

Systems Engineering Fundamentals Part 1

Five (5) Continuing Education Hours Course #ME1470

Approved Continuing Education for Licensed Professional Engineers

EZ-pdh.com Ezekiel Enterprises, LLC 301 Mission Dr. Unit 571 New Smyrna Beach, FL 32170 800-433-1487 support@ez-pdh.com



Course Description:

The Systems Engineering Fundamentals Part 1 course satisfies five (5) hours of professional development.

Part one of a two-part course, this course is designed as a distance learning course that provides a basic, conceptual-level description of engineering management disciplines that relate to the development and life cycle management of a system.

Objectives:

The primary objective of this course is to enable the student to understand systems engineering and to learn the tools and processes necessary to develop a system efficiently and support the life cycle of that system.

Grading:

Students must achieve a minimum score of 70% on the online quiz to pass this course. The quiz may be taken as many times as necessary to successful pass and complete the course.

A copy of the quiz questions are attached to last pages of this document.

Table of Contents

Systems Engineering Fundamentals Part 1

1: Introduction to Systems Engineering Management 1
2: Systems Engineering Management in DOD
Acquisition9
3: Systems Engineering Process Overview
4: Requirements Analysis 24
5: Functional Analysis and Allocation
6: Design Synthesis
7: Verification
8: Systems Engineering Process Outputs 45
9: Work Breakdown Structure 54
10: Configuration Management 60
Quiz Questions 68

CHAPTER 1

INTRODUCTION TO SYSTEMS ENGINEERING MANAGEMENT

1.1 PURPOSE

This chapter establishes some of the basic premises that are expanded throughout this course. Basic terms explained in this chapter are the foundation for following definitions. Key systems engineering ideas and viewpoints are presented, starting with a definition of a system.

1.2 DEFINITIONS

A System Is ...

Simply stated, a system is an integrated composite of people, products, and processes that provide a capability to satisfy a stated need or objective.

Systems Engineering Is...

Systems engineering consists of two significant disciplines: the technical knowledge domain in which the systems engineer operates, and systems engineering management. This course focuses on the process of systems engineering management.

Three commonly used definitions of systems engineering are provided by the best known technical standards that apply to this subject. They all have a common theme:

• A logical sequence of activities and decisions that transforms an operational need into a description of system performance parameters and a preferred system configuration. (MIL-STD- 499A, *Engineering Management*, 1 May 1974. Now cancelled.)

- An interdisciplinary approach that encompasses the entire technical effort, and evolves into and verifies an integrated and life cycle balanced set of system people, products, and process solutions that satisfy customer needs. (EIA Standard IS-632, *Systems Engineering*, December 1994.)
- An interdisciplinary, collaborative approach that derives, evolves, and verifies a life-cycle balanced system solution which satisfies customer expectations and meets public acceptability. (IEEE P1220, *Standard for Application and Management of the Systems Engineering Process*, [Final Draft], 26 September 1994.)

In summary, systems engineering is an interdisciplinary engineering management process that evolves and verifies an integrated, life-cycle balanced set of system solutions that satisfy customer needs.

Systems Engineering Management Is...

As illustrated by Figure 1-1, systems engineering management is accomplished by integrating three major activities:

- Development phasing that controls the design process and provides baselines that coordinate design efforts,
- A systems engineering process that provides a structure for solving design problems and



Figure 1-1. Three Activities of Systems Engineering Management

tracking requirements flow through the design effort, and

• Life cycle integration that involves customers in the design process and ensures that the system developed is viable throughout its life.

Each one of these activities is necessary to achieve proper management of a development effort. Phasing has two major purposes: it controls the design effort and is the major connection between the technical management effort and the overall acquisition effort. It controls the design effort by developing design baselines that govern each level of development. It interfaces with acquisition management by providing key events in the development process, where design viability can be assessed. The viability of the baselines developed is a major input for acquisition management Milestone (MS) decisions. As a result, the timing and coordination between technical development phasing and the acquisition schedule is critical to maintain a healthy acquisition program.

The systems engineering process is the heart of systems engineering management. Its purpose is to provide a structured but flexible process that transforms requirements into specifications, architectures, and configuration baselines. The discipline of this process provides the control and traceability to develop solutions that meet customer needs. The systems engineering process may be repeated one or more times during any phase of the development process.

Life cycle integration is necessary to ensure that the design solution is viable throughout the life of the system. It includes the planning associated with product and process development, as well as the integration of multiple functional concerns into the design and engineering process. In this manner, product cycle-times can be reduced, and the need for redesign and rework substantially reduced.

1.3 DEVELOPMENT PHASING

Development usually progresses through distinct levels or stages:

- Concept level, which produces a system concept description (usually described in a concept study);
- System level, which produces a system description in performance requirement terms; and
- Subsystem/Component level, which produces first a set of subsystem and component product performance descriptions, then a set of corresponding detailed descriptions of the products' characteristics, essential for their production.

The systems engineering process is applied to each level of system development, one level at a time, to produce these descriptions commonly called configuration baselines. This results in a series of configuration baselines, one at each development level. These baselines become more detailed with each level.

In the Department of Defense (DoD) the configuration baselines are called the functional baseline for the system-level description, the allocated baseline for the subsystem/ component performance descriptions, and the product baseline for the subsystem/component detail descriptions. Figure 1-2 shows the basic relationships between the baselines. The triangles represent baseline control decision points, and are usually referred to as technical reviews or audits.

Levels of Development Considerations

Significant development at any given level in the system hierarchy should not occur until the configuration baselines at the higher levels are considered complete, stable, and controlled. Reviews and audits are used to ensure that the baselines are ready for the next level of development. As will be shown in the next chapter, this review and audit process also provides the necessary assessment of system maturity, which supports the DoD Milestone decision process.

1.4 THE SYSTEMS ENGINEERING PROCESS

The systems engineering process is a top-down comprehensive, iterative and recursive problem



Figure 1-2. Development Phasing

solving process, applied sequentially through all stages of development, that is used to:

- Transform needs and requirements into a set of system product and process descriptions (add-ing value and more detail with each level of development),
- Generate information for decision makers, and
- Provide input for the next level of development.

As illustrated by Figure 1-3, the fundamental systems engineering activities are Requirements Analysis, Functional Analysis and Allocation, and Design Synthesis—all balanced by techniques and tools collectively called System Analysis and Control. Systems engineering controls are used to track decisions and requirements, maintain technical baselines, manage interfaces, manage risks, track cost and schedule, track technical performance, verify requirements are met, and review/audit the progress. During the systems engineering process architectures are generated to better describe and understand the system. The word "architecture" is used in various contexts in the general field of engineering. It is used as a general description of how the subsystems join together to form the system. It can also be a detailed description of an aspect of a system: for example, the Operational, System, and Technical Architectures used in Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR), and software intensive developments. However, Systems Engineering Management as developed in DoD recognizes three universally usable architectures that describe important aspects of the system: functional, physical, and system architectures. This course will focus on these architectures as necessary components of the systems engineering process.

The *Functional Architecture* identifies and structures the allocated functional and performance requirements. The *Physical Architecture* depicts the



Figure 1-3. The Systems Engineering Process

system product by showing how it is broken down into subsystems and components. The *System Architecture* identifies all the products (including enabling products) that are necessary to support the system and, by implication, the processes necessary for development, production/construction, deployment, operations, support, disposal, training, and verification.

Life Cycle Integration

Life cycle integration is achieved through integrated development—that is, concurrent consideration of all life cycle needs during the development process. DoD policy requires integrated development, called Integrated Product and Product Development (IPPD) in DoD, to be practiced at all levels in the acquisition chain of command as will be explained in the chapter on IPPD. Concurrent consideration of all life cycle needs can be greatly enhanced through the use of interdisciplinary teams. These teams are often referred to as Integrated Product Teams (IPTs).

The objective of an Integrated Product Team is to:

- Produce a design solution that satisfies initially defined requirements, and
- Communicate that design solution clearly, effectively, and in a timely manner.

Multi-functional, integrated teams:

- Place balanced emphasis on product and process development, and
- Require early involvement of all disciplines appropriate to the team task.

Design-level IPT members are chosen to meet the team objectives and generally have distinctive competence in:

- Technical management (systems engineering),
- Life cycle functional areas (eight primary functions),

- Technical specialty areas, such as safety, risk management, quality, etc., or
- When appropriate, business areas such as finance, cost/budget analysis, and contracting.

Life Cycle Functions

Life cycle functions are the characteristic actions associated with the system life cycle. As illustrated by Figure 1-4, they are development, production and construction, deployment (fielding), operation, support, disposal, training, and verification. These activities cover the "cradle to grave" life cycle process and are associated with major functional groups that provide essential support to the life cycle process. These key life cycle functions are commonly referred to as the eight primary functions of systems engineering.

The customers of the systems engineer perform the life-cycle functions. The system user's needs are emphasized because their needs generate the requirement for the system, but it must be remembered that all of the life-cycle functional areas generate requirements for the systems engineering process once the user has established the basic need. *Those that perform the primary functions also provide life-cycle representation in designlevel integrated teams.*

Primary Function Definitions

Development includes the activities required to evolve the system from customer needs to product or process solutions.

Manufacturing/Production/Construction 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 subsystems.

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.



Figure 1-4. Primary Life Cycle Functions

Operation is the user function and includes activities necessary to satisfy defined operational objectives and tasks in peacetime and wartime environments.

Support includes the activities necessary to provide operations support, maintenance, logistics, and material management.

Disposal includes the activities necessary to ensure that the disposal of decommissioned, destroyed, or irreparable system components meets all applicable regulations and directives.

Training includes the activities necessary to achieve and maintain the knowledge and skill levels necessary to efficiently and effectively perform operations and support functions.

Verification includes the activities necessary to evaluate progress and effectiveness of evolving system products and processes, and to measure specification compliance.

Systems Engineering Considerations

Systems engineering is a standardized, disciplined management process for development of system solutions that provides a constant approach to system development in an environment of change and uncertainty. It also provides for simultaneous product and process development, as well as a common basis for communication.

Systems engineering ensures that the correct technical tasks get done during development through planning, tracking, and coordinating. Responsibilities of systems engineers include:

- Development of a total system design solution that balances cost, schedule, performance, and risk,
- Development and tracking of technical information needed for decision making,
- Verification that technical solutions satisfy customer requirements,

- Development of a system that can be produced economically and supported throughout the life cycle,
- Development and monitoring of internal and external interface compatibility of the system and subsystems using an open systems approach,
- Establishment of baselines and configuration control, and
- Proper focus and structure for system and major sub-system level design IPTs.

1.5 GUIDANCE

DoD 5000.2-R establishes two fundamental requirements for program management:

- It requires that an Integrated Product and Process approach be taken to design wherever practicable, and
- It requires that a disciplined systems engineering process be used to translate operational needs and/or requirements into a system solution.

Tailoring the Process

System engineering is applied during all acquisition and support phases for large- and small-scale systems, new developments or product improvements, and single and multiple procurements. The process must be tailored for different needs and/or requirements. Tailoring considerations include system size and complexity, level of system definition detail, scenarios and missions, constraints and requirements, technology base, major risk factors, and organizational best practices and strengths.

For example, systems engineering of software should follow the basic systems engineering approach as presented in this course. However, it must be tailored to accommodate the software development environment, and the unique progress tracking and verification problems software development entails. In a like manner, all technology domains are expected to bring their own unique needs to the process.

This course provides a conceptual-level description of systems engineering management. The specific techniques, nomenclature, and recommended methods are not meant to be prescriptive. Technical managers must tailor their systems engineering planning to meet their particular requirements and constraints, environment, technical domain, and schedule/ budget situation.

However, the basic time-proven concepts inherent in the systems engineering approach must be retained to provide continuity and control. For complex system designs, a full and documented understanding of what the system must do should precede development of component performance descriptions, which should precede component detail descriptions. Though some parts of the system may be dictated as a constraint or interface, in general, solving the design problem should start with analyzing the requirements and determining what the system has to do before physical alternatives are chosen. Configurations must be controlled and risk must be managed.

Tailoring of this process has to be done carefully to avoid the introduction of substantial unseen risk and uncertainty. Without the control, coordination, and traceability of systems engineering, an environment of uncertainty results which will lead to surprises. Experience has shown that these surprises almost invariably lead to significant impacts to cost and schedule. Tailored processes that reflect the general conceptual approach of this course have been developed and adopted by profes-sional societies, academia, industry associations, government agencies, and major companies.

1.6 SUMMARY POINTS

• Systems engineering management is a multifunctional process that integrates life cycle functions, the systems engineering problemsolving process, and progressive baselining.

- The systems engineering process is a problem-solving process that drives the balanced development of system products and processes.
- Integrated Product Teams should apply the systems engineering process to develop a life cycle balanced-design solution.
- The systems engineering process is applied to each level of development, one level at a time.
- Fundamental systems engineering activities are Requirements Analysis, Functional Analysis/ Allocation, and Design Synthesis, all of which are balanced by System Analysis and Control.

- Baseline phasing provides for an increasing level of descriptive detail of the products and processes with each application of the systems engineering process.
- Baselining in a nut shell is a concept description that leads to a system definition which, in turn, leads to component definitions, and then to component designs, which finally lead to a product.
- The output of each application of the systems engineering process is a major input to the next process application.

CHAPTER 2

SYSTEMS ENGINEERING MANAGEMENT IN DOD ACQUISITION

2.1 INTRODUCTION

The DoD acquisition process has its foundation in federal policy and public law. The development, acquisition, and operation of military systems is governed by a multitude of public laws, formal DoD directives, instructions and manuals, numerous Service and Component regulations, and many inter-service and international agreements.

Managing the development and fielding of military systems requires three basic activities: technical management, business management, and contract management. As described in this course, systems engineering management is the technical management component of DoD acquisition management.

