4

THE SYSTEM DEVELOPMENT PROCESS

4.1 SYSTEMS ENGINEERING THROUGH THE SYSTEM LIFE CYCLE

As was described in Chapter 1, modern engineered systems come into being in response to societal needs or because of new opportunities offered by advancing technology, or both. The evolution of a particular new system from the time when a need for it is recognized and a feasible technical approach is identified, through its development and introduction into operational use, is a complex effort, which will be referred to as the *system development process*. This chapter is devoted to describing the basic system development process and how systems engineering is applied at each step of this process.

A typical major system development exhibits the following characteristics:

- It is a complex effort.
- It meets an important user need.
- It usually requires several years to complete.
- It is made up of many interrelated tasks.

Systems Engineering Principles and Practice, Second Edition. Alexander Kossiakoff, William N. Sweet, Samuel J. Seymour, and Steven M. Biemer

- It involves several different disciplines.
- It is usually performed by several organizations.
- It has a specific schedule and budget.

The development and introduction into the use of a complex system inherently requires increasingly large commitments of resources as it progresses from concept through engineering, production, and operational use. Further, the introduction of new technology inevitably involves risks, which must be identified and resolved as early as possible. These factors require that the system development be conducted in a step-by-step manner, in which the success of each step is demonstrated, and the basis for the next one validated, before a decision is made to proceed to the next step.

4.2 SYSTEM LIFE CYCLE

The term "system life cycle" is commonly used to refer to the stepwise evolution of a new system from concept through development and on to production, operation, and ultimate disposal. As the type of work evolves from analysis in the early conceptual phases to engineering development and testing, to production and operational use, the role of systems engineering changes accordingly. As noted previously, the organization of this book is designed to follow the structure of the system life cycle, so as to more clearly relate systems engineering functions to their roles in specific periods during the life of the system. This chapter presents an overview of the system development process to create a context for the more detailed discussion of each step in the later chapters.

Development of a Systems Engineering Life Cycle Model for This Book

System life cycle models have evolved significantly over the past two decades. Furthermore, the number of models has grown as additional unique and custom applications were explored. Additionally, software engineering has spawned a significant number of development models that have been adopted by the systems community. The end result is that there is no single life cycle model that (1) is accepted worldwide and (2) fits every possible situation. Various standards organizations, government agencies, and engineering communities have published their particular models or frameworks that can be used to construct a model. Therefore, adopting one model to serve as an appropriate framework for this book was simply not prudent.

Fortunately, all life cycle models subdivide the system life into a set of basic steps that separate major decision milestones. Therefore, the derivation of a life cycle model to serve as an appropriate framework for this book had to meet two primary objectives. First, the steps in the life cycle had to correspond to the progressive transitions in the principal systems engineering activities. Second, these steps had to be capable of being mapped into the principal life cycle models in use by the systems engineering community. The derived model will be referred to as the "systems engineering life cycle,"

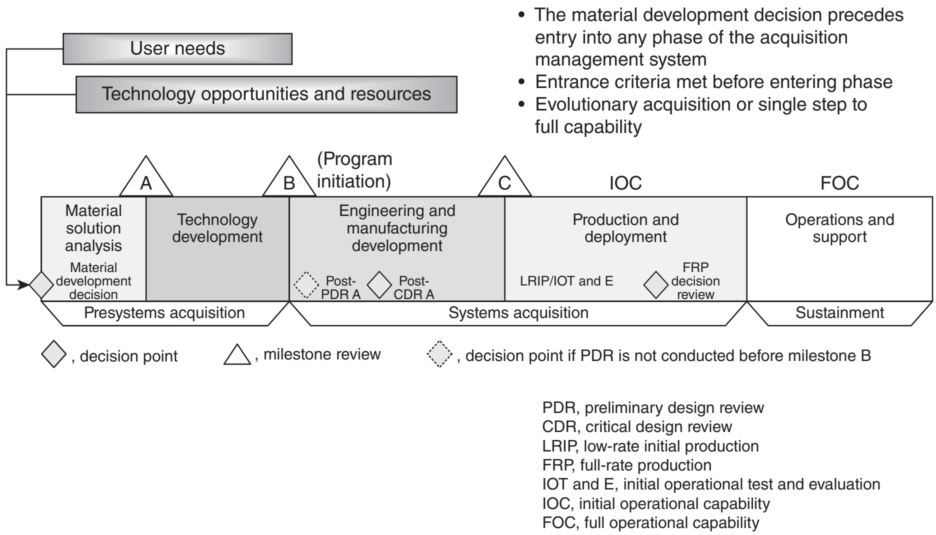

Figure 4.1. DoD system life cycle model.

and will be based on three different sources: the Department of Defense (DoD) Acquisition Management model (DoD 5000.2), the International model ISO/IEC 15288, and the National Society of Professional Engineers (NSPE) model.

DoD Acquisition Management Model. In the second half of the twentieth century, the United States was in the forefront of developing large-scale complex military systems such as warships, airplanes, tanks, and command and control systems. To manage the risks in the application of advanced technology, and to minimize costly technical or management failures, the DoD has evolved comprehensive system acquisition guidelines, which are contained in the DoD 5000 series of directives. The fall 2008 version of the DoD life cycle model, which reflects the acquisition guidelines, is displayed in Figure 4.1. It consists of five phases: material solution analysis, technology development, engineering and manufacturing development, production and deployment, and operations and support. The two activities of user need determination and technology opportunities and resources are considered to be part of the process but are not included in the formal portion of the acquisition cycle.

