and Operations Manuals, and allow upgrade insertion, failure work-arounds, and software and hardware modifications throughout the life of the system.

Technology Demonstration

The Plan addresses approach to ensure that all Advanced Technology Demonstration, Advanced Concept Technology Demonstration, experimental leave-behind systems and end items, and any space flight systems and end items provide for OSS&E.

Schedule

The Plan addresses approach to develop the program schedule to permit implementation of the OSS&E assurance process described in this SMC Instruction. The Plan addresses approach to document all OSS&E readiness reviews.

Budget

The OSS&E Plan contains a planned budget and funding profile necessary to ensure OSS&E throughout the operational life of the system or end item, including support of required reviews, training, hardware/software upgrades, and contractual issues.

Reviews and Certificates

OSS&E Assurance Assessments (OAA)

The OSS&E Plan should describe scope, schedules and budgets of independent reviews. All independent reviews in the OSS&E Plan are coordinated with the SMC Chief Engineer (SMC/EN).

Continuing OSS&E Assessment

The OSS&E Plan for each system and end item must include a detailed description of its COA process.

OSS&E Verification

The Plan should address approach to verification prior to fielding a new system. The plan will address how the system will be verified that it can be operated in an operationally safe, suitable, and effective manner and that the OSS&E baseline will be maintained throughout its operational life.

OSS&E Certification

The plan contains the baselined SFW Criteria and the process for CC approval of the criteria. The Plan also contains the process for CC and Program Office Director formal review for progress and the process to certify the space flight worthiness of the system against the SFW criteria. The OSS&E Plan will state the process to be used to certify the SFW criteria. The OSS&E Plan will also provide rationale for tailoring out criteria that are not applicable.

Reviews

The plan details incremental phased reviews, such as SRRs, SDRs, PDRs, and CDRs for major upgrades. These reviews must include assessments of overall system safety, suitability, and effectiveness, and the steps taken to mitigate risks to mission success. Also address OSS&E as it pertains to readiness reviews

The plan should identify how OSS&E activities are formally reflected in the contract, as appropriate, and how OSS&E activities are incorporated/referenced into the program management plan and/or other appropriate program documentation.

Maintenance of the OSS&E Baseline

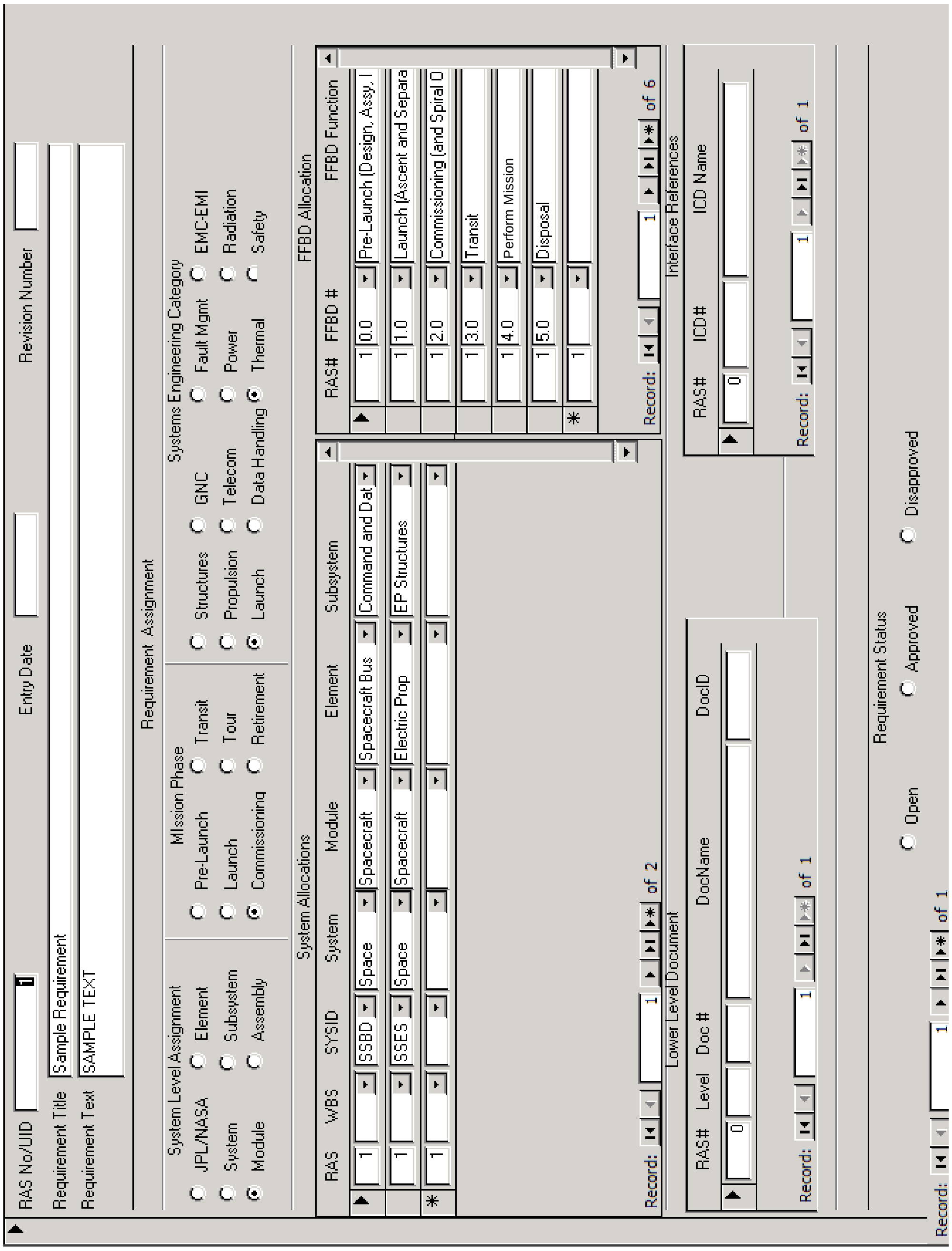
The plan identifies how the user/operator community will be engaged to accomplish the COA in accordance with defined AFMC metrics levels (see Appendix B of SMCI 63-1201). This includes documented agreements between the SM and the using command or users which establish key or critical OSS&E characteristics of the system or end item. The plan details actions to be taken to assure and preserve the OSS&E baseline throughout the operational life of the system or end item, including processes to continuously compare the actual OSS&E characteristics with the baseline and to resolve any discrepancies. The plan describes the mechanisms by which operational lessons learned can be captured and disseminated across programs and contractors after a system or end item is fielded.