The acquisition process runs parallel to the requirements generation process and the budgeting process (Planning, Programming, and Budgeting System.) User requirements tend to be event driven by threat. The budgeting process is date driven by constraints of the Congressional calendar. Systems Engineering Management bridges these processes and must resolve the dichotomy of event driven needs, event driven technology development, and a calendar driven budget.

Direction and Guidance

The Office of Management and Budget (OMB) provides top-level guidance for planning, budgeting, and acquisition in OMB Circular A-11, Part 3, and the Supplemental Capital Programming Guide: Planning, Budgeting, and Acquisition of Capital Assets, July 1997. These documents establish the broad responsibilities and ground rules to be followed in funding and acquiring major assets. The departments of the executive branch of government are then expected to draft their own guidance consistent with the guidelines established. The principal guidance for defense system acquisitions is the DoD 5000 series of directives and regulations. These documents reflect the actions required of DoD acquisition managers to:

- Translate operational needs into stable, affordable programs,
- Acquire quality products, and
- Organize for efficiency and effectiveness.

2.2 RECENT CHANGES

The DoD 5000 series documents were revised in 2000 to make the process more flexible, enabling the delivery of advanced technology to warfighters more rapidly and at reduced total ownership cost. The new process encourages multiple entry points, depending on the maturity of the fundamental technologies involved, and the use of evolutionary methods to define and develop systems. This encourages a tailored approach to acquisition and engineering management, but it does not alter the basic logic of the underlying systems engineering process.

2.3 ACQUISITION LIFE CYCLE

The revised acquisition process for major defense systems is shown in Figure 2-1. The process is



Figure 2-1. Revised DoD 5000 Acquisition Process

defined by a series of phases during which technology is defined and matured into viable concepts, which are subsequently developed and readied for production, after which the systems produced are supported in the field.

The process allows for a given system to enter the process at any of the development phases. For example, a system using unproven technology would enter at the beginning stages of the process and would proceed through a lengthy period of technology maturation, while a system based on mature and proven technologies might enter directly into engineering development or, conceivably, even production. The process itself (Figure 2-1) includes four phases of development. The first, Concept and Technology Development, is intended to explore alternative concepts based on assessments of operational needs, technology readiness, risk, and affordability. Entry into this phase does not imply that DoD has committed to a new acquisition program; rather, it is the initiation of a process to determine whether or not a need (typically described in a Mission Need Statement (MNS)) can be met at reasonable levels of technical risk and at affordable costs. The decision to enter into the Concept and Technology Development phase is made formally at the Milestone A forum.

The Concept and Technology Development phase begins with concept exploration. During this stage, concept studies are undertaken to define alternative concepts and to provide information about capability and risk that would permit an objective comparison of competing concepts. A decision review is held after completion of the concept exploration activities. The purpose of this review is to determine whether further technology development is required, or whether the system is ready to enter into system acquisition. If the key technologies involved are reasonably mature and have already been demonstrated, the Milestone Decision Authority (MDA) may agree to allow the system to proceed into system acquisition; if not, the system may be directed into a component advanced development stage. (See Supplement A to this chapter for a definition of Technology Readiness levels.) During this stage, system architecture definition will continue and key technologies will be demonstrated in order to ensure that technical and cost risks are understood and are at acceptable levels prior to entering acquisition. In any event, the

Concept and Technology Development phase ends with a defined system architecture supported by technologies that are at acceptable levels of maturity to justify entry into system acquisition.

Formal system acquisition begins with a Milestone B decision. The decision is based on an integrated assessment of technology maturity, user requirements, and funding. A successful Milestone B is followed by the System Development and Demonstration phase. This phase could be entered directly as a result of a technological opportunity and urgent user need, as well as having come through concept and technology development. The System Development and Demonstration phase consists of two stages of development, system integration and system demonstration. Depending upon the maturity level of the system, it could enter at either stage, or the stages could be combined. This is the phase during which the technologies, components and subsystems defined earlier are first integrated at the system level, and then demonstrated and tested. If the system has never been integrated into a complete system, it will enter this phase at the system integration stage. When subsystems have been integrated, prototypes demonstrated, and risks are considered acceptable, the program will normally enter the system demonstration stage following an interim review by the MDA to ensure readiness. The system demonstration stage is intended to demonstrate that the system has operational utility consistent with the operational requirements. Engineering demonstration models are developed and system level development testing and operational assessments are performed to ensure that the system performs as required. These demonstrations are to be conducted in environments that represent the eventual operational environments intended. Once a system has been demonstrated in an operationally relevant environment, it may enter the Production and Deployment phase.

The **Production and Deployment** phase consists of two stages: production readiness and low rate initial production (LRIP), and rate production and deployment. The decision forum for entry into this phase is the Milestone C event. Again, the fundamental issue as to where a program enters the process is a function of technology maturity, so the possibility exists that a system could enter directly into this phase if it were sufficiently mature, for example, a commercial product to be produced for defense applications. However the entry is made—directly or through the maturation process described, the production readiness and LRIP stage is where initial operational test, live fire test, and low rate initial production are conducted. Upon completion of the LRIP stage and following a favorable Beyond LRIP test report, the system enters the rate production and deployment stage during which the item is produced and deployed to the user. As the system is produced and deployed, the final phase, Sustainment and Disposal, begins.

The last, and longest, phase is the Sustainment and Disposal phase of the program. During this phase all necessary activities are accomplished to maintain and sustain the system in the field in the most cost-effective manner possible. The scope of activities is broad and includes everything from maintenance and supply to safety, health, and environmental management. This period may also include transition from contractor to organic support, if appropriate. During this phase, modifications and product improvements are usually implemented to update and maintain the required levels of operational capability as technologies and threat systems evolve. At the end of the system service life it is disposed of in accordance with applicable classified and environmental laws, regulations, and directives. Disposal activities also include recycling, material recovery, salvage of reutilization, and disposal of by-products from development and production.

The key to this model of the acquisition process is that programs have the flexibility to enter at any of the first three phases described. The decision as to where the program should enter the process is primarily a function of user needs and technology maturity. The MDA makes the decision for the program in question. Program managers are encouraged to work with their users to develop evolutionary acquisition strategies that will permit deliveries of usable capabilities in as short a timeframe as possible, with improvements and enhancements added as needed through continuing definition of requirements and development activities to support the evolving needs.

2.4 SYSTEMS ENGINEERING IN ACQUISITION

As required by DoD 5000.2-R, the systems engineering process shall:

- 1. Transform operational needs and requirements into an integrated system design solution through concurrent consideration of all lifecycle needs (i.e., development, manufacturing, test and evaluation, verification, deployment, operations, support, training and disposal).
- 2. Ensure the compatibility, interoperability and integration of all functional and physical interfaces and ensure that system definition and design reflect the requirements for all system elements: hardware, software, facilities, people, and data; and

- 3. Characterize and manage technical risks.
- 4. Apply scientific and engineering principles to identify security vulnerabilities and to minimize or contain associated information assurance and force protection risks.

These objectives are accomplished with use of the management concepts and techniques described in the chapters which follow in this course. The appli-cation of systems engineering management coin-cides with acquisition phasing. In order to support milestone decisions, major technical reviews are conducted to evaluate system design maturity.

Concept and Technology Development

The Concept and Technology Development phase consists of two pre-acquisition stages of development. The first, **Concept Exploration**, is represented in Figure 2-2. The exploration of concepts is usually accomplished through multiple shortterm studies. Development of these studies is



Figure 2-2. Concept and Technology Development (Concept Exploration Stage)

expected to employ various techniques including the systems engineering process that translates inputs into viable concept architectures whose functionality can be traced to the requirements. In addition, market surveys, Business Process Reengineering activities, operational analysis, and trade studies support the process.

The primary inputs to these activities include requirements, in form of the MNS, assessments of technology opportunities and status, and the outputs from any efforts undertaken to explore potential solutions. When the contractor studies are complete, a specific concept to be pursued is selected based on a integrated assessment of technical performance; technical, schedule and cost risk; as well as other relevant factors. A decision review is then held to evaluate the concept recommended and the state of technology upon which the concept depends. The MDA then makes a decision as to whether the concept development work needs to be extended or redirected, or whether the technology maturity is such that the program can proceed directly to either Mile-stone B (System Development and Demonstration) or Milestone C (Production and Deployment).

If the details of the concept require definition, i.e., the system has yet to be designed and demonstrated previously, or the system appears to be based on technologies that hold significant risk, then it is likely that the system will proceed to the second stage of the Concept and Technology Development phase. This stage, Component Advanced Development, is represented in Figure 2-3. This is also a pre-acquisition stage of development and is usually characterized by extensive involvement of the DoD Science and Technology (S&T) community. The fundamental objectives of this stage of development are to define a systemlevel architecture and to accomplish risk-reduction activities as required to establish confidence that the building blocks of the system are sufficiently well-defined, tested and demonstrated to provide confidence that, when integrated into higher level assemblies and subsystems, they will perform reliably.

Development of a system-level architecture entails continuing refinement of system level requirements based on comparative analyses of the system concepts under consideration. It also requires that consideration be given to the role that the system



Figure 2-3. Concept and Technology Development (Component Advanced Development Stage)

will play in the system of systems of which it will be a part. System level interfaces must be established. Communications and interoperability requirements must be established, data flows defined, and operational concepts refined. Top level planning should also address the strategies that will be employed to maintain the supportability and affordability of the system over its life cycle including the use of common interface standards and open systems architectures. Important design requirements such as interoperability, open systems, and the use of commercial components should also be addressed during this stage of the program.

Risk reduction activities such as modeling and simulation, component testing, bench testing, and man-in-the-loop testing are emphasized as decisions are made regarding the various technologies that must be integrated to form the system. The primary focus at this stage is to ensure that the key technologies that represent the system components (assemblies and sub-systems) are well understood and are mature enough to justify their use in a system design and development effort. The next stage of the life cycle involves engineering development, so research and development (R&D) activities conducted within the science and technology appropriations should be completed during this stage.

System Development and Demonstration

The decision forum for entry into the System Development and Demonstration (SD&D) phase is the Milestone B event. Entry into this phase represents *program initiation*, the formal beginning of a system acquisition effort. This is the government commitment to pursue the program. Entry requires mature technology, validated requirements, and funding. At this point, the program requirement must be defined by an Operational Requirements Document (ORD). This phase consists of two primary stages, system integration (Figure 2-4) and system demonstration (Figure 2-5).



Figure 2-4. System Development and Demonstration (System Integration Stage)



Figure 2-5. System Development and Demonstration (System Demonstration Stage)

There is no hard and fast guidance that stipulates precisely how the systems engineering process is to intersect with the DoD acquisition process. There are no specified technical events, e.g., DoD designated technical reviews, that are to be accomplished during identified stages of the SD&D phase. However, the results of a SD&D phase should support a production go-ahead decision at Milestone C. That being the case, the process described below reflects a configuration control approach that includes a system level design (functional baseline), final preliminary designs (allocated baselines), and detail designs (initial product baselines). Along with their associated documentation, they represent the systems engineering products developed during SD&D that are most likely needed to appropriately support Milestone C.

System Integration is that stage of SD&D that applies to systems that have yet to achieve system level design maturity as demonstrated by the integration of components at the system level in relevant environments. For an unprecedented system

(one not previously defined and developed), this stage will continue the work begun in the component advanced development stage, but the flavor of the effort now becomes oriented to engineering development, rather than the research-oriented efforts that preceded this stage. A formal ORD, technology assessments, and a high-level system architecture have been established. (These will form major inputs to the systems engineering process.) The engineering focus becomes establishment and agreement on system level technical requirements stated such that designs based on those technical requirements will meet the intent of the operational requirements. The system level technical requirements are stabilized and documented in an approved system level requirements specification. In addition, the system-level requirements baseline (functional baseline) is established. This baseline is verified by development and demonstration of prototypes that show that key technologies can be integrated and that associated risks are sufficiently low to justify developing the system.

Program initiation signals the transition from an S&T focus to management by the program office. The R&D community, the users, and the program office may have all been involved in defining the concepts and technologies that will be key to the system development. It is appropriate at this point, therefore, to conduct a thorough requirements analysis and review to ensure that the user, the contractor, and the program office all hold a common view of the requirements and to preserve the lessons learned through the R&D efforts conducted in the earlier phase. The risk at this point can be high, because misunderstandings and errors regarding system-level requirements will flow down to subsequent designs and can eventually result in overruns and even program failure. The contractor will normally use the occasion of the system requirements review early in this stage to set the functional baseline that will govern the flow-down of requirements to lower level items as preliminary designs are elaborated.

The Interim Progress Review held between System Integration and System Demonstration has no established agenda. The agenda is defined by the MDA and can be flexible in its timing and content. Because of the flexibility built into the acquisition process, not all programs will conform to the model presented here. Programs may find themselves in various stages of preliminary design and detailed design as the program passes from one stage of the SD&D phase to the succeeding stage. With these caveats, System Demonstration (Figure 2-5) is the stage of the SD&D phase during which preliminary and detailed designs are elaborated, engineering demonstration models are fabricated, and the system is demonstrated in operationally relevant environments.

System level requirements are flowed down to the lower level items in the architecture and requirements are documented in the item performance specifications, which represent the preliminary design requirements for those items. The item performance specifications and supporting documentation, when finalized, together form the allocated baseline for the system. Design then proceeds toward the elaboration of a detailed design for the product or system. The product baseline is drafted as the design is elaborated. This physical description of the system may change as a result of testing that will follow, but it forms the basis for initial fabrication and demonstration of these items. If the system has been previously designed and fabricated, then, clearly, this process would be curtailed to take advantage of work already completed.

Following the elaboration of the detailed design, components and subsystems are fabricated, integrated, and tested in a bottom-up approach until system level engineering demonstration models are developed. These demonstration models are not, as a rule, production representative systems. Rather, they are system demonstration models, or integrated commercial items, that serve the purpose of enabling the developer to accomplish development testing on the integrated system. These models are often configured specifically to enable testing of critical elements of the system, for example, in the case of an aircraft development, there may be separate engineering demonstration models developed specifically to test the integrated avionics subsystems, while others demonstrate the flying qualities and flight controls subsystems.