The DoD model is tailored toward managing large, complex system development efforts where reviews and decisions are needed at key events throughout the life cycle. The major reviews are referred to as milestones and are given letter designations: A, B, and C. Each of the three major milestones is defined with respect to entry and exit conditions. For example, at milestone A, a requirements document needs to be approved by a military oversight committee before a program will be allowed to transition to the next phase. In addition to milestones, the process contains four additional decision points: material development decision (MDD), preliminary design review (PDR), critical design review (CDR) and full-rate production (FRP) decision review. Therefore, DoD management is able to review and decide on the future of the program at up to seven major points within the life cycle.

International ISO/IEC 15288 Model. In 2002, the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) issued the result of several years of effort: a systems engineering standard designated ISO/IEC 15288, *Systems Engineering—System Life Cycle Processes*. The basic model is divided into six stages and 25 primary processes. The processes are intended to represent a menu of activities that may need to be accomplished within the basic stages. The ISO standard purposely does not align the stages and processes. The six basic stages are concept, development, production, utilization, support, and retirement.

Professional Engineering Model. The NSPE model is tailored to the development of commercial systems. This model is mainly directed to the development of new products, usually resulting from technological advances ("technology driven"). Thus, the NSPE model provides a useful alternative view to the DoD model of how a typical system life cycle may be divided into phases. The NSPE life cycle is partitioned into six stages: conceptual, technical feasibility, development, commercial validation and production preparation, full-scale production, and product support.

Systems Engineering Life Cycle Model. In structuring a life cycle model that corresponded to significant transitions in systems engineering activities throughout the system's active life, it was found most desirable to subdivide the life cycle into three broad stages and to partition these into eight distinct phases. This structure is shown in Figure 4.2 and will be discussed below. The names of these subdivisions were chosen

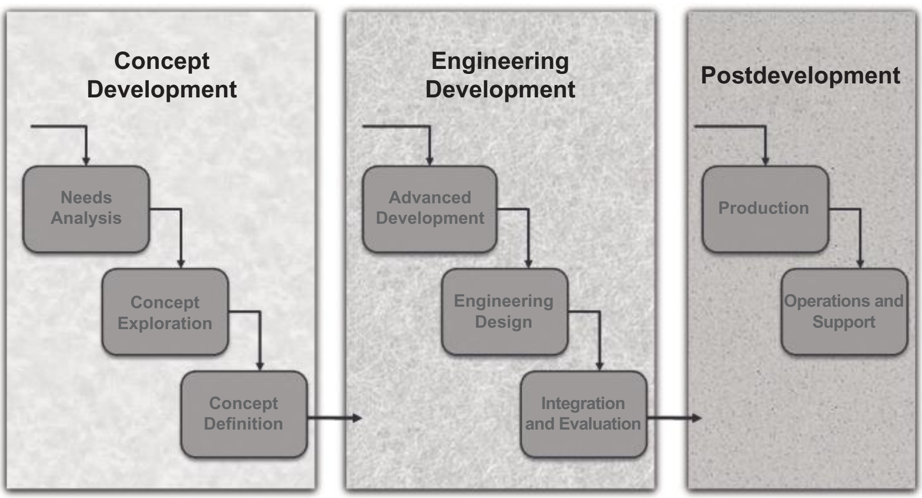

Figure 4.2. System life cycle model.

to reflect the primary activities occurring in each part of the process. Inevitably, some of these names are the same or similar to the names of corresponding parts of one or more of the existing life cycles.

Software Life Cycle Models. The system life cycle stages and their constituent phases represented by the above models apply to the majority of complex systems, including those containing significant software functionality at the component level. However, software-intensive systems, in which software performs virtually all the functionality, as in modern financial systems, airline reservation systems, the World Wide Web, and other information systems, generally follow life cycles similar in form but often involving iteration and prototyping. Chapter 11 describes the differences between software and hardware, discusses the activities involved in the principal stages of software system development, and contains a section dealing with examples of software system life cycles representing software-intensive systems. However, with that exception, the systems engineering life cycle model, as will be discussed in Chapters 5 through 15, provides a natural framework for describing the evolution of systems engineering activity throughout the active life of all engineered complex systems.

Systems Engineering Life Cycle Stages

As described above, and illustrated in Figure 4.2, the systems life cycle model consists of three stages, the first two encompassing the developmental part of the life cycle, and the third the postdevelopment period. These stages mark the more basic transitions in the system life cycle, as well as the changes in the type and scope of effort involved in systems engineering. In this book, these stages will be referred to as (1) The *concept development* stage, which is the initial stage of the formulation and definition of a system concept perceived to best satisfy a valid need; (2) the *engineering development* stage, which covers the translation of the system concept into a validated physical system design meeting the operational, cost, and schedule requirements; and (3) the *postdevelopment* stage, which includes the production, deployment, operation, and support of the system throughout its useful life. The names for the individual stages are intended to correspond generally to the principal type of activity characteristic of these stages.

The concept development stage, as the name implies, embodies the analysis and planning that is necessary to establish the need for a new system, the feasibility of its realization, and the specific system architecture perceived to best satisfy the user needs. Systems engineering plays the lead role in translating the operational needs into a technically and economically feasible system concept. Maier and Rechtin (2009) call this process "systems architecting," using the analogy of the building architect translating a client's needs into plans and specifications that a builder can bid on and build from. The level of effort during this stage is generally much smaller than in subsequent stages. This stage corresponds to the DoD activities of material solution analysis and technology development.