The OSS&E Plan should evolve over time to reflect much of the following depending on where the program is at with respect to the acquisition lifecycle:

- Describe approach to establish/updating baseline during system updates.
- The OSS&E baseline should include the configuration baseline (specifications, drawings, and software code listings), modifications and approved engineering change proposals, operational requirements documents, Acquisition Program Baselines, technical orders (TOs), certifications, training, maintenance facilities, spare parts, test plans, concepts of operation, and threat scenarios.
- Describe safety and health compliance requirements and approach to ensuring acceptable risks
- Identify hazards, assessment of risk, determination mitigating measures, and acceptance of residual risk.

- Describe the suitability requirements of the system/subsystem to be placed satisfactorily in an operational environment and approach to ensuring suitability requirements are met.
- Availability, manufacturability, compatibility, transportability, interoperability, reliability, redundancy, wartime use rates, maintainability, safety, human factors, architectural and infrastructure compliance, manpower supportability, logistics supportability, natural environmental effects and impacts, and documentation and training requirements.
- Describe the Operational effectiveness (degree of mission accomplishment) of the system.

Appendix C14–Example Requirements Allocation Sheet



Appendix C15–Requirements Evaluation and Acceptability Criteria

Correct–

No errors exist that effect design. Parameter units are correct and consistent with related specs, ICDs, source requirements. Related functionality is accurately captured in requirement statement. Allocations are correct, consistent, & clear.

Concise–

There is only one plausible semantic interpretation. Statement is written in unambiguous contract language. Terms are adequately defined.

Traceable–

There is an audit trail from the origin of the requirement and how it evolved. The sources are valid and current. Traceability to supporting rationale for requirements derivations, supporting analyses/trades, Issues, actions, and decisions to conclude each requirement. Traceability used for verification requirements and verification planning.

Verifiable–

It is possible to verify that the system meets the requirement. Requirement is verifiable by Test, Demo, Analysis, and/or Inspection.

Modifiable–

Necessary change to be made completely and consistently. Reserved for requirements predicted to change. Related to evolving capabilities achieved through spiral development

Validated–

Applicable to an expressed user need and consistent with program concept of operations.

Completeness–

Defines a system that satisfies all user requirements. Ensure all user requirements trace to system and lower level requirements. Plan for and track validation of user requirements during prototyping, qualifying and IOT&E.

Consistency–

One system requirement does not conflict with another. We identify, verify, and track functionality of the external interfaces, during systems definition.

Efficiency–

No overlap or redundancy - minimal set of requirements.

Appendix D–References and Bibliography

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.

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, Fifth Edition, Defense Acquisition University, June 2002

Department of Defense, Defense Air University, Defense Acquisition Deskbook

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 Cy-bernetics, 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

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.

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.

Pugh Controlled Convergence

Pugh, Stuart. Total Design: Integrated Methods for Successful Produce Engineering. Wokingham, England: Addi-son-Wesley, 1990.

NXN Chart

Lano, R.J. A Technique for Software and Systems Design. Amsterdam, The Netherlands: North-Holland Publishing Co., 1979.

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.

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.

Quality Function Deployment

Akao, Yoji, ed. Quality Function Deployment: Integrating Customer Requirements into Product Design. Cambridge, MA: Productivity Press, 1990.

Department of Defense Instructions and Guidance

Air Force policy Directive, AFPD 99-1, Test and Evaluation Process

Chairman of the Joint Chiefs of Staff (CJCS) Instruction 3170.01C and CJCS Manual 3170.01, both dated 24 June 2003

Department of Defense, DoD Directive 5000.1, May 12, 2003, Operation of the Defense Acquisition System

Department of Defense, DoD Instruction 5000.2, Operation of the Defense Acquisition System, May 12, 2003

Department of Defense, DoDD 5141.2, Director, Operational Test and Evaluation

Merriam Webster’s Collegiate Dictionary, Tenth Edition

Microsoft® Encarta® Reference Library 2004, Microsoft Corporation

Military Handbook, MIL-HDBK-881, DoD Handbook – Work Breakdown Structure, 2 January 1998

National Security Space Acquisition Policy, Number 03-01, Version 3.0, July 28, 2003

Space and Missile Systems Center, Policy Letter on Specification and Standards Usage at SMC, SMC/CC, Jan. 14, 2003

Appendix E–Customer Review & Feedback Form

This form is provided for your convenience.

	DATE: **10 November 2003**
TITLE OF DOCUMENT: **SMC Systems Engineering Primer & Handbook** **Concepts, Processes, and Techniques**	DOCUMENT: ***2nd 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 WORD version of this form and forward completed form to:

Mr. Dave Davis david.davis@losangeles.af.mil

or

Barry Portner bportner@tor.bdsys.com .

Index

Prepared by

bd SYSTEMS

SPACE & MISSILE SYSTEMS CENTER

THE AEROSPACE CORPORATION