For purposes of making decisions relative to progress through the acquisition process, these system-level demonstrations are not intended to be restricted to laboratory test and demonstrations. They are expected to include rigorous demonstrations that the integrated system is capable of performing operationally useful tasks under conditions that, while not necessarily equal to the rigor of formal operational testing, represent the eventual environment in which the system must perform. The result of these demonstrations provide the confidence required to convince the decisionmaker (MDA) that the system is ready to enter the production phase of the life cycle. This implies that the system has demonstrated not only that technical performance is adequate, but also that the affordability, supportability, and producibility risks are sufficiently low to justify a production decision.



Figure 2-6. Production and Deployment

Production and Deployment

Milestone C is the decision forum for entry into the Production and Deployment phase of the program. Like other phases, this phase is also divided into stages of development. Production Readiness and LRIP is the first of these. At this point, system-level demonstrations have been 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), and this is also the means by which manufacturing rates are ramped upward to the rates intended when manufacturing is fully underway.

Following the completion of formal testing, the submission of required Beyond-LRIP and Live Fire 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. Once this has been accomplished and production again becomes stable, detailed audits are held to

confirm that the Product Baseline documentation correctly describes the system being produced. The Product Baseline is then put under formal configuration control.

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.

Sustainment and Disposal

There is no separate milestone decision required for a program to enter this phase of the system life cycle. The requirement for the Sustainment phase is implicit in the decision to produce and deploy the system. This phase overlaps the Production phase. Systems Engineering activities in the Sustainment phase are focused on maintaining the system's performance capability relative to the threat the system faces. If the military threat changes or a technology opportunity emerges, then the system may require modification. These modifications must be approved at an appropriate level for the particular change being considered. The change then drives the initiation of new systems engineering processes, starting the cycle (or parts of it) all over again.



Figure 2-7. Sustainment and Disposal

Also, in an evolutionary development environment, there will be a continuing effort to develop and refine additional operational requirements based on the experience of the user with the portion of the system already delivered. As new requirements are generated, a new development cycle begins, with technology demonstrations, risk reduction, system demonstrations and testing—the same cycle just described—all tailored to the specific needs and demands of the technology to be added to the core system already delivered.

The final activity in the system life cycle is Disposal. System engineers plan for and conduct system disposal throughout the life cycle beginning with concept development. System components can require disposal because of decommissioning, their destruction, or irreparable damage. In addition, processes and material used for development, production, operation, or maintenance can raise disposal issues throughout the life cycle. Disposal must be done in accordance with applicable laws, regulations, and directives that are continually changing, usually to require more severe constraints. They mostly relate to security and environment issues that include recycling, material recovery, salvage, and disposal of by-products from development and production.

Every Development is Different

The process described above is intended to be very flexible in application. There is no "typical" system acquisition. The process is therefore defined to accommodate a wide range of possibilities, from systems that have been proven in commercial applications and are being purchased for military use, to systems that are designed and developed essentially from scratch. The path that the system development takes through the process will depend primarily on the level of maturity of the technology employed. As explained in the preceding discussion, if the system design will rely significantly on the use of proven or commercial items, then process can be adjusted to allow the system to skip phases, or move quickly from stage to stage within phases. If the type of system is well understood within the applicable technical domains, or it is an advanced version of a current well understood system, then the program definition and risk reduction efforts could be adjusted appropriately.

It is the role of the system engineer to advise the program manager of the recommended path that the development should take, outlining the reasons for that recommendation. The decision as to the appropriate path through the process is actually made by the MDA, normally based on the recommendation of the program manager. The process must be tailored to the specific development, both because it is good engineering and because it is DoD policy as part of the Acquisition Reform initiative. But tailoring must done with the intent of preserving the requirements traceability, baseline control, lifecycle focus, maturity tracking, and integration inherent in the systems engineering approach. The validity of tailoring the process should always be a risk management issue. Acquisition Reform issues will be addressed again in Part IV of this text.

2.5 SUMMARY POINTS

- The development, acquisition, and operation of military systems is governed by a multitude of public laws, formal DoD directives, instructions and manuals, numerous Service and Component regulations, and many inter-service and international agreements.
- The system acquisition life cycle process is a model used to guide the program manager through the process of maturing technology based systems and readying them for production and deployment to military users.
- The acquisition process model is intended to be flexible and to accommodate systems and technologies of varying maturities. Systems dependent on immature technologies will take longer to develop and produce, while those that employ mature technologies can proceed through the process relatively quickly.
- The system engineering effort is integrated into the systems acquisition process such that the activities associated with systems engineering

(development of documentation, technical reviews, configuration management, etc.) support and strengthen the acquisition process. The challenge for the engineering manager is to ensure that engineering activities are conducted at appropriate points in the process to ensure that the system has, in fact, achieved the levels of maturity expected prior to progressing into succeeding phases.

CHAPTER 3

SYSTEMS ENGINEERING PROCESS OVERVIEW

3.1 THE PROCESS

The Systems Engineering Process (SEP) is a comprehensive, iterative and recursive problem solving process, applied sequentially top-down by integrated teams. It transforms needs and requirements into a set of system product and process descriptions, generate information for decision makers, and provides input for the next level of development. The process is applied sequentially, one level at a time, adding additional detail and definition with each level of development. As shown by Figure 3-1, the process includes: inputs and outputs; requirements analysis; functional analysis and allocation; requirements loop; synthesis; design loop; verification; and system analysis and control.

Systems Engineering Process Inputs

Inputs consist primarily of the customer's needs, objectives, requirements and project constraints.



Figure 3-1. The Systems Engineering Process

Inputs can include, but are not restricted to, missions, measures of effectiveness, environments, available technology base, output requirements from prior application of the systems engineering process, program decision requirements, and requirements based on "corporate knowledge."

Requirements Analysis

The first step of the Systems Engineering Process is to analyze the process inputs. Requirements analysis is used to develop functional and performance requirements; that is, customer requirements are translated into a set of requirements that define what the system must do and how well it must perform. The systems engineer must ensure that the requirements are understandable, unambiguous, comprehensive, complete, and concise.

Requirements analysis must clarify and define functional requirements and design constraints. Functional requirements define quantity (how many), quality (how good), coverage (how far), time lines (when and how long), and availability (how often). Design constraints define those factors that limit design flexibility, such as: environmental conditions or limits; defense against internal or external threats; and contract, customer or regulatory standards.

Functional Analysis/Allocation

Functions are analyzed by decomposing higherlevel functions identified through requirements analysis into lower-level functions. The performance requirements associated with the higher level are allocated to lower functions. The result is a description of the product or item in terms of what it does logically and in terms of the performance required. This description is often called the functional architecture of the product or item. Functional analysis and allocation allows for a better understanding of what the system has to do, in what ways it can do it, and to some extent, the priorities and conflicts associated with lower-level functions. It provides information essential to optimizing physical solutions. Key tools in functional analysis and allocation are Functional Flow

Block Diagrams, Time Line Analysis, and the Requirements Allocation Sheet.

Requirements Loop

Performance of the functional analysis and allocation results in a better understanding of the requirements and should prompt reconsideration of the requirements analysis. Each function identified should be traceable back to a requirement. This iterative process of revisiting requirements analysis as a result of functional analysis and allocation is referred to as the requirements loop.

Design Synthesis

Design synthesis is the process of defining the product or item in terms of the physical and software elements which together make up and define the item. The result is often referred to as the physical architecture. Each part must meet at least one functional requirement, and any part may support many functions. The physical architecture is the basic structure for generating the specifications and baselines.

Design Loop

Similar to the requirements loop described above, the design loop is the process of revisiting the functional architecture to verify that the physical design synthesized can perform the required functions at required levels of performance. The design loop permits reconsideration of how the system will perform its mission, and this helps optimize the synthesized design.

Verification

For each application of the system engineering process, the solution will be compared to the requirements. This part of the process is called the verification loop, or more commonly, Verification. Each requirement at each level of development must be verifiable. Baseline documentation developed during the systems engineering process must establish the method of verification for each requirement. Appropriate methods of verification include examination, demonstration, analysis (including modeling and simulation), and testing. Formal test and evaluation (both developmental and operational) are important contributors to the verification of systems.

Systems Analysis and Control

Systems Analysis and Control include technical management activities required to measure progress, evaluate and select alternatives, and document data and decisions. These activities apply to all steps of the systems engineering process.

System analysis activities include trade-off studies, effectiveness analyses, and design analyses. They evaluate alternative approaches to satisfy technical requirements and program objectives, and provide a rigorous quantitative basis for selecting performance, functional, and design requirements. Tools used to provide input to analysis activities include modeling, simulation, experimentation, and test.

Control activities include risk management, configuration management, data management, and performance-based progress measurement including event-based scheduling, Technical Performance Measurement (TPM), and technical reviews.

The purpose of Systems Analysis and Control is to ensure that:

- Solution alternative decisions are made only after evaluating the impact on system effectiveness, life cycle resources, risk, and customer requirements,
- Technical decisions and specification requirements are based on systems engineering outputs,

- Traceability from systems engineering process inputs to outputs is maintained,
- Schedules for development and delivery are mutually supportive,
- Required technical disciplines are integrated into the systems engineering effort,
- Impacts of customer requirements on resulting functional and performance requirements are examined for validity, consistency, desirability, and attainability, and,
- Product and process design requirements are directly traceable to the functional and performance requirements they were designed to fulfill, and vice versa.

Systems Engineering Process Output

Process output is dependent on the level of development. It will include the decision database, the system or configuration item architecture, and the baselines, including specifications, appropriate to the phase of development. In general, it is any data that describes or controls the product configuration or the processes necessary to develop that product.

3.2 SUMMARY POINTS

- The system engineering process is the engine that drives the balanced development of system products and processes applied to each level of development, one level at a time.
- The process provides an increasing level of descriptive detail of products and processes with each system engineering process application. The output of each application is the input to the next process application.

CHAPTER 4 REQUIREMENTS ANALYSIS

4.1 SYSTEMS ENGINEERING PROCESS INPUTS

The inputs to the process include the customer's requirements and the project constraints. Requirements relate directly to the performance characteristics of the system being designed. They are the stated life-cycle customer needs and objectives for the system, and they relate to how well the system will work in its intended environment.

Constraints are conditions that exist because of limitations imposed by external interfaces, project support, technology, or life cycle support systems. Constraints bound the development teams' design opportunities.

Requirements are the primary focus in the systems engineering process because the process's primary purpose is to transform the requirements into designs. The process develops these designs within the constraints. They eventually must be verified to meet both the requirements and constraints.

Types of Requirements

Requirements are categorized in several ways. The following are common categorizations of requirements that relate to technical management:

Customer Requirements: Statements of fact and assumptions that define the expectations of the system in terms of mission objectives, environment, constraints, and measures of effectiveness and suitability (MOE/MOS). The customers are those that perform the eight primary functions of systems engineering (Chapter 1), with special emphasis on the operator as the key customer. Operational requirements will define the basic need and, at a minimum, answer the questions posed in Figure 4-1.

Operational distribution or deployment: Where will the system be used?

Mission profile or scenario: How will the system accomplish its mission objective?

Performance and related parameters: What are the critical system parameters to accomplish the mission?

Utilization environments: How are the various system components to be used?

Effectiveness requirements: How effective or efficient must the system be in performing its mission?

Operational life cycle: How long will the system be in use by the user?

Environment: What environments will the system be expected to operate in an effective manner?

Figure 4-1. Operational Requirements – Basic Questions

Functional Requirements: The necessary task, action or activity that must be accomplished. Functional (what has to be done) requirements identified in requirements analysis will be used as the top-level functions for functional analysis.

Performance Requirements: The extent to which a mission or function must be executed; generally measured in terms of quantity, quality, coverage, timeliness or readiness. During requirements analysis, performance (how well does it have to be done) requirements will be interactively developed across all identified functions based on system life cycle factors; and characterized in terms of the degree of certainty in their estimate, the degree of criticality to system success, and their relationship to other requirements.

Design Requirements: The "build to," "code to," and "buy to" requirements for products and "how to execute" requirements for processes expressed in technical data packages and technical manuals.

Derived Requirements: Requirements that are implied or transformed from higher-level requirement. For example, a requirement for long range or high speed may result in a design requirement for low weight.

Allocated Requirements: A requirement that is established by dividing or otherwise allocating a high-level requirement into multiple lower-level requirements. Example: A 100-pound item that consists of two subsystems might result in weight requirements of 70 pounds and 30 pounds for the two lower-level items.

Attributes of Good Requirements

The attributes of good requirements include the following:

• A requirement must be achievable. It must reflect need or objective for which a solution is technically achievable at costs considered affordable.

- It must be verifiable—that is, not defined by words such as excessive, sufficient, resistant, etc. The expected performance and functional utility must be expressed in a manner that allows verification to be objective, preferably quantitative.
- A requirement must be unambiguous. It must have but one possible meaning.
- It must be complete and contain all mission profiles, operational and maintenance concepts, utilization environments and constraints. All information necessary to understand the customer's need must be there.
- It must be expressed in terms of need, not solution; that is, it should address the "why" and "what" of the need, not how to do it.
- It must be consistent with other requirements. Conflicts must be resolved up front.
- It must be appropriate for the level of system hierarchy. It should not be too detailed that it constrains solutions for the current level of design. For example, detailed requirements relating to components would not normally be in a system-level specification.

4.2 REQUIREMENTS ANALYSIS

Requirements analysis involves defining customer needs and objectives in the context of planned customer use, environments, and identified system characteristics to determine requirements for system functions. Prior analyses are reviewed and updated, refining mission and environment definitions to support system definition.