The principal objectives of the concept development stage are

1. to establish that there is a valid need (and market) for a new system that is technically and economically feasible;
2. to explore potential system concepts and formulate and validate a set of system performance requirements;
3. to select the most attractive system concept, define its functional characteristics, and develop a detailed plan for the subsequent stages of engineering, production, and operational deployment of the system; and
4. to develop any new technology called for by the selected system concept and to validate its capability to meet requirements.

The engineering development stage corresponds to the process of engineering the system to perform the functions specified in the system concept, in a physical embodiment that can be produced economically and maintained and operated successfully in its operational environment. Systems engineering is primarily concerned with guiding the engineering development and design, defining and managing interfaces, developing test plans, and determining how discrepancies in system performance uncovered during test and evaluation (T&E) should best be rectified. The main bulk of the engineering effort is carried out during this stage. The engineering development stage corresponds to the DoD activities of engineering and manufacturing development and is a part of production and deployment.

The principal objectives of the engineering development stage are

1. to perform the engineering development of a prototype system satisfying the requirements of performance, reliability, maintainability, and safety; and
2. to engineer the system for economical production and use and to demonstrate its operational suitability.

The postdevelopment stage consists of activities beyond the system development period but still requires significant support from systems engineering, especially when unanticipated problems requiring urgent resolution are encountered. Also, continuing advances in technology often require in-service system upgrading, which may be just as dependent on systems engineering as the concept and engineering development stages. This stage corresponds to a part of the DoD production and deployment phase and all of the operations and support phase.

The postdevelopment stage of a new system begins after the system successfully undergoes its operational T&E, sometimes referred to as *acceptance testing*, and is released for production and subsequent operational use. While the basic development has been completed, systems engineering continues to play an important supporting role in this effort.

The relations among the principal stages in the system life cycle are illustrated in the form of a flowchart in Figure 4.3. The figure shows the principal inputs and outputs of each of the stages. The legends above the blocks relate to the flow of information

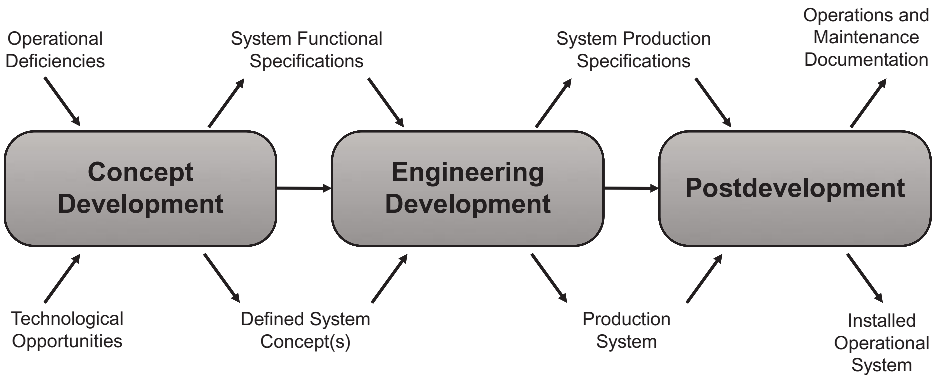

Figure 4.3. Principal stages in a system life cycle.

in the form of requirements, specifications, and documentation, beginning with operational needs. The inputs and outputs below the blocks represent the stepwise evolution of the design representations of an engineered system from the concept to the operational system. It is seen that both the documentation and design representations become increasingly complete and specific as the life cycle progresses. The later section entitled "System Materialization" is devoted to a discussion of the factors involved in this process.

Example: Development Stages of a New Commercial Aircraft. To illustrate the application of this life cycle model, consider the evolution of a new passenger aircraft. The concept development stage would include the recognition of a market for a new aircraft, the exploration of possible configurations, such as number, size, and location of engines, body dimensions, wing platform, and so on, leading to the selection of the optimum configuration from the standpoint of production cost, overall efficiency, passenger comfort, and other operational objectives. The above decisions would be based largely on analyses, simulations, and functional designs, which collectively would constitute justifications for selecting the chosen approach.

The engineering development stage of the aircraft life cycle begins with the acceptance of the proposed system concept and a decision by the aircraft company to proceed with its engineering. The engineering effort would be directed to validating the use of any unproven technology, implementing the system functional design into hardware and software components, and demonstrating that the engineered system meets the user needs. This would involve building prototype components, integrating them into an operating system and evaluating it in a realistic operational environment. The postdevelopment stage includes the acquisition of production tooling and test equipment, production of the new aircraft, customizing it to fit requirements of different customers, supporting regular operations, fixing any faults discovered during use, and periodically overhauling or replacing engines, landing gear, and other highly stressed components. Systems engineering plays a limited but vital supporting and problem-solving role during this stage.

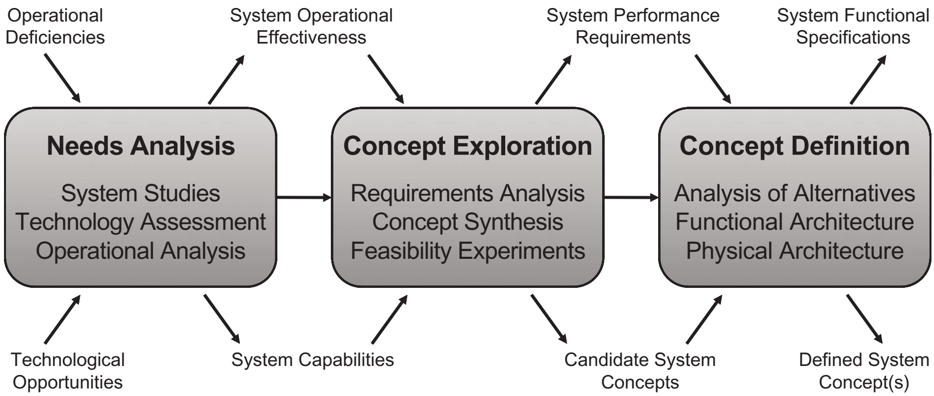

Figure 4.4. Concept development phases of a system life cycle.