Requirements analysis is conducted iteratively with functional analysis to optimize performance requirements for identified functions, and to verify that synthesized solutions can satisfy customer requirements. The purpose of Requirements Analysis is to:

- Refine customer objectives and requirements;
- Define initial performance objectives and refine them into requirements;
- Identify and define constraints that limit solutions; and
- Define functional and performance requirements based on customer provided measures of effectiveness.

In general, Requirements Analysis should result in a clear understanding of:

- Functions: What the system has to do,
- Performance: How well the functions have to be performed,
- Interfaces: Environment in which the system will perform, and
- Other requirements and constraints.

The understandings that come from requirements analysis establish the basis for the functional and physical designs to follow. Good requirements analysis is fundamental to successful design definition.

Inputs

Typical inputs include customer needs and objectives, missions, MOE/MOS, environments, key performance parameters (KPPs), technology base, output requirements from prior application of SEP, program decision requirements, and suitability requirements. (See Figure 4-2 for additional considerations.)

Input requirements must be comprehensive and defined for both system products and system processes such as development, manufacturing, verification, deployment, operations, support, training and disposal (eight primary functions).

Role of Integrated Teams

The operator customers have expertise in the operational employment of the product or item being developed. The developers (government and contractor) are not necessarily competent in the operational aspects of the system under development. Typically, the operator's need is neither clearly nor completely expressed in a way directly



Figure 4-2. Inputs to Requirements Analysis

usable by developers. It is unlikely that developers will receive a well-defined problem from which they can develop the system specification. Thus, teamwork is necessary to understand the problem and to analyze the need. It is imperative that customers are part of the definition team.

On the other hand, customers often find it easier to describe a system that attempts to solve the problem rather than to describe the problem itself. Although these "solutions" may be workable to some extent, the optimum solution is obtained through a proper technical development effort that properly balances the various customer mission objectives, functions, MOE/MOS, and constraints. An integrated approach to product and process development will balance the analysis of requirements by providing understanding and accommodation among the eight primary functions.

Requirements Analysis Questions

Requirements Analysis is a process of inquiry and resolution. The following are typical questions that can initiate the thought process:

- What are the reasons behind the system development?
- What are the customer expectations?
- Who are the users and how do they intend to use the product?
- What do the users expect of the product?
- What is their level of expertise?
- With what environmental characteristics must the system comply?
- What are existing and planned interfaces?
- What functions will the system perform, expressed in customer language?
- What are the constraints (hardware, software, economic, procedural) to which the system must comply?

• What will be the final form of the product: such as model, prototype, or mass production?

This list can start the critical, inquisitive outlook necessary to analyze requirements, but it is only the beginning. A tailored process similar to the one at the end of this chapter must be developed to produce the necessary requirements analysis outputs.

4.3 REQUIREMENTS ANALYSIS OUTPUTS

The requirements that result from requirements analysis are typically expressed from one of three perspectives or views. These have been described as the Operational, Functional, and Physical views. All three are necessary and must be coordinated to fully understand the customers' needs and objectives. All three are documented in the decision database.

Operational View

The Operational View addresses how the system will serve its users. It is useful when establishing requirements of "how well" and "under what condition." Operational view information should be documented in an operational concept document that identifies:

- Operational need definition,
- System mission analysis,
- Operational sequences,
- Operational environments,
- Conditions/events to which a system must respond,
- Operational constraints on system,
- Mission performance requirements,
- User and maintainer roles (defined by job tasks and skill requirements or constraints),

- Structure of the organizations that will operate, support and maintain the system, and
- Operational interfaces with other systems.

Analyzing requirements requires understanding the operational and other life cycle needs and constraints.

Functional View

The Functional View focuses on WHAT the system must do to produce the required operational behavior. It includes required inputs, outputs, states, and transformation rules. The functional requirements, in combination with the physical requirements shown below, are the primary sources of the requirements that will eventually be reflected in the system specification. Functional View information includes:

- System functions,
- System performance,
 - Qualitative how well
 - Quantitative how much, capacity
 - Timeliness how often
- Tasks or actions to be performed,
- Inter-function relationships,
- Hardware and software functional relationships,
- Performance constraints,
- Interface requirements including identification of potential open-system opportunities (potential standards that could promote open systems should be identified),
- Unique hardware or software, and
- Verification requirements (to include inspection, analysis/simulation, demo, and test).

Physical View

The Physical View focuses on HOW the system is constructed. It is key to establishing the physical interfaces among operators and equipment, and technology requirements. Physical View information would normally include:

- Configuration of System:
 - Interface descriptions,
 - Characteristics of information displays and operator controls,
 - Relationships of operators to system/ physical equipment, and
 - Operator skills and levels required to perform assigned functions.
- Characterization of Users:
 - Handicaps (special operating environments), and
 - Constraints (movement or visual limitations).
- System Physical Limitations:
 - Physical limitations (capacity, power, size, weight),
 - Technology limitations (range, precision, data rates, frequency, language),
 - Government Furinished Equipment (GFE), Commercial-Off-the-Shelf (COTS), Nondevelopmental Item (NDI), reusability requirements, and
 - Necessary or directed standards.

4.4 SUMMARY POINTS

- An initial statement of a need is seldom defined clearly.
- A significant amount of collaboration between various life cycle customers is necessary to produce an acceptable requirements document.
- Requirements are a statement of the problem to be solved. Unconstrained and nonintegrated requirements are seldom sufficient for designing a solution.

• Because requirements from different customers will conflict, constraints will limit options, and resources are not unlimited; trade studies must be accomplished in order to select a balanced set of requirements that provide feasible solutions to customer needs.

CHAPTER 5

FUNCTIONAL ANALYSIS AND ALLOCATION

5.1 INTRODUCTION

The purpose of this systems engineering process activity is to transform the functional, performance, interface and other requirements that were identified through requirements analysis into a coherent description of system functions that can be used to guide the Design Synthesis activity that follows. The designer will need to know what the system must do, how well, and what constraints will limit design flexibility.

This is accomplished by arranging functions in logical sequences, decomposing higher-level functions into lower-level functions, and allocating performance from higher- to lower-level functions. The tools used include functional flow block diagrams and timeline analysis; and the product is a functional architecture, i.e., a description of the system—but in terms of functions and performance parameters, rather than a physical description. Functional Analysis and Allocation facilitates traceability from requirements to the solution descriptions that are the outcome of Design Synthesis.

Functions are discrete actions (use action verbs) necessary to achieve the system's objectives. These functions may be stated explicitly, or they may be derived from stated requirements. The functions will ultimately be performed or accomplished through use of equipment, personnel, facilities, software, or a combination.

5.2 FUNCTIONAL ANALYSIS AND ALLOCATION

Functional and performance requirements at any level in the system are developed from higher-level

requirements. Functional Analysis and Allocation is repeated to define successively lower-level functional and performance requirements, thus defining architectures at ever-increasing levels of detail. System requirements are allocated and defined in sufficient detail to provide design and verification criteria to support the integrated system design.

This top-down process of translating systemlevel requirements into detailed functional and performance design criteria includes:

- Defining the system in functional terms, then decomposing the top-level functions into subfunctions. That is, identifying at successively lower levels what actions the system has to do,
- Translating higher-level performance requirements into detailed functional and performance design criteria or constraints. That is, identifying how well the functions have to be performed,
- Identifying and defining all internal and external functional interfaces,
- Identifying functional groupings to minimize and control interfaces (functional partitioning),
- Determining the functional characteristics of existing or directed components in the system and incorporating them in the analysis and allocation,
- Examining all life cycle functions, including the eight primary functions, as appropriate for the specific project,
- Performing trade studies to determine alternativefunctional approaches to meet requirements, and

• Revisiting the requirements analysis step as necessary to resolve functional issues.

The objective is to identify the functional, performance, and interface design requirements; it is not to design a solution...yet!

Functional Partitioning

Functional partitioning is the process of grouping functions that logically fit with the components likely to be used, and to minimize functional interfaces. Partitioning is performed as part of functional decomposition. It identifies logical groupings of functions that facilitate the use of modular components and open-system designs. Functional partitioning is also useful in understanding how existing equipment or components (including commercial) will function with or within the system.

Requirements Loop

During the performance of the Functional Analysis and Allocation process, it is expected that revisiting the requirements analysis process will be necessary. This is caused by the emergence of functional issues that will require re-examination of the higher-level requirements. Such issues might include directed components or standards that cause functional conflict, identification of a revised approach to functional sequencing, or, most likely, a conflict caused by mutually incompatible requirements.

Figure 5-1 gives an overview of the basic parameters of Functional Analysis and Allocation. The output of the process is the functional architecture. In its most basic form, the functional architecture is a simple hierarchical decomposition of the functions with associated performance requirements. As the architecture definition is refined and made more specific with the performance of the

• Outputs:

- Functional architecture and supporting detail

- Inputs:
 - Outputs of the Requirements Analysis
- Enablers:
 - Multi-discipline product teams, decision database; Tools & Models, such as QFD, Functional Flow Block Diagrams, IDEF, N2 charts, Requirement Allocation Sheet, Timelines, Data Flow Diagrams, State/Mode Diagrams, Behavior Diagrams
- Controls:
 - Constraints; GFE, COTS, & Reusable S/W; System concept & subsystem choices; organizational procedures

• Activities:

- Define system states and modes
- Define system functions & external interfaces
- Define functional interfaces
- Allocate performance requirements to functions
- Analyze performance
- Analyze timing and resources
- Analyze failure mode effects and criticality
- Define fault detection and recovery behavior
- Integrate functions



Figure 5-1. Functional Analysis and Allocation

activities listed in Figure 5-1, the functional architecture becomes more detailed and comprehensive. These activities provide a functional architecture with sufficient detail to support the Design Synthesis. They are performed with the aid of traditional tools that structure the effort and provide documentation for traceability. There are many tools available. The following are traditional tools that represent and explain the primary tasks of Functional Analysis and Allocation (several of these are defined and illustrated beginning on page 49):

- Functional flow block diagrams that define task sequences and relationships,
- IDEF0 diagrams that define process and data flows,
- Timeline analyses that define the time sequence of time critical functions, and
- Requirements allocation sheets that identify allocated performance and establish traceability of performance requirements.

5.3 FUNCTIONAL ARCHITECTURE

The functional architecture is a top-down decomposition of system functional and performance requirements. The architecture will show not only the functions that have to be performed, but also the logical sequencing of the functions and performance requirements associated with the functions. It also includes the functional description of existing and government-furnished items to be used in the system. This may require reverse engineering of these existing components.

The functional architecture produced by the Functional Analysis and Allocation process is the detailed package of documentation developed to analyze the functions and allocate performance requirements. It includes the functional flow block diagrams, timeline sheets, requirements allocation sheets, IDEFO diagrams, and all other documentation developed to describe the functional characteristics of the system. However, there is a basic logic to the functional architecture, which in its preliminary form is presented in the example of Figure 5-2. The Functional Analysis and Allocation process would normally begin with the



Figure 5-2. Functional Architecture Example

IPT drafting such a basic version of the architecture. This would generally give the IPT an understanding of the scope and direction of the effort.

Functional Architecture Example

The Marine Corps has a requirement to transport troops in squad-level units over a distance of 50 kilometers. Troops must be transported within 90 minutes from the time of arrival of the transport system. Constant communication is required during the transportation of troops. Figure 5-2 illustrates a preliminary functional architecture for this simple requirement.

5.4 SUMMARY POINTS

Functional analysis begins with the output of requirements analysis (that is, the identification of higher-level functional and performance requirements). Functional Analysis and Allocation consists of decomposition of higher-level functions to lower-levels and then allocation of requirements to those functions.

- There are many tools available to support the development of a Functional Architecture, such as: functional-flow block diagrams, timeline analysis sheet, requirements allocation sheet, Integrated Definition, and others.
- Use of the tools illustrated in this chapter is not mandatory, but the process they represent is:
 - Define task sequences and relationships (functional flow block diagram (FFBD)),
 - Define process and data flows (IDEF0 diagrams),
 - Define the time sequence of time-critical functions (timeline analysis sheets (TLS)), and
 - Allocate performance and establish traceability of performance requirements (requirements allocation sheets (RAS)).
CHAPTER 6 DESIGN SYNTHESIS

6.1 DESIGN DEVELOPMENT

Design Synthesis is the process by which concepts or designs are developed based on the functional descriptions that are the products of Functional Analysis and Allocation. Design synthesis is a creative activity that develops a physical architecture (a set of product, system, and/or software elements) capable of performing the required functions within the limits of the performance parameters prescribed. Since there may be several hardware and/ or software architectures developed to satisfy a given set of functional and performance requirements, synthesis sets the stage for trade studies to select the best among the candidate architectures. The objective of design synthesis is to combine and restructure hardware and software components in such a way as to achieve a design solution capable of satisfying the stated requirements. During concept development, synthesis produces system concepts and establishes basic relationships among the subsystems. During preliminary and detailed design, subsystem and component descriptions are elaborated, and detailed interfaces between all system components are defined.

The physical architecture forms the basis for design definition documentation, such as, specifications, baselines, and work breakdown structures (WBS). Figure 6-1 gives an overview of the basic parameters of the synthesis process.



Figure 6-1. Design Synthesis

Characteristics

Physical architecture is a traditional term. Despite the name, it includes software elements as well as hardware elements. Among the characteristics of the physical architecture (the primary output of Design Synthesis) are the following:

- The correlation with functional analysis requires that each physical or software component meets at least one (or part of one) functional requirement, though any component can meet more than one requirement,
- The architecture is justified by trade studies and effectiveness analyses,
- A product WBS is developed from the physical architecture,
- Metrics are developed to track progress among KPPs, and
- All supporting information is documented in a database.

Modular Designs

Modular designs are formed by grouping components that perform a single independent function or single logical task; have single entry and exit points; and are separately testable. Grouping related functions facilitates the search for modular design solutions and furthermore increases the possibility that open-systems approaches can be used in the product architecture.

Desirable attributes of the modular units include low coupling, high cohesion, and low connectivity. Coupling between modules is a measure of their interdependence, or the amount of information shared between two modules. Decoupling modules eases development risks and makes later modifications easier to implement. Cohesion (also called binding) is the similarity of tasks performed within the module. High cohesion is desirable because it allows for use of identical or like (family or series) components, or for use of a single component to perform multiple functions. Connectivity refers to the relationship of internal elements within one module to internal elements within another module. High connectivity is undesirable in that it creates complex interfaces that may impede design, development, and testing.

Design Loop

The design loop involves revisiting the functional architecture to verify that the physical architecture developed is consistent with the functional and performance requirements. It is a mapping between the functional and physical architectures. Figure 6-2 shows an example of a simple physical architecture and how it relates to the functional architecture. During design synthesis, re-evaluation of the functional analysis may be caused by the discovery of design issues that require re-examination of the initial decomposition, performance allocation, or even the higher-level requirements. These issues might include identification of a promising physical solution or open-system opportunities that have different functional characteristics than those foreseen by the initial functional architecture requirements.

6.2 SYNTHESIS TOOLS

During synthesis, various analytical, engineering, and modeling tools are used to support and document the design effort. Analytical devices such as trade studies support decisions to optimize physical solutions. Requirements Allocation Sheets (RAS) provide traceability to the functional and performance requirements. Simple descriptions like the Concept Decription Sheet (CDS) help visualize and communicate the system concept. Logic models, such as the Schematic Block Diagram (SBD), establish the design and the interrelationships within the system.

Automated engineering management tools such as Computer-Aided Design (CAD), Computer-Aided-Systems Engineering (CASE), and the Computer-Aided-Engineering (CAE) can help organize, coordinate and document the design effort. CAD generates detailed documentation describing the product design including SBDs, detailed



Figure 6–2. Functional/Physical Matrix

drawings, three dimensional and solid drawings, and it tracks some technical performance measurements. CAD can provide significant input for virtual modeling and simulations. It also provides a common design database for integrated design developments. Computer-Aided Engineering can provide system requirements and performance analysis in support of trade studies, analysis related to the eight primary functions, and cost analyses. Computer-Aided Systems Engineering can provide automation of technical management analyses and documentation.

Modeling

Modeling techniques allow the physical product to be visualized and evaluated prior to design decisions. Models allow optimization of hardware and software parameters, permit performance predictions to be made, allow operational sequences to be derived, and permit optimum allocation of functional and performance requirements among the system elements. The traditional logical prototyping used in Design Synthesis is the Schematic Block Diagram.

6.3 SUMMARY POINTS

• Synthesis begins with the output of Functional Analysis and Allocation (the functional architecture). The functional architecture is transformed into a physical architecture by defining physical components needed to perform the functions identified in Functional Analysis and Allocation.

- Many tools are available to support the development of a physical architecture:
 - Define and depict the system concept (CDS),
 - Define and depict components and their relationships (SBD), and
- Establish traceability of performance requirements to components (RAS).
- Specifications and the product WBS are derived from the physical architecture.

CHAPTER 7 VERIFICATION

7.1 GENERAL

The Verification process confirms that Design Synthesis has resulted in a physical architecture that satisfies the system requirements. Verification represents the intersection of systems engineering and test and evaluation.

Verification Objectives

The objectives of the Verification process include using established criteria to conduct verification of the physical architecture (including software and interfaces) from the lowest level up to the total system to ensure that cost, schedule, and performance requirements are satisfied with acceptable levels of risk. Further objectives include generating data (to confirm that system, subsystem, and lower level items meet their specification requirements) and validating technologies that will be used in system design solutions. A method to verify each requirement must be established and recorded during requirements analysis and functional allocation activities. (If it can not be verified it can not be a legitimate requirement.) The verification list should have a direct relationship to the requirements allocation sheet and be continually updated to correspond to it.



Figure 7-1. Systems Engineering and Verification

Verification Activities

System design solutions are verified by the following types of activities:

- Analysis the use of mathematical modeling and analytical techniques to predict the compliance of a design to its requirements based on calculated data or data derived from lower level component or subsystem testing. It is generally used when a physical prototype or product is not available or not cost effective. Analysis includes the use of both modeling and simulation which is covered in some detail in chapter 13,
- 2. Inspection the visual examination of the system, component, or subsystem. It is generally used to verify physical design features or specific manufacturer identification,
- Demonstration the use of system, subsystem, or component operation to show that a requirement can be achieved by the system. It is generally used for a basic confirmation of performance capability and is differentiated from testing by the lack of detailed data gathering, or
- 4. Test the use of system, subsystem, or component operation to obtain detailed data to verify performance or to provide sufficient information to verify performance through further analysis. Testing is the detailed quantifying method of verification, and as described later in this chapter, it is ultimately required in order to verify the system design.

Choice of verification methods must be considered an area of potential risk. Use of inappropriate methods can lead to inaccurate verification. Required defining characteristics, such as key performance parameters (KPPs) are verified by demonstration and/or test. Where total verification by test is not feasible, testing is used to verify key characteristics and assumptions used in design analysis or simulation. Validated models and simulation tools are included as analytical verification methods that complement other methods. The focus and nature of verification activities change as designs progress from concept to detailed designs to physical products.

During earlier design stages, verification focuses on proof of concept for system, subsystem and component levels. During later stages, as the product definition effort proceeds, the focus turns to verifying that the system meets the customer requirements. As shown by Figure 7-1, design is a top-down process while the Verification activity is a bottom-up process. Components will be fabricated and tested prior to the subsystems. Subsystems will be fabricated and tested prior to the completed system.

Performance Verification

Performance requirements must be objectively verifiable, i.e., the requirement must be measurable. Where appropriate, Technical Performance Measurements (TPM) and other management metrics are used to provide insight on progress toward meeting performance goals and requirements. IEEE Standard P1220 provides a structure for Verification activity. As shown in Figure 7-2 the structure is comprehensive and provides a good starting point for Verification planning.

7.2 DOD TEST AND EVALUATION

DoD Test and Evaluation (T&E) policies and procedures directly support the system engineering process of Verification. Testing is the means by which objective judgments are made regarding the extent to which the system meets, exceeds, or fails to meet stated objectives. The purpose of evaluation is to review, analyze, and assess data obtained from testing and other means to aid in making systematic decisions. The purpose of DoD T&E is to verify technical performance, operational effectiveness, operational suitability; and it provides essential information in support of decision making.

Common Types of T&E in DoD

T&E policy requires developmental tests. They confirm that technical requirements have been



Figure 7-2. Verification Tasks

satisfied, and independent analysis and tests verify the system's operational effectiveness and suitability. DoD T&E traditionally and by directive is categorized as:

- Developmental T&E which focuses primarily on technical achievement,
- Operational T&E which focuses on operational effectiveness and suitability and includes Early Operational Assessments (EOA), Operational Assessment (OA), Initial Operational Test and

Evaluation (IOT&E), and Follow-On Operational Test and Evaluation (FOT&E), and

• Live Fire T&E which provides assessment of the vulnerability and lethality of a system by subjecting it to real conditions comparable to the required mission.

T&E

The program office plans and manages the test effort to ensure testing is timely, efficient, comprehensive and complete—and that test results are converted into system improvements. Test planning will determine the effectiveness of the verification process. Like all systems engineering planning activities, careful attention to test planning can reduce program risk. The key test planning document is the Test and Evaluation Master Plan (TEMP). This document lays out the objectives, schedule, and resources reflecting program office and operational test organization planning decisions. To ensure integration of this effort, the program office organizes a Test Planning Work Group (TPWG) or Test Working Level IPT (WIPT) to coordinate the test planning effort.

Test Planning Work Group/Test WIPT

The TPWG/Test WIPT is intended to facilitate the integration of test requirements and activities through close coordination between the members who represent the material developer, designer community, logistic community, user, operational tester, and other stakeholders in the system development. The team outlines test needs based on system requirements, directs test design, determines needed analyses for each test, identifies potential users of test results, and provides rapid dissemination of test and evaluation results.

Test and Evaluation Master Plan (TEMP)

The Test and Evaluation Master Plan is a mandatory document prepared by the program office. The operational test organization reviews it and provides the operational test planning for inclusion. The TEMP is then negotiated between the program office and operational test organization. After differences are resolved, it is approved at appropriate high levels in the stakeholder organizations. After approval it becomes binding on managers and designers (similar to the binding nature of the Operational Requirements Document (ORD)).

The TEMP is a valuable Verification tool that provides an excellent template for technology, system, and major subsystem-level Verification planning. The TEMP includes a reaffirmation of the user requirements, and to an extent, an interpretation of what those requirements mean in various operational scenarios. Part I of the required TEMP format is System Introduction, which provides the mission description, threat assessment, MOEs/ MOSs, a system description, and an identification of critical technical parameters. Part II, Integrated Test Program Summary, provides an integrated test program schedule and a description of the overall test management process. Part III, Developmental Test & Evaluation (DT&E) Outline, lays out an overview of DT&E efforts and a description of future DT&E. Part IV, Operational Test & Evaluation (OT&E) Outline, is provided by the operational test organization and includes an OT&E overview, critical operational issues, future OT&E description, and LFT&E description. Part V, Test & Evaluation Resource Summary, identifies the necessary physical resources and activity responsibilities. This last part includes such items as test articles, test sites, test instrumentation, test support equipment, threat representation, test targets and other expendables, operational force test support, simulations, models, test-beds, special requirements, funding, and training.

Key Performance Parameters

Every system will have a set of KPPs that are the performance characteristics that *must* be achieved by the design solution. They flow from the operational requirements and the resulting derived MOEs. They can be identified by the user, the decision authority, or the operational tester. They are documented in the TEMP.

Developmental Test and Evaluation

The DT&E verifies that the design solution meets the system technical requirements and the system is prepared for successful OT&E. DT&E activities assess progress toward resolving critical operational issues, the validity of cost-performance tradeoff decisions, the mitigation of acquisition technical risk, and the achievement of system maturity.

DT&E efforts:

• Identify potential operational and technological capabilities and limitations of the alternative concepts and design options being pursued;



Figure 7-3. DT&E During System Acquisition

- Support the identification of cost-performance tradeoffs by providing analyses of the capabilities and limitations of alternatives;
- Support the identification and description of design technical risks;
- Assess progress toward resolving Critical Operational Issues, mitigating acquisition technical risk, achieving manufacturing process requirements and system maturity;
- Assess validity of assumptions and analysis conclusions; and
- Provide data and analysis to certify the system ready for OT&E, live-fire testing and other required certifications.

Figure 7-3 highlights some of the more significant DT&E focus areas and where they fit in the acquisition life cycle.

Live Fire Test and Evaluation

LFT&E is performed on any Acquisition Category (ACAT) I or II level weapon system that includes features designed to provide protection to the system or its users in combat. It is conducted on a production configured article to provide information concerning potential user casualties, vulnerabilities, and lethality. It provides data that can establish the system's susceptibility to attack and performance under realistic combat conditions.

Operational Test and Evaluation

OT&E programs are structured to determine the operational effectiveness and suitability of a system under realistic conditions, and to determine if the minimum acceptable operational performance requirements as specified in the ORD and reflected by the KPPs have been satisfied. OT&E uses threatrepresentative forces whenever possible, and employs typical users to operate and maintain the system or item under conditions simulating both combat stress and peacetime conditions. Operational tests will use production or production-



Figure 7-4. OT&E During System Acquisition

representative articles for the operational tests that support the full-rate production decision. Live Fire Tests are usually performed during the operational testing period. Figure 7-4 shows the major activities associated with operational testing and where they fit in the DoD acquisition life cycle.

OT&E Differences

Though the overall objective of both DT&E and OT&E is to verify the effectiveness and suitability of the system, there are distinct differences in their specific objects and focus. DT&E primarily focuses on verifying system technical requirements, while OT&E focuses on verifying operational requirements. DT&E is a program office responsibility that is used to develop the design. OT&E is an independent evaluation of design maturity that

is used to determine if the program should proceed to full-rate production. Figure 7-5 lists the major differences between the two.

7.3 SUMMARY POINTS

The Verification activities of the Systems Engineering Process are performed to verify that physical design meets the system requirements.

 DoD T&E policy supports the verification process through a sequence of Developmental, Operational, and Live-Fire tests, analyses, and assessments. The primary management tools for planning and implementing the T&E effort are the TEMP and the integrated planning team.

Development Tests	Operational Tests		
 Controlled by program manager One-on-one tests Controlled environment Contractor environment 	 Controlled by independent agency Many-on-many tests Realistic/tactical environment with operational scenario 		
 Trained, experienced operators Precise performance objectives and threshold measurements Test to specification Developmental, engineering, or production representative test article 	 No system contractor involvement User troops recently trained Performance measures of operational effectiveness and suitability Test to operational requirements Production representative test article 		

Figure 7-5. DT/OT Comparison

CHAPTER 8

SYSTEMS ENGINEERING PROCESS OUTPUTS

8.1 DOCUMENTING REQUIREMENTS AND DESIGNS

Outputs of the systems engineering process consist of the documents that define the system requirements and design solution. The physical architecture developed through the synthesis process is expanded to include enabling products and services to complete the system architecture. *This system level architecture then becomes the reference model for further development of system requirements and documents.* System engineering process outputs include the system and configuration item architectures, specifications, and baselines, and the decision database.

Outputs are dependent on the level of development. They become increasingly technically detailed as system definition proceeds from concept to detailed design. As each stage of system definition is achieved, the information developed forms the input for succeeding applications of the system engineering process.

Architectures: System/Configuration Item

The System Architecture describes the entire system. It includes the physical architecture produced through design synthesis and adds the enabling products and services required for life cycle employment, support, and management. Military Handbook (MIL-HDBK)-881, *Work Breakdown Structures*, provides reference models for weapon systems architectures. As shown by Figure 8-1, MIL-HDBK-881 illustrates the first three levels of typical system architectures. Program Offices can use MIL-HDBK-881 templates during system definition to help develop a top-level architecture tailored to the needs of the specific system considered. The design contractor will normally develop the levels below these first three. Chapter 9 of this text describes the WBS in more detail.