Concept Development Phases

While the three stages described above constitute the dominant subdivisions of the system life cycle, each of these stages contains recognizable subdivisions with characteristically different objectives and activities. In the case of large programs, formal decision points also mark most of these subdivisions, similar to those marking the transition between stages. Furthermore, the roles of systems engineering tend to differ significantly among these intermediate subdivisions. Hence, to understand how the evolution of the system life cycle relates to the systems engineering process, it is useful to develop a model of its structure down to this second level of subdivision.

The concept development stage of the systems engineering life cycle encompasses three phases: *needs analysis*, *concept exploration*, and *concept definition*. Figure 4.4 shows these phases, their principal activities and inputs and outputs in a format analogous to Figure 4.3.

Needs Analysis Phase. The needs analysis phase defines the need for a new system. It addresses the questions "Is there a valid need for a new system?" and "Is there a practical approach to satisfying such a need?" These questions require a critical examination of the degree to which current and perceived future needs cannot be satisfied by a physical or operational modification of available means, as well as whether or not available technology is likely to support the increased capability desired. In many cases, the beginning of the life of a new system evolves from a continuing analysis of operational needs, or an innovative product development, without a sharply identified beginning.

The output of this phase is a description of the capabilities and operational effectiveness needed in the new system. In many ways, this description is the first iteration of the system itself, albeit a very basic conceptual model of the system. The reader should take note of how the "system" evolves from this very beginning phase throughout its life cycle. Although we would not yet call this description a set of requirements, they certainly are the forerunner of what will be defined as official requirements. Some communities refer to this early description as an initial capability description.

Several classes of tools and practices exist to support the development of the system capabilities and effectiveness description. Most fall into two categories of mathematics, known as operational analysis and operations research. However, technology assessments and experimentation are an integral part of this phase and will be used in conjunction with mathematical techniques.

Concept Exploration Phase. This phase examines potential system concepts in answering the questions "What performance is required of the new system to meet the perceived need?" and "Is there at least one feasible approach to achieving such performance at an affordable cost?" Positive answers to these questions set a valid and achievable goal for a new system project prior to expending a major effort on its development.

The output of this phase includes our first "official" set of requirements, typically known as system performance requirements. What we mean by official is that a contractor or agency can be measured against this set of required capabilities and performance. In addition to an initial set of requirements, this phase produces a set of candidate system concepts. Note the plural—more than one alternative is important to explore and understand the range of possibilities in satisfying the need.

A variety of tools and techniques are available in this phase and range from process methods (e.g., requirements analysis) to mathematically based (e.g., decision support methods) to expert judgment (e.g., brainstorming). Initially, the number of concepts can be quite large from some of these techniques; however, the set quickly reduces to a manageable set of alternatives. It is important to understand and "prove" the feasibility of the final set of concepts that will become the input of the next phase.

Concept Definition Phase. The concept definition phase selects the preferred concept. It answers the question "What are the key characteristics of a system concept that would achieve the most beneficial balance between capability, operational life, and cost?" To answer this question, a number of alternative concepts must be considered, and their relative performance, operational utility, development risk, and cost must be compared. Given a satisfactory answer to this question, a decision to commit major resources to the development of the new system can be made.

The output is really two perspectives on the same system: a set of functional specifications that describe what the system must do, and how well, and a selected system concept. The latter can be in two forms. If the complexity of the system is rather low, a simple concept description is sufficient to communicate the overall design strategy for the development effort to come. However, if the complexity is high, a simple concept description is insufficient and a more comprehensive system architecture is needed to communicate the various perspectives of the system. Regardless of the depth of description, the concept needs to be described in several ways, primarily from a functional perspective and from a physical perspective. Further perspectives may very well be needed if complexity is particularly high.

The tools and techniques available fall into two categories: analysis of alternatives (a particular method pioneered by the DoD, but fully part of operations research), and systems architecting (pioneered by Ebbert Rechtin in the early 1990s).

As noted previously, in commercial projects (NSPE model), the first two phases are often considered as a single preproject effort. This is sometimes referred to as a "feasibility study" and its results constitute a basis for making a decision as to whether or not to invest in a concept definition effort. In the defense acquisition life cycle, the second and third phases are combined, but the part corresponding to the second phase is performed by the government, resulting in a set of system performance requirements, while that corresponding to the third can be conducted by a government–contractor team or performed by several contractors competing to meet the above requirements.

In any case, before reaching the engineering development stage, only a fractional investment has usually been made in the development of a particular system, although some years and considerable effort may have been spent in developing a firm understanding of the operational environment and in exploring relevant technology at the subsystem level. The ensuing stages are where the bulk of the investment will be required.

Engineering Development Phases

Figure 4.5 shows the activities, inputs, and outputs of the constituent phases of the engineering development stage of the system life cycle in the same format as used in Figure 4.3. These are referred to as *advanced development*, *engineering design*, and *integration and evaluation*.