Specifications

A specification is a document that clearly and accurately describes the essential technical requirements for items, materials, or services including the procedures by which it can be determined that the requirements have been met. Specifications help avoid duplication and inconsistencies, allow for accurate estimates of necessary work and resources, act as a negotiation and reference document for engineering changes, provide documentation of configuration, and allow for consistent communication among those responsible for the eight primary functions of Systems Engineering. They provide IPTs a precise idea of the problem to be solved so that they can efficiently design the system and estimate the cost of design alternatives. They provide guidance to testers for verification (qualification) of each technical requirement.

Program-Unique Specifications

During system development a series of specifications are generated to describe the system at different levels of detail. These program unique specifications form the core of the configuration baselines. As shown by Figure 8-2, in addition to referring to different levels within the system hierarchy, these baselines are defined at different phases of the design process.

Initially the system is described in terms of the top-level (system) functions, performance, and interfaces. These technical requirements are derived from the operational requirements established by



Figure 8-1. Example from MIL-HDBK-881

the user. This system-level technical description is documented in the System Specification, which is the primary documentation of the system-level Functional Baseline. The system requirements are then flowed down (allocated) to the items below the system level, such that a set of design criteria are established for each of those items. These item descriptions are captured in a set of Item Performance Specifications, which together with other interface definitions, process descriptions, and drawings, document the Allocated Baseline (sometimes referred to as the "Design To" baseline). Having baselined the design requirements for the individual items, detailed design follows. Detailed design involves defining the system from top to bottom in terms of the physical entities that will be employed to satisfy the design requirements. When detailed design is complete, a final baseline is defined. This is generally referred to as the Product Baseline, and, depending on the stage of development, may reflect a "Build to" or "As built" description. The Product Baseline is documented

by the Technical Data Package, which will include not only Item Detail Specifications, but also, Process and Material Specifications, as well as drawings, parts lists, and other information that describes the final system in full physical detail. Figure 8-3 shows how these specifications relate to their associated baselines.

Role of Specifications

Requirements documents express why the development is needed. Specification documents are an intermediate expression of what the needed system has to do in terms of technical requirements (function, performance, and interface). Design documents (drawings, associated lists, etc.) describe the means by which the design requirements are to be satisfied. Figure 8-4 illustrates how requirements flow down from top-level specifications to design documentation. Preparation of specifications are part of the system engineering process, but also involve techniques that relate to



Figure 8-2. Specifications and Levels of Development

Specification	Content	Baseline
System Spec	Defines mission/technical performance requirements. Allocates requirements to functional areas and defines interfaces.	Functional
Item Performance Spec	Defines performance characteristics of CIs and CSCIs. Details design requirements and with drawings and other documents form the Allocated Baseline.	Allocated "Design To"
ltem Detail Spec	Item Detail SpecDefines form, fit, function, performance, and test requirements for acceptance. (Item, process, and material specs start the Product Baseline effort, but the final audited baseline includes all the items in the TDP.)	
Process Spec	Defines process performed during fabrication.	
Material SpecDefines production of raw materials or semi-fabricated material used in fabrication.		

Figure	8-3.	Specification	Types
--------	------	---------------	-------

communication skills, both legal and editorial. Figure 8-5 provides some rules of thumb that illustrate this.

In summary, specifications document what the system has to do, how well it has to do it, and how to verify it can do it.

Baselines

Baselines formally document a product at some given level of design definition. They are references for the subsequent development to follow. Most DoD systems are developed using the three classic baselines described above: functional, allocated, and product. Though the program unique specifications are the dominant baseline documentation, they alone do not constitute a baseline.

Additional documents include both end and enabling product descriptions. End product baseline documents normally include those describing system requirements, functional architecture, physical architecture, technical drawing package, and requirements traceability. Enabling product baseline documents include a wide range of documents that could include manufacturing plans and processes, supportability planning, supply documentation, manuals, training plans and programs, test planning, deployment planning, and others. All enabling products should be reviewed for their susceptibility to impact from system configuration changes. If a document is one that describes a part of a system and could require change if the configuration changes, then most likely it should be included as a baseline document.

Acquisition Program Baselines

Acquisition Program Baselines and Configuration Baselines are related. To be accurate the Program baseline must reflect the realities of the Configuration Baseline, but the two should not be confused. Acquisition Program Baselines are high level assessments of program maturity and viability. Configuration Baselines are system descriptions. Figure 8-6 provides additional clarification.



Figure 8-4. How Specifications Lead to Design Documents

- Use a table of contents and define all abbreviations and acronyms.
- Use active voice.
- Use "shall" to denote mandatory requirement and "may" or "should" to denote guidance provisions.
- Avoid ambiguous provisions, such as "as necessary," "contractor's best practice," "smooth finish," and similar terms.
- Use the System Engineering Process to identify requirements. Do not over-specify.
- Avoid "tiering." Any mandatory requirement in a document below the first tier, should be stated in the specification.
- Only requirement sections of the MIL-STD-491D formats are binding. Do not put requirements in non-binding sections, such as *Scope*, *Documents*, or *Notes*.
- Data documentation requirements are specified in a Contract Data Requirements List.

Figure 8–5. Rules of Thumb for Specification Preparation

Decision Database

The decision database is the documentation that supports and explains the configuration solution decisions. It includes trade studies, cost effectiveness analyses, Quality Function Deployment (QFD) analysis, models, simulations, and other data generated to understand a requirement, develop alternative solutions, or make a choice between them. These items are retained and controlled as part of the Data Management process described in Chapter 10.

8.2 DOD POLICY AND PRACTICE— SPECIFICATIONS AND STANDARDS

DoD uses specifications to communicate product requirements and standards to provide guidance concerning proven methods and practices.

Specifications

DoD uses three basic classifications of specifications: materiel specifications (developed by DoD components), Program-Unique Specifications, and non-DoD specifications.

- Program Baselines
 - Embody only the most important cost, schedule, and performance objectives and thresholds
 - Threshold breach results in re-evaluation of program at MDA level
 - Selected key performance parameters
 - Specifically evolves over the development cycle and is updated at each major milestone review or program restructure
- Required on ALL programs for measuring and reporting status

- Configuration Baselines Identify and define an item's functional and physical characteristics
 - Functional Baseline Describes system level requirements
 - Allocated Baseline Describes design requirements for items below system level
 - Product Baseline Describes product physical detail
- Documents outputs of Systems Engineering
 Process

Figure 8–6. Acquisition Program Baselines and Configuration Baselines

DoD developed specifications describe essential technical requirements for purchase of materiel. Program-Unique Specifications are an integral part of the system development process. Standard practice for preparation of DoD and Program-Unique Specifications is guided by MIL-STD-961D. This standard provides guidance for the development of performance and detail specifications. MIL-STD-961D, Appendix A provides further guidance for the development of Program-Unique Specifications.

Non-DoD specifications and standards approved for DoD use are listed in the *DoD Index of Specifications and Standards* (DoDISS).

DoD Policy (Specifications)

DoD policy is to develop *performance* specifications for procurement and acquisition. In general, detail specifications are left for contractor development and use. Use of a detail specification in DoD procurement or acquisition should be considered only where absolutely necessary, and then only with supporting trade studies and acquisition authority approval.

DoD policy gives preference to the use of commercial solutions to government requirements, rather than development of unique designs. Therefore, the use of commercial item specifications and descriptions should be a priority in system architecture development. Only when no commercial solution is available should government detail specifications be employed.

In the case of re-procurement, where detail specifications and drawings are government owned, standardization or interface requirements may present a need for use of detailed specifications. Trade studies that reflect total ownership costs and the concerns related to all eight primary functions should govern decisions concerning the type of specification used for re-procurement of systems, subsystems, and configuration items. Such trade studies and cost analysis should be preformed prior to the use of detail specifications or the decision to develop and use performance specifications in a reprocurement.

Performance Specifications

Performance Specifications state requirements in terms of the required results with criteria for verifying compliance, but without stating the methods for achieving the required results. In general, performance specifications define products in terms of functions, performance, and interface requirements. They define the functional requirements for the item, the environment in which it must operate, and interface and interchangeability characteristics. The contractor is provided the flexibility to decide how the requirements are best achieved, subject to the constraints imposed by the government, typically through interface requirements. System Specifications and Item Performance Specifications are examples of performance specifications.

Detail Specifications

Detail Specifications, such as Item Detail, Material and Process Specifications, provide design requirements. This can include materials to be used, how a requirement is to be achieved, or how an item is to be fabricated or constructed. If a specification contains both performance and detail requirements, it is considered a Detail Specification, with the following exception: Interface and interchangeability requirements in Performance Specifications may be expressed in detailed terms. For example, a Performance Specification for shoes would specify size requirements in detailed terms, but material or method of construction would be stated in performance terms.

Software Documentation – IEEE/EIA 12207

IEEE/EIA 12207, *Software Life Cycle Processes*, describes the U.S. implementation of the ISO standard on software processes. This standard describes the development of software specifications as one aspect of the software development process.

The process described in IEEE/EIA 12207 for allocating requirements in a top-down fashion and documenting the requirements at all levels parallels the systems engineering process described in this text. The standard requires first that system-level requirements be allocated to software items (or configuration items) and that the software requirements then be documented in terms of functionality, performance, and interfaces, and that qualification requirements be specified. Software item requirements must be traceable to systemlevel, and be consistent and verifiable.

The developer is then required to decompose each software item into software components and then into software units that can be coded. Requirements are allocated from item level, to component, and finally to unit level. This is the detailed design activity and IEEE/EIA 12207 requires that these allocations of requirements be documented in documents that are referred to as "descriptions," or, if the item is a "stand alone" item, as "specifications." The content of these documents is defined in the IEEE/EIA standard; however, the level of detail required will vary by project. Each project must therefore ensure that a common level of expectation is established among all stakeholders in the software development activity.

Standard Practice for Defense Specifications – MIL-STD-961D

The purpose of MIL-STD-961D is to establish uniform practices for specification preparation, to ensure inclusion of essential requirements, to ensure Verification (qualification) methods are established for each requirement, and to aid in the use and analysis of specification content. MIL-STD-961D establishes the format and content of system, configuration item, software, process and material specifications. These Program-Unique Specifications are developed through application of the systems engineering process and represent a hierarchy as shown in Figure 8-7.

Standards

Standards establish engineering and technical limitations and applications for items, materials, processes, methods, designs, and engineering practices. They are "corporate knowledge" documents describing how to do some process or a



Figure 8–7. Specification Hierarchy

description of a body of knowledge. Standards come from many sources, reflecting the practices or knowledge base of the source. Format and content of Defense Standards, including Handbooks, are governed by MIL-STD-962. Other types of standards in use in DoD include Commercial Standards, Corporate Standards, International Standards, Federal Standards, and Federal Information Processing Standards.

DoD Policy (Standards)

DoD policy does not require standard management approaches or manufacturing processes on contracts. This policy applies to the imposition of both Military Specifications and Standards and, in addition, to the imposition of Commercial and Industry Standards. In general, the preferred approach is to allow contractors to use industry, government, corporate, or company standards they have determined to be appropriate to meet government's needs. The government reviews and accepts the contractor's approach through a contract selection process or a contractual review process.

The government should impose a process or standard only as a last resort, and only with the support of an appropriate trade study analysis. If a specific standard is imposed in a solicitation or contract, a waiver will be required from an appropriate Service authority.

However, there is need on occasion to direct the use of some standards for reasons of standardization, interfaces, and development of open systems. A case in point is the mandated use of the Joint Technical Architecture (JTA) for defining interoperability standards. The JTA sets forth the set of interface standards that are expected to be employed in DoD systems. The JTA is justifiably mandatory because it promotes needed interoperability standardization, establishes supportable interface standards, and promotes the development of open systems.

DoD technical managers should be alert to situations when directed standards are appropriate to their program. Decisions concerning use of directed standards should be confirmed by trade studies and requirements traceability.

DoD Index of Specifications and Standards

The DoDISS lists all international, adopted industry standardization documents authorized for use by the military departments, federal and military specifications and standards. Published in three volumes, it contains over 30,000 documents in 103 Federal Supply Groups broken down into 850 Federal Supply Classes. It covers the total DoD use of specifications and standards, ranging from fuel specifications to international quality standards.

8.3 SUMMARY POINTS

- System Engineering Process Outputs include the system/configuration item architecture, specifications and baselines, and the decision database.
- System/Configuration Item Architectures include the physical architecture and the associated products and services.
- Program-Unique specifications are a primary output of the System Engineering Process. Program-Unique specifications describe what the system or configuration item must accomplish and how it will be verified. Program-Unique specifications include the System, Item Performance, and Item Detail Specifications. The System Specification describes the system requirements, while Item Performance and Item Detail Specifications describe configuration item requirements.
- Configuration baselines are used to manage and control the technical development. Program baselines are used for measuring and supporting program status.
- The Decision Database includes those documents or software that support understanding and decision making during formulation of the configuration baselines.

- DoD policy is to develop *performance* specifications for procurement and acquisition. Use of other than performance specifications in a contract must be justified and approved.
- It is DoD policy not to require standard management approaches or manufacturing processes on contracts.
- Mandatory use of some standard practices are necessary, but must be justified through analysis. A case in point is the mandatory use of the standards listed in the Joint Technical Architecture.

CHAPTER 9

WORK BREAKDOWN STRUCTURE

9.1 INTRODUCTION

The Work Breakdown Structure (WBS) is a means of organizing system development activities based on system and product decompositions. The systems engineering process described in earlier chapters produces system and product descriptions. These product architectures, together with associated services (e.g., program management, systems engineering, etc.) are organized and depicted in a hierarchical tree-like structure that is the WBS. (See Figure 9-1.)

Because the WBS is a direct derivative of the physical and systems architectures it could be considered an output of the systems engineering process. It is being presented here as a Systems Analysis and Control tool because of its essential utility for all aspects of the systems engineering process. It is used to structure development activities, to identify data and documents, and to organize integrated teams, and for other non-technical program management purposes.

WBS Role in DoD Systems Engineering

DoD 5000.2-R requires that a program WBS be established to provide a framework for program and technical planning, cost estimating, resource allocation, performance measurement, and status reporting. The WBS is used to define the total system, to display it as a product-oriented family tree composed of hardware, software, services, data, and facilities, and to relate these elements to each other and to the end product. Program offices are to tailor a program WBS using the guidance provided in MIL-HDBK-881.



Figure 9–1. Architecture to WBS Flow

The program WBS is developed initially to define the top three levels. As the program proceeds through development and is further defined, program managers should ensure that the WBS is extended to identify all high-cost and high-risk elements for management and reporting, while ensuring the contractor has complete flexibility to extend the WBS below the reporting requirement to reflect how work will be accomplished.

Basic Purposes of the WBS

Organizational:

The WBS provides a coordinated, complete, and comprehensive view of program management. It establishes a structure for organizing system development activities, including IPT design, development, and maintenance.

Business:

It provides a structure for budgets and cost estimates. It is used to organize collection and analysis of detailed costs for earned value reports (Cost Performance Reports or Cost/Schedule Control System Criteria reporting).

Technical:

The WBS establishes a structure for:

- Identifying products, processes, and data,
- Organizing risk management analysis and tracking,
- Enabling configuration and data management. It helps establish interface identification and control.
- Developing work packages for work orders and material/part ordering, and
- Organizing technical reviews and audits.

The WBS is used to group product items for specification development, to develop Statements of Work (SOW), and to identify specific contract deliverables.

WBS – Benefits

The WBS allows the total system to be described through a logical breakout of product elements into work packages. A WBS, correctly prepared, will account for all program activity. It links program objectives and activities with resources, facilitates initial budgets, and simplifies subsequent cost reporting. The WBS allows comparison of various independent metrics and other data to look for comprehensive trends.

It is a foundation for all program activities, including program and technical planning, event schedule definition, configuration management, risk management, data management, specification preparation, SOW preparation, status reporting and problem analysis, cost estimates, and budget formulation.

9.2 WBS DEVELOPMENT

The physical and system architectures are used to prepare the WBS. The architectures should be reviewed to ensure that all necessary products and services are identified, and that the top-down structure provides a continuity of flow down for all tasks. Enough levels must be provided to identify work packages for cost/schedule control purposes. If too few levels are identified, then management visibility and integration of work packages may suffer. If too many levels are identified, then program review and control actions may become excessively time-consuming.

- The first three WBS 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.

DoD Practice

In accordance with DoD mandatory procedures in DoD 5000.2-R and common DoD practice as established in MIL-HDBK-881, the program office develops a program WBS and a contract WBS for each contract. The program WBS is the WBS that represents the total system, i.e., the WBS that describes the system architecture. The contract WBS is the part of the program WBS that relates to deliverables and tasks of a specific contract.

MIL-HDBK-881 is used by the program office to support the systems engineering process in developing 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 cost and budget professionals, as well as contracting officers. An integrated team representing these stakeholders should be formed to support WBS development.

WBS Anatomy

A program WBS has an end product part and an enabling product part. The end product part of the

system typically consists of the prime mission product(s) delivered to the operational customer. This part of the WBS is based on the physical architectures developed from operational requirements. It represents that part of the WBS involved in product development. Figure 9-2 presents a simple example of a program WBS product part.

The "enabling product" part of the system includes the products and services required to develop, produce, and support the end product(s). This part of the WBS includes the horizontal elements of the system architecture (exclusive of the end products), and identifies all the products and services necessary to support the life cycle needs of the product. Figure 9-3 shows an example of the top three levels of a complete WBS tree.

Contract WBS

A contract WBS is developed by the program office in preparation for contracting for work required to develop the system. It is further developed by the contractor after contract award. The contract WBS is that portion of the program WBS that is specifically being tasked through the contract. A simple example of a contract WBS derived from the program WBS shown in Figure 9-2 is provided by Figure 9-4. Figure 9-4, like Figure 9-2, only includes the product part of the contract WBS. A



Figure 9-2. Program WBS – The Product Part (Physical Architecture)



Figure 9-3. The Complete Work Breakdown Structure

complete contract WBS would include associated enabling products, similar to those identified in Figure 9-3. The resulting complete contract WBS is used to organize and identify contractor tasks. The program office's preliminary version is used to develop a SOW for the Request for Proposals.



Figure 9–4. Contract WBS

9.3 DESIGNING AND TRACKING WORK

A prime use of the WBS is the design and tracking of work. The WBS is used to establish what work is necessary, a logical decomposition down to work packages, and a method for organizing feedback. As shown by Figure 9-5, the WBS element is matrixed against those organizations in the company responsible for the task. This creates cost accounts and task definition at a detailed level. It allows rational organization of integrated teams and other organizational structures by helping establish what expertise and functional support is required for a specific WBS element. It further allows precise tracking of technical and other management.

WBS Dictionary

As part of the work and cost control use of the WBS, a Work Breakdown Dictionary is developed. For each WBS element a dictionary entry is prepared that describes the task, what costs (activities) apply, and the references to the associated Contract Line Item Numbers and SOW paragraph. An example of a level 2 WBS element dictionary entry is shown as Figure 9-6.

9.4 SUMMARY POINTS

• The WBS is an essential tool for the organization and coordination of systems engineering



Figure 9-5. WBS Control Matrix

Index Item No. 2 WBS Level 2			CONTRACT NUMBER F33657-72-C-0923			
WBS Ele A10100	ement		WBS Title Contract Air Vehicle Line Item:			
Date Chg	Revision	No.	Revision Auth	Approved	0001, 0001AA, 0001AE, 0001/	0001AB, 0001AC, 0001AD AF, 0001AG, 0001AH
Specific 689E078	Specification No. Specification Title: Prime Item Development Prime Item Development 689E078780028 Specification for AGM 86A Air Vehicle/ Airframe			ient 186A Air Vehicle/		
Element Task Description		Cost Description				
Technical Content: Technical Content: The Air Vehicle element task description refers to the effort required to develop, fabricate, integrate and test the airframe segment, portions of the Navigation/Guidance element, and Airborne Development Test Equipment and Airborne Operational Test Equipment and to the integration assembly and check-out of these complete elements, together with the Engine Segment, to produce the complete Air Vehicle. The lower-level element are: Airframe Segment (A11100), Navigation/Guidance Segment (A32100), Airborne Development Test Equipment Test Equipment (A61200).			efers to the effort id test the ion/Guidance Equipment and to the integra- plete elements, duce the eents included it are: on/Guidance ment Test perational Test	MPC/PMC A10100 Cost Content – Sys The cost to be accur a summarization of a fabricate, assemble, i testing, analysis and includes all costs ass integrating, assembli create this element.	Work Order/Work Auth See lower level WBS Elements tem Contractor nulated against this element includes Il costs required to plan, develop, integrate and perform development reporting for the air vehicle. It also sociated with the required efforts in ing and checking our GFP required to Applicable SOW Paragraph 3.6.2	

Figure 9-6. Work Breakdown Dictionary

processes, and it is a product of the systems engineering process.

• Its importance extends beyond the technical community to business professionals and contracting officers. The needs of all stakeholders must be considered in its development. The program office develops the program WBS and a high-level contract WBS for each contract. The

contractors develop the lower levels of the contract WBS associated with their contract.

- The system architecture provides the structure for a program WBS. SOW tasks flow from this WBS.
- The WBS provides a structure for organizing IPTs and tracking metrics.

CHAPTER 10 CONFIGURATION MANAGEMENT

10.1 FOUNDATIONS

Configuration Defined

A "configuration" consists of the functional, physical, and interface characteristics of existing or planned hardware, firmware, software or a combination thereof as set forth in technical documentation and ultimately achieved in a product. The configuration is formally expressed in relation to a Functional, Allocated, or Product configuration baseline as described in Chapter 8.

Configuration Management

Configuration management permits the orderly development of a system, subsystem, or configuration item. A good configuration management program ensures that designs are traceable to requirements, that change is controlled and documented, that interfaces are defined and understood, and that there is consistency between the product and its supporting documentation. Configuration management provides documentation that describes what is supposed to be produced, what is being produced, what has been produced, and what modifications have been made to what was produced.

Configuration management is performed on baselines, and the approval level for configuration modification can change with each baseline. In a typical system development, customers or user representatives control the operational requirements and usually the system concept. The developing agency program office normally controls the functional baseline. Allocated and product baselines can be controlled by the program office, the producer, or a logistics agent depending on the life cycle management strategy. This sets up a hierarchy of configuration control authority corresponding to the baseline structure. Since lower level baselines have to conform to a higher-level baseline, changes at the lower levels must be examined to assure they do not impact a higher-level baseline. If they do, they must be approved at the highest level impacted. For example, suppose the only engine turbine assembly affordably available for an engine development cannot provide the continuous operating temperature required by the allocated baseline. Then not only must the impact of the change at the lower level (turbine) be examined, but the change should also be reviewed for possible impact on the functional baseline, where requirements such as engine power and thrust might reside.

Configuration management is supported and performed by integrated teams in an Integrated Product and Process Development (IPPD) environment. Configuration management is closely associated with technical data management and interface management. Data and interface management is essential for proper configuration management, and the configuration management effort has to include them.

DoD Application of Configuration Management

During the development contract, the Government should maintain configuration control of the functional and performance requirements only, giving contractors responsibility for the detailed design. (SECDEF Memo of 29 Jun 94.) This implies government control of the Functional (system requirements) Baseline. Decisions regarding whether or not the government will take control of the lower-level baselines (allocated and product baselines), and when ultimately depends on the requirements and strategies needed for the particular program. In general, government control of lower-level baselines, if exercised, will take place late in the development program after design has stabilized.

Configuration Management Planning

When planning a configuration management effort you should consider the basics: what has to be done, how should it be done, who should do it, when should it be done, and what resources are required. Planning should include the organizational and functional structure that will define the methods and procedures to manage functional and physical characteristics, interfaces, and documents of the system component. It should also include statements of responsibility and authority, methods of control, methods of audit or verification, milestones, and schedules. EIA IS-649, National Consensus Standard for Configuration Management, and MIL-HDBK-61 can be used as planning guidance.

Configuration Item (CI)

A key concept that affects planning is the configuration item (CI). CI decisions will determine what configurations will be managed. CIs are an aggregation of hardware, firmware, or computer software, or any of their discrete portions, which satisfies an end-use function and is designated for separate configuration management. Any item required for logistic support and designated for separate procurement is generally identified as CI. Components can be designated CIs because of crucial interfaces or the need to be integrated with operation with other components within or outside of the system. An item can be designated CI if it is developed wholly or partially with government funds, including nondevelopmental items (NDI) if additional development of technical data is required. All CIs are directly traceable to the WBS.

Impact of CI Designation

CI designation requires a separate configuration management effort for the CI, or groupings of

related CIs. The decision to place an item, or items, under formal configuration control results in:

- Separate specifications,
- Formal approval of changes,
- Discrete records for configuration status accounting,
- Individual design reviews and configuration audits,
- Discrete identifiers and name plates,
- Separate qualification testing, and
- Separate operating and user manuals.

10.2 CONFIGURATION MANAGEMENT STRUCTURE

Configuration management comprises four interrelated efforts:

- Identification,
- Control,
- Status Accounting, and
- Audits.

Also directly associated with configuration management are data management and interface management. Any configuration management planning effort must consider all six elements.

Identification

Configuration Identification consists of documentation of formally approved baselines and specifications, including:

- Selection of the CIs,
- Determination of the types of configuration documentation required for each CI,

- Documenting the functional and physical characteristics of each CI,
- Establishing interface management procedures, organization, and documentation,
- Issuance of numbers and other identifiers associated with the system/CI configuration structure, including internal and external interfaces, and
- Distribution of CI identification and related configuration documentation.

Configuration Documentation

Configuration documentation is technical documentation that identifies and defines the item's functional and physical characteristics. It is developed, approved, and maintained through three distinct evolutionary increasing levels of detail. The three levels of configuration documentation form the three baselines and are referred to as functional, allocated, and product configuration documentation. These provide the specific technical description of a system or its components at any point in time.

Configuration Control

Configuration Control is the systematic proposal, justification, prioritization, evaluation, coordination, approval or disapproval, and implementation of all approved changes in the configuration of a system/CI after formal establishment of its baseline. In other words, it is how a system (and its CIs) change control process is executed and managed.

Configuration Control provides management visibility, ensures all factors associated with a proposed change are evaluated, prevents unnecessary or marginal changes, and establishes change priorities. In DoD it consists primarily of a change process that formalizes documentation and provides a management structure for change approval.

Change Documents Used for Government Controlled Baselines

There are three types of change documents used to control baselines associated with government configuration management: Engineering Change Proposal, Request for Deviation, and Request for Waivers.

- Engineering Change Proposals (ECP) identify need for a permanent configuration change. Upon approval of an ECP a new configuration is established.
- Requests for Deviation or Waiver propose a temporary departure from the baseline. They allow for acceptance of non-conforming material. After acceptance of a deviation or waiver the documented configuration remains unchanged.

Engineering Change Proposal (ECP)

An ECP is documentation that describes and suggests a change to a configuration baseline. Separate ECPs are submitted for each change that has a distinct objective. To provide advanced notice and reduce paperwork, Preliminary ECPs or Advance Change/Study Notices can be used preparatory to issue of a formal ECP. Time and effort for the approval process can be further reduced through use of joint government and contractor integrated teams to review and edit preliminary change proposals.