Advanced Development Phase. The success of the engineering development stage of a system project is critically dependent on the soundness of the foundation laid

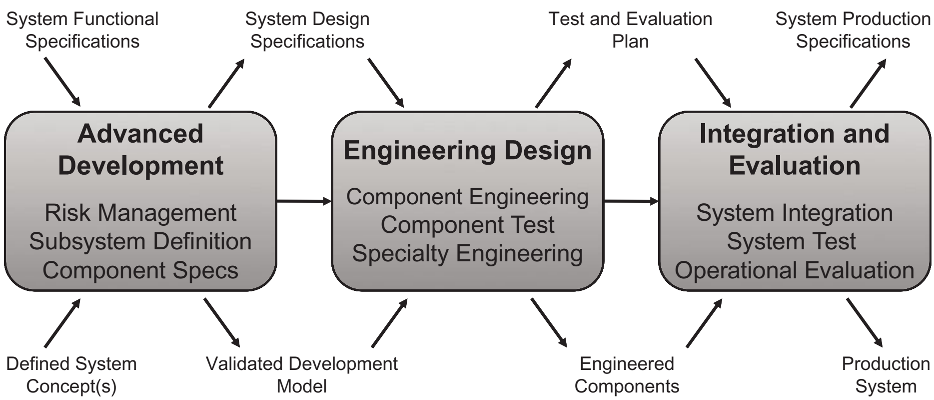

Figure 4.5. Engineering development phases in a system life cycle.

during the concept development stage. However, since the conceptual effort is largely analytical in nature and carried out with limited resources, significant unknowns invariably remain that are yet to be fully defined and resolved. It is essential that these "unknown unknowns" be exposed and addressed early in the engineering stage. In particular, every effort must be made to minimize the number of as yet undisclosed problems prior to translating the functional design and associated system requirements into engineering specifications for the individual system hardware and software elements.

The advanced development phase has two primary purposes: (1) the identification and reduction of development risks and (2) the development of system design specifications. The advanced development phase is especially important when the system concept involves advanced technology not previously used in a similar application, or where the required performance stresses the system components beyond proven limits. It is devoted to designing and demonstrating the undeveloped parts of the system, to proving the practicality of meeting their requirements, and to laying the basis for converting the functional system requirements into system specifications and component design requirements. Systems engineering is central to the decisions of what needs to be validated and how, and to the interpretation of the results.

This phase corresponds to the defense acquisition phase called "engineering and manufacturing development," once referred to as "demonstration and validation." When the risks of using unproven technology are large, this phase is often contracted separately, with contracts for the remaining engineering phase contingent on its success.

Matching the purpose of this phase, the two primary outputs are the design specifications and a validated development model. The specifications are a refinement and evolution of the earlier function specifications. The development model is the final outcome of a very comprehensive risk management task—where those unknowns mentioned above have been identified and resolved. This is what we mean when we use the adjective "validated." The systems engineer needs to be convinced that this system can be designed and manufactured before transitioning from this phase. Therefore, all risks at this phase must be rated as manageable before proceeding.

Modern risk management tools and techniques are essential to reduce and ultimately to mitigate risks inherent in the program. As these risks are managed, the level of definition continues to migrate down, from the system to the subsystem. Furthermore, a set of specifications for the next level of decomposition, at the component level, occurs. In all of these cases, both experimental models and simulations are often employed at this stage to validate component and subsystem design concepts at minimum cost.

Engineering Design Phase. The detailed engineering design of the system is performed during this phase. Because of the scale of this effort, it is usually punctuated by formal design reviews. An important function of these reviews is to provide an opportunity for the customer or user to obtain an early view of the product, to monitor its cost and schedule, and to provide valuable feedback to the system developer.

While issues of reliability, producibility, maintainability, and other "ilities" have been considered in previous phases, they are of paramount importance in the engineering design phase. These types of issues are typically known as "specialty engineering." Since the product consists of a set of components capable of being integrated and tested as a system, the systems engineer is responsible for ensuring that the engineering design of the individual components faithfully implements the functional and compatibility requirements, and for managing the engineering change process to maintain interface and configuration control.

The tasks of this phase deals with converting the component specifications into a set of component designs. Of course, testing these components is essential to occur immediately after design, or in some cases, concurrently with design. One additional task is performed during this phase: the refinement of the system T&E plan. We use the term *refinement* to distinguish between the initiation and continuation. The T&E plan is initially developed much earlier in the life cycle. At this phase, the T&E plan is largely finished, using the knowledge gained from the previous phases.

The two primary outputs are the T&E plan and an engineered prototype. The prototype can take many forms and should not be thought of in the same way as we think of a software prototype. This phase may produce a prototype that is virtual, physical, or a hybrid, depending on the program. For example, if the system is an ocean-going cargo vessel, the prototype at this stage may be a hybrid of virtual and physical mock-ups. A full-scale prototype of a cargo ship may not be possible or prudent at this phase. On the other hand, if the system is a washing machine, a full-scale prototype may be totally appropriate.

Modern computer-aided design tools are available as design engineers perform their trade. System models and simulations are also updated as designs are finalized and tested.

Integration and Evaluation Phase. The process of integrating the engineered components of a complex system into a functioning whole, and evaluating the system's operation in a realistic environment, is nominally part of the engineering design process because there is no formal break in the development program at this point. However, there is a basic difference between the role and responsibility of systems engineering during the engineering design of the system elements and that during the integration and evaluation process. Since this book is focused on the functions of systems engineering, the system integration and evaluation process is treated as a separate phase in the system life cycle.

It is important to realize that the first time a new system can be assembled and evaluated as an operating unit is after all its components are fully engineered and built. It is at this stage that all the component interfaces must fit and component interactions must be compatible with the functional requirements. While there may have been prior tests at the subsystem level or at the level of a development prototype, the integrity of the total design cannot be validated prior to this point.

It should also be noted that the system integration and evaluation process often requires the design and construction of complex facilities to closely simulate operational stimuli and constraints and to measure the system's responses. Some of these facilities may be adapted from developmental equipment, but the magnitude of the task should not be underestimated.

The outputs of this phase are twofold: (1) the specifications to guide the manufacturing of the system, typically called the system production specifications (sometimes referred to as the production baseline), and (2) the production system itself. The latter includes everything necessary to manufacture and assemble the system and may include a prototype system.

Modern integration techniques and T&E tools, methods, facilities, and principles are available to assist and enable the engineers in these tasks. Of course, before full-scale production can occur, the final production system needs to be verified and validated through an evaluation within the operational environment or a sufficient surrogate for the operational environment.

Postdevelopment Phases

Production Phase. The production phase is the first of the two phases comprising the postdevelopment stage, which are exactly parallel to the defense acquisition phases of “production and deployment” and “operations and support.”

No matter how effectively the system design has been engineered for production, problems inevitably arise during the production process. There are always unexpected disruptions beyond the control of project management, for example, a strike at a vendor’s plant, unanticipated tooling difficulties, bugs in critical software programs, or an unexpected failure in a factory integration test. Such situations threaten costly disruptions in the production schedule that require prompt and decisive remedial action. Systems engineers are often the only persons qualified to diagnose the source of the problem and to find an effective solution. Often a systems engineer can devise a “work-around” that solves the problem for a minimal cost. This means that an experienced cadre of systems engineers intimately familiar with the system design and operation needs to be available to support the production effort. Where specialty engineering assistance may be required, the systems engineers are often best qualified to decide who should be called in and when.

Operations and Support Phase. In the operations and support phase, there is an even more critical need for systems engineering support. The system operators and maintenance personnel are likely to be only partially trained in the finer details of system operation and upkeep. While specially trained field engineers generally provide support, they must be able to call on experienced systems engineers in case they encounter problems beyond their own experience.

Proper planning for the operational phase includes provision of a logistic support system and training programs for operators and maintenance personnel. This planning should have major participation from systems engineering. There are always unanticipated problems that arise after the system becomes operational that must be recognized and included in the logistic and training systems. Very often, the instrumentation required for training and maintenance is itself a major component of the system to be delivered.

Most complex systems have lifetimes of many years, during which they undergo a number of minor and major upgrades. These upgrades are driven by evolution in the system mission, as well as by advances in technology that offer opportunities to improve operation, reliability, or economy. Computer-based systems are especially subject to periodic upgrades, whose cumulative magnitude may well exceed the initial system development. While the magnitude of an individual system upgrade is a fraction of that required to develop a new system, it usually entails a great many complex decisions requiring the application of systems engineering. Such an enterprise can be extremely complex, especially in the conceptual stage of the upgrade effort. Anyone that has undergone a significant home alteration, such as the addition of one bedroom and bath, will appreciate the unexpected difficulty of deciding just how this can be accomplished in such a way as to retain the character of the original structure and yet realize the full benefits of the added portion, as well as be performed for an affordable price.

4.3 EVOLUTIONARY CHARACTERISTICS OF THE DEVELOPMENT PROCESS

The nature of the system development process can be better understood by considering certain characteristics that evolve during the life cycle. Four of these are described in the paragraphs below. The section The Predecessor System discusses the contributions of an existing system on the development of a new system that is to replace it. The section System Materialization describes a model of how a system evolves from concept to an engineered product. The section The Participants describes the composition of the system development team and how it changes during the life cycle. The section System Requirements and Specifications describes how the definition of the system evolves in terms of system requirements and specifications as the development progresses.

The Predecessor System

The process of engineering a new system may be described without regard to its resemblance to current systems meeting the same or similar needs. The entire concept and all of its elements are often represented as starting with a blank slate, a situation that is virtually never encountered in practice.

In the majority of cases, when new technology is used to achieve radical changes in such operations as transportation, banking, or armed combat, there exist predecessor systems. In a new system, the changes are typically confined to a few subsystems, while the existing overall system architecture and other subsystems remain substantially unchanged. Even the introduction of automation usually changes the mechanics but not the substance of the process. Thus, with the exception of such breakthroughs as the first generation of nuclear systems or of spacecraft, a new system development can expect to have a predecessor system that can serve as a point of departure.

A predecessor system will impact the development of a new system in three ways:

1. The deficiencies of the predecessor system are usually recognized, often being the driving force for the new development. This focuses attention on the most

important performance capabilities and features that must be provided by the new system.

2. If the deficiencies are not so serious as to make the current system worthless, its overall concept and functional architecture may constitute the best starting point for exploring alternatives.
3. To the extent that substantial portions of the current system perform their function satisfactorily and are not rendered obsolete by recent technology, great cost savings (and risk reduction) may be achieved by utilizing them with minimum change.

Given the above, the average system development will almost always be a hybrid, in that it will combine new and undemonstrated components and subsystems with previously engineered and fully proven ones. It is a particular responsibility of systems engineering to ensure that the decisions as to which predecessor elements to use, which to reengineer, which to replace by new ones, and how these are to be interfaced are made through careful weighing of performance, cost, schedule, and other essential criteria.