ECPs are identified as Class I or Class II. Class I changes require government approval before changing the configuration. These changes can result from problems with the baseline requirement, safety, interfaces, operating/servicing capability, preset adjustments, human interface including skill level, or training. Class I changes can also be used to upgrade already delivered systems to the new configuration through use of retrofit, mod kits, and the like. Class I ECPs are also used to change contractual provisions that do not directly impact the configuration baseline; for example, changes affecting cost, warranties, deliveries, or



Figure 10-1. ECP Designators

data requirements. Class I ECPs require program office approval, which is usually handled through a formal Configuration Control Board, chaired by the government program manager or delegated representative.

Class II changes correct minor conflicts, typos, and other "housekeeping" changes that basically correct the documentation to reflect the current configuration. Class II applies only if the configuration is not changed when the documentation is changed. Class II ECPs are usually handled by the in-plant government representative. Class II ECPs generally require only that the government concurs that the change is properly classified. Under an initiative by the Defense Contract Management Command (DCMC), contractors are increasingly delegated the authority to make ECP classification decisions.

Figure 10-1 shows the key attributes associated with ECPs. The preliminary ECP, mentioned in Figure 10-1, is a simplified version of a formal ECP that explains the proposed ECP, and establishes an approximate schedule and cost for the change. The expense of an ECP development is avoided if review of the Preliminary ECP indicates the change is not viable. The approach used for preliminary ECPs vary in their form and name. Both Preliminary ECPs and Advanced Change/Study Notices have been used to formalize this process, but forms tailored to specific programs have also been used.

Configuration Control Board (CCB)

A CCB is formed to review Class I ECPs for approval, and make a recommendation to approve or not approve the proposed change. The CCB chair, usually the program manager, makes the final decision. Members advise and recommend, but the authority for the decision rests with the chair. CCB membership should represent the eight primary functions with the addition of representation of the procurement office, program control (budget), and Configuration Control manager, who serves as the CCB secretariat.

The CCB process is shown in Figure 10-2. The process starts with the contractor. A request to the contractor for an ECP or Preliminary ECP is necessary to initiate a government identified configuration change. The secretariat's review process includes assuring appropriate government



Figure 10-2. Configuration Control Board

contractual and engineering review is done prior to receipt by the CCB.

CCB Management Philosophy

The CCB process is a configuration control process, but it is also a contractual control process. Decisions made by the CCB chair affects the contractual agreement and program baseline as well as the configuration baseline. Concerns over contractual policy, program schedule, and budget can easily come into conflict with concerns relating to configuration management, technical issues, and technical activity scheduling. The CCB technical membership and CCB secretariat is responsible to provide a clear view of the technical need and the impact of alternate solutions to these conflicts. The CCB secretariat is further responsible to see that the CCB is fully informed and prepared, including ensuring that:

• A government/contractor engineering working group has analyzed the ECP and supporting data, prepared comments for CCB consideration, and is available to support the CCB;

- All pertinent information is available for review;
- The ECP has been reviewed by appropriate functional activities; and
- Issues have been identified and addressed.

CCB Documentation

Once the CCB chair makes a decision concerning an ECP, the CCB issues a Configuration Control Board Directive that distributes the decision and identifies key information relating to the implementation of the change:

- Implementation plan (who does what when);
- Contracts affected (prime and secondary);
- Dates of incorporation into contracts;
- Documentation affected (drawings, specifications, technical manuals, etc.), associated cost, and schedule completion date; and

• Identification of any orders or directives needed to be drafted and issued.

Request for Deviation or Waiver

A deviation is a specific written authorization, granted prior to manufacture of an item, to depart from a performance or design requirement for a specific number of units or a specific period of time.

A waiver is a written authorization to accept a CI that departs from specified requirements, but is suitable for use "as is" or after repair.

Requests for deviation and waivers relate to a temporary baseline departure that can affect system design and/or performance. The baseline remains unchanged and the government makes a determination whether the alternative "non-conforming" configuration results in an acceptable substitute. Acceptable substitute usually implies that there will be no impact on support elements, systems affected can operate effectively, and no follow-up or correction is required. The Federal Acquisition Regulations (FAR) requires "consideration" on government contracts when the Government accepts a "non-conforming" unit.

The distinction between Request for Deviation and Request for a Waiver is that a deviation is used *before* final assembly of the affected unit, and a waiver is used *after* final assembly or acceptance testing of the affected unit.

Status Accounting

Configuration Status Accounting is the recording and reporting of the information that is needed to manage the configuration effectively, including:

- A listing of the approved configuration documentation,
- The status of proposed changes, waivers and deviations to the configuration identification,
- The implementation status of approved changes, and

• The configuration of all units, including those in the operational inventory.

Purpose of Configuration Status Accounting

Configuration Status Accounting provides information required for configuration management by:

- Collecting and recording data concerning:
 - Baseline configurations,
 - Proposed changes, and
 - Approved changes.
- Disseminating information concerning:
 - Approved configurations,
 - Status and impact of proposed changes,
 - Requirements, schedules, impact and status of approved changes, and
 - Current configurations of delivered items.

Audits

Configuration Audits are used to verify a system and its components' conformance to their configuration documentation. Audits are key milestones in the development of the system and do not stand alone. The next chapter will show how they fit in the overall process of assessing design maturity.

Functional Configuration Audits (FCA) and the System Verification Review (SVR) are performed in the Production Readiness and LRIP stage of the Production and Development Phase. FCA is used to verify that actual performance of the configuration item meets specification requirements. The SVR serves as system-level audit after FCAs have been conducted.

The Physical Configuration Audit (PCA) is normally held during Rate Production and Development stage as a formal examination of a production representative unit against the draft technical data package (product baseline documentation).

Most audits, whether FCA or PCA, are today approached as a series of "rolling" reviews in which items are progressively audited as they are produced such that the final FCA or PCA becomes significantly less oppressive and disruptive to the normal flow of program development.

10.3 INTERFACE MANAGEMENT

Interface Management consists of identifying the interfaces, establishing working groups to manage the interfaces, and the group's development of interface control documentation. Interface Management identifies, develops, and maintains the external and internal interfaces necessary for system operation. It supports the configuration management effort by ensuring that configuration decisions are made with full understanding of their impact outside of the area of the change.

Interface Identification

An interface is a functional, physical, electrical, electronic, mechanical, hydraulic, pneumatic, optical, software, or similar characteristic required to exist at a common boundary between two or more systems, products, or components. Normally, in a contractual relationship the procuring agency identifies external interfaces, sets requirements for integrated teams, and provides appropriate personnel for the teams. The contracted design agent or manufacturer manages internal interfaces; plans, organizes, and leads design integrated teams; maintains internal and external interface requirements; and controls interfaces to ensure accountability and timely dissemination of changes.

Interface Control Working Group (ICWG)

The ICWG is the traditional forum to establish official communications link between those responsible for the design of interfacing systems or components. Within the IPPD framework ICWGs can be integrated teams that establish linkage between interfacing design IPTs, or could be integrated into a system-level engineering working group. Membership of ICWGs or comparable integrated teams should include membership from each contractor, significant vendors, and participating government agencies. The procuring program office (external and selected top-level interfaces) or prime contractor (internal interfaces) generally designates the chair.

Interface Control Documentation (ICD)

Interface Control Documentation includes Interface Control Drawings, Interface Requirements Specifications, and other documentation that depicts physical and functional interfaces of related or co-functioning systems or components. ICD is the product of ICWGs or comparable integrated teams, and their purpose is to establish and maintain compatibility between interfacing systems or components.

Open Systems Interface Standards

To minimize the impact of unique interface designs, improve interoperability, maximize the use of commercial components, and improve the capacity for future upgrade, an open-systems approach should be a significant part of interface control planning. The open-systems approach involves selecting industry-recognized specifications and standards to define system internal and external interfaces. An open system is characterized by:

- Increased use of functional partitioning and modular design to enhance flexibility of component choices without impact on interfaces,
- Use of well-defined, widely used, non-proprietary interfaces or protocols based on standards developed or adopted by industry recognized standards institutions or professional societies, and
- Explicit provision for expansion or upgrading through the incorporation of additional or higher performance elements with minimal impact on the system.

DoD mandatory guidance for information technology standards is in the Joint Technical Architecture.

10.4 DATA MANAGEMENT

Data management documents and maintains the database reflecting system life cycle decisions, methods, feedback, metrics, and configuration control. It directly supports the configuration status accounting process. Data Management governs and controls the selection, generation, preparation, acquisition, and use of data imposed on contractors.

Data Required By Contract

Data is defined as recorded information, regardless of form or characteristic, and includes all the administrative, management, financial, scientific, engineering, and logistics information and documentation required for delivery from the contractor. Contractually required data is classified as one of three types:

- Type I: Technical data
- Type II: Non-technical data
- Type III: One-time use data (technical or non-technical)

Data is acquired for two basic purposes:

- Information feedback from the contractor for program management control, and
- Decision making information needed to manage, operate, and support the system (e.g., specifications, technical manuals, engineering drawings, etc.).

Data analysis and management is expensive and time consuming. Present DoD philosophy requires that the contractor manage and maintain significant portions of the technical data, including the Technical Data Package (TDP). Note that this does *not* mean the government isn't paying for its development or shouldn't receive a copy for postdelivery use. Minimize the TDP cost by requesting the contractor's format (for example, accepting the same drawings they use for production), and asking only for details on items developed with government funds.

Data Call for Government Contracts

As part of the development of an Invitation for Bid or Request for Proposals, the program office issues a letter that describes the planned procurement and asks integrated team leaders and effected functional managers to identify and justify their data requirements for that contract. A description of each data item needed is then developed by the affected teams or functional offices, and reviewed by the program office. Data Item Descriptions, located in the Acquisition Management Systems Data List (AMSDL) (see Chapter 8) can be used for guidance in developing these descriptions.

Concurrent with the DoD policy on specifications and standards, there is a trend to avoid use of standard Data Item Descriptions on contracts, and specify the data item with a unique tailored data description referenced in the Contract Data Requirements List.

10.5 SUMMARY POINTS

- Configuration management is essential to control the system design throughout the life cycle.
- Use of integrated teams in an IPPD environment is necessary for disciplined configuration management of complex systems.
- Technical data management is essential to trace decisions and changes and to document designs, processes and procedures.
- Interface management is essential to ensure that system elements are compatible in terms of form, fit, and function.
- Three configuration baselines are managed:
 - Functional (System level)
 - Allocated (Design To)
 - Product (Build To/As Built)

Configuration management is a shared responsibility between the government and the contractor. Contract manager (CM) key elements are Identification, Control, Status Accounting, and Audits.

- 1. True or False. Systems engineering is an interdisciplinary engineering management process that evolves and verifies an integrated, life-cycle balanced set of system solutions that satisfy customer needs.
 - C True
 - False
- 2. What process is a top-down comprehensive, iterative and recursive problem-solving process, applied sequentially through all stages of development?
 - $^{igodoldsymbol{ iny}}$ The systems engineering process
 - O Design process
 - O Development process
 - O Discipline of this processes
- 3. Which of the following functions includes the activities necessary to evaluate progress and effectiveness of evolving system products and processes, and to measure specification compliance?
 - C Support
 - Operation
 - Verification
 - O Development
- 4. Managing the development and fielding of systems requires what basic activities?
 - Technical management
 - Business management
 - Contract management
 - All of the above

5. What milestone signifies the start of the system development and demonstration phase?

- O Milestone A
- O Milestone B
- O Milestone C
- O Milestone D

6. The item performance specifications and supporting documentation, when finalized, together form the _____ for the system.

- allocated baseline
- O product baseline
- functional baseline
- Configuration baseline

7. Of all the phases which is the longest and most expensive?

- Production and Deployment
- ^O System Development and Demonstration
- Sustainment and Disposal
- Concept and Technology Development

8. What is the first step of the system's engineering process?

- Functional Analysis
- O Design Synthesis
- O Verification
- Requirements Analysis

9. Which of the following is not an attribute of a good requirement?

- O Verifiable
- Expressed in terms of a solution
- O Unambiguous
- C Achievable
10. True or False. An initial statement of a need is seldom defined clearly.

- O True
- False
- 11. Functional Analysis and Allocation facilitates what from requirements to the solution descriptions that are the outcome of Design Synthesis?
 - Traceability
 - Physical description
 - O Unconstrained attributes
- 12. Functional Analysis and Allocation consists of _____ of higher-level functions to lower-levels and then allocation of requirements to those functions.
 - Verification
 - O Design Synthesis
 - O Decomposition
 - C Expansion

13. What is the mapping between the functional and physical architectures?

- Decomposition
- O Design loop
- O Modeling
- Work breakdown structures (WBS)

14. True or False. Verification represents the intersection of systems engineering and test and evaluation.

- O True
- False

15. Every system will have a set of _____ that are the performance characteristics that must be achieved by the design solution.

- Key Performance Parameters (KPPs)
- O Mandatory attributes
- Critical solutions
- Technical Performance Measurements (TPM)

16. System engineering outputs include which of the following?

- System Item Architectures
- Configuration Item Architectures
- O Specifications
- C Baselines
- All the above

17. What verb denotes a mandatory requirement? (Ref. Fig 8-5 Rules of Thumb for Specification Preparation)

- May
- O Would
- C Shall
- O Will
- 18. What is used to define the total system, to display it as a product-oriented family-tree composed of hardware, software, services, data, and facilities, and to relate these elements to each other and to the end product?
 - System Engineering Process
 - System Architecture
 - O Work Breakdown Structure
 - C Technical Data Package

19. True or False. A prime use of the WBS is the design and tracking of work.

- O True
- False

20. Configuration management comprises of four interrelated efforts, what are they?

- Identification, Control, Status Accounting, and Audits
- Planning, Control, Item Accounting, and Reviews
- Technical Reviews, Material Review, Status Accounting, and Audits
- Identification, Design, Change Request, and Audits