System Materialization

The steps in the development of a new system can be thought of as an orderly progressive "materialization" of the system from an abstract need to an assemblage of actual components cooperating to perform a set of complex functions to fulfill that need. To illustrate this process, Table 4.1 traces the growth of materialization throughout the phases of the project life cycle. The rows of the table represent the levels of system subdivision, from the system itself at the top to the part level at the bottom. The columns are successive phases of the system life cycle. The entries are the primary activities at each system level and phase, and their degree of materialization. The shaded areas indicate the focus of the principal effort in each phase.

It is seen that each successive phase defines (materializes) the next lower level of system subdivision until every part has been fully defined. Examining each row from left to right, say, at the component level, it is also seen that the process of definition starts with visualization (selecting the general type of system element), then proceeds to defining its functions (functional design, what it must do), and then to its implementation (detailed design, how it will do it).

The above progression holds true through the engineering design phase, where the components of the system are fully "materialized" as finished system building blocks. In the integration and evaluation phase, the materialization process takes place in a distinctly different way, namely, in terms of the materialization of an integrated and validated operational system from its individual building blocks. These differences are discussed further in Chapter 13.

It is important to note from Table 4.1 that while the detailed design of the system is not completed until near the end of its development, its general characteristics must be visualized very early in the process. This can be understood from the fact that the selection of the specific system concept requires a realistic estimate of the cost to

TABLE 4.1. Evolution of System Materialization through the System Life Cycle

	Phase					
	Concept development			*Engineering development*		
Level	Needs analysis	Concept exploration	Concept definition	Advanced development	Engineering design	Integration and evaluation
System	Define system capabilities and effectiveness	Identify, explore, and synthesize concepts	Define selected concept with specifications	Validate concept		Test and evaluate
Subsystem		Define requirements and ensure feasibility	Define functional and physical architecture	Validate subsystems		Integrate and test
Component			Allocate functions to components	Define specifications	Design and test	Integrate and test
Subcomponent	Visualize			Allocate functions to subcomponents	Design	
Part					Make or buy	

develop and produce it, which in turn requires a visualization of its general physical implementation as well as its functionality. In fact, it is essential to have at least a general vision of the physical embodiment of the system functions during even the earliest investigations of technical feasibility. It is of course true that these early visualizations of the system will differ in many respects from its final materialization, but not so far as to invalidate conclusions about its practicality.

The role of systems architecting fulfills this visualization requirement by providing visual perspectives into the system concept early in the life cycle. As a system project progresses through its life cycle, the products of the architecture are decomposed to ever-lower levels.

At any point in the cycle, the current state of system definition can be thought of as the current system model. Thus, during the concept development stage, the system model includes only the system functional model that is made up entirely of descriptive material, diagrams, tables of parameters, and so on, in combination with any simulations that are used to examine the relationships between system-level performance and specific features and capabilities of individual system elements. Then, during the engineering development stage, this model is augmented by the gradual addition of hardware and software designs for the individual subsystems and components, leading finally to a completed engineering model. The model is then further extended to a production model as the engineering design is transformed into producible hardware designs, detailed software definition, production tooling, and so on. At every stage of the process, the current system model necessarily includes models of all externally imposed interfaces as well as the internal system interfaces.

The Participants

A large project involves not only dozens or hundreds of people but also several different organizational entities. The ultimate user may or may not be an active participant in the project but plays a vital part in the system's origin and in its operational life. The two most common situations are when (1) the government serves as the system acquisition agent and user, with a commercial prime contractor supported by subcontractors as the system developer and producer, and (2) a commercial company serves as the acquisition manager, system developer, and producer. Other commercial companies or the general public may be the users. The principal participants in each phase of the project are also different. Therefore, one of the main functions of systems engineering is to provide the continuity between successive participating levels in the hierarchy and successive development phases and their participants through both formal documentation and informal communications.

A typical distribution of participants in an aerospace system development is shown in Figure 4.6. The height of the columns represents the relative number of engineering personnel involved. The entries are the predominant types of personnel in each phase. It is seen that, in general, participation varies from phase to phase, with systems engineering providing the main continuity.

The principal participants in the early phases are analysts and architects (system and operations/market). The concept definition phase is usually carried out by an

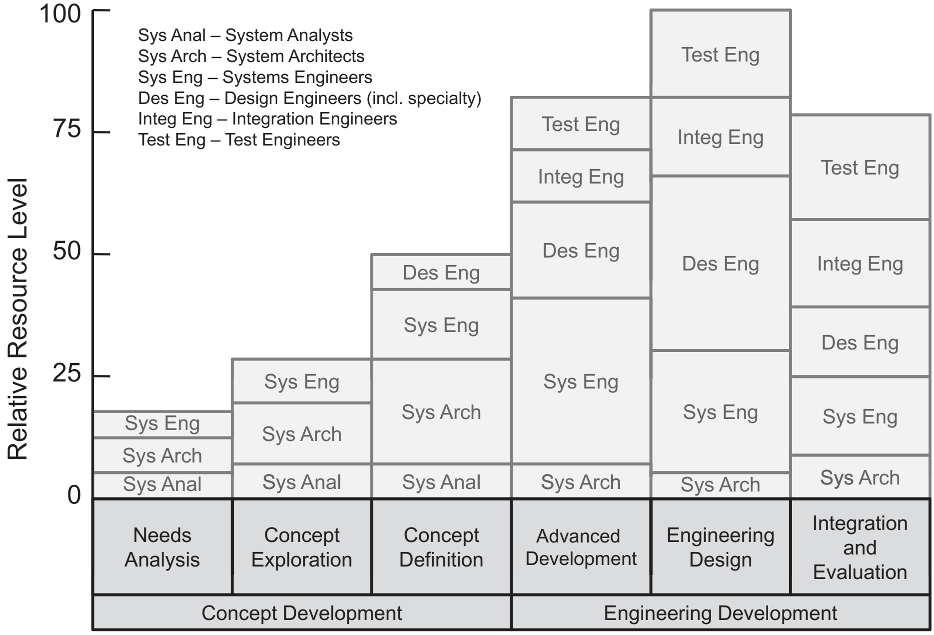

Figure 4.6. Principal participants in a typical aerospace system development.

expedited team effort, representing all elements necessary to select and document the most cost-effective system concept for meeting the stated requirements.

The advanced development phase usually marks the initial involvement of the system design team that will carry the project through the engineering stage and on into production. It is led by systems engineering, with support from the design and test engineers engaged in the development of components and subsystems requiring development.

The engineering design phase further augments the effort with a major contribution from specialty engineering (reliability, maintainability, etc.), as well as test and production engineering. For software, this phase involves designers, as well as coders, to the extent that prototyping is employed.

The integration and evaluation phase relies heavily on test engineering with guidance from systems engineering and support from design engineers and engineering specialists.

System Requirements and Specifications

Just as the system design gradually materializes during the successive steps of system development, so the successive forms of system requirements and specifications become more and more specific and detailed. These start with a set of operational requirements and end with a complete set of production specifications, operation, maintenance, and training manuals and all other information needed to replicate, operate, maintain, and repair the system. Thus, each phase can be thought of as producing a more detailed description of the system: what it does, how it works, and how it is built.

Since the above documents collectively determine both the course of the development effort and the form and capabilities of the system as finally delivered, oversight of their definition and preparation is a primary responsibility of systems engineering. This effort must, however, be closely coordinated with the associated design specialists and other involved organizations.

The evolution of system requirements and specifications is shown in the first row of Table 4.2 as a function of the phases in the system life cycle. It should be emphasized that each successive set of documents does not replace the versions defined during the previous phases but rather supplements them. This produces an accumulation rather than a succession of system requirements and other documents. These are “living documents,” which are periodically revised and updated.

The necessity for an aggregation of formal requirements and specifications developed during successive phases of the system development can be better understood by recalling the discussion of “Participants” and Figure 4.6. In particular, not only are there many different groups engaged in the development process, but many, if not most, of the key participants change from one phase to the next. This makes it essential that a complete and up-to-date description exists that defines what the system must do and also, to the extent previously defined, how it must do it.

The system description documents not only lay the basis for the next phase of system design but they also specify how the results of the effort are to be tested in order to validate compliance with the requirements. They provide the information base needed for devising both the production tools and the tools to be used for inspecting and testing the product of the forthcoming phase.

The representations of system characteristics also evolve during the development process, as indicated in the second row of Table 4.2. Most of these will be recognized as architecture views and conventional engineering design and software diagrams and models. Their purpose is to supplement textual descriptions of successive stages of system materialization by more readily understandable visual forms. This is especially important in defining interfaces and interactions among system elements designed by different organizations.

4.4 THE SYSTEMS ENGINEERING METHOD

In the preceding sections, the engineering of a complex system was seen to be divisible into a series of steps or phases. Beginning with the identification of an opportunity to achieve a major extension of an important operational capability by a feasible technological approach, each succeeding phase adds a further level of detailed definition (materialization) of the system, until a fully engineered model is achieved that proves to meet all essential operational requirements reliably and at an affordable cost. While many of the problems addressed in a given phase are peculiar to that state of system definition, the systems engineering principles that are employed, and the relations

TABLE 4.2. Evolution of System Representation

	Concept development			*Engineering development*		
	Needs analysis	Concept exploration	Concept definition	Advanced development	Engineering design	Integration and evaluation
Documents	System capabilities and effectiveness	System performance requirements	System functional requirements	System design specifications	Design documents	Test plans and evaluation reports
System models	Operational diagrams, mission simulations	System diagrams, high-level system simulations	Architecture products and views, simulations, mock-ups	Architecture products and views, detailed simulations, breadboards	Architecture drawings and schematics, engineered components, computer-aided design (CAD) products	Test setups, simulators, facilities, and test articles

among them, are fundamentally similar from one phase to the next. This fact, and its importance in understanding the system development process, has been generally recognized; the set of activities that tends to repeat from one phase to the next has been referred to in various publications on systems engineering as the "systems engineering process," or the "systems engineering approach," and is the subject of the sections below. In this book, this iterative set of activities will be referred to as the "systems engineering method."

The reason for selecting the word "method" in place of the more widely used "process" or "approach" is that it is more definitive and less ambiguous. The word method is more specific than process, having the connotation of an orderly and logical process. Furthermore, the term systems engineering process is sometimes used to apply to the total system development. Method is also more appropriate than approach, which connotes an attitude rather than a process. With all this said, the use of a more common terminology is perfectly acceptable.

Survey of Existing Systems Engineering Methods and Processes

The first organization to codify a formal systems engineering process was the U.S. DoD, captured in the military standard, MIL-STD-498. Although the process evolved through several iterations, the last formal standard to exist (before being discontinued) was MIL-STD-499B. This process is depicted in Figure 4.7 and contains four major activities: requirements analysis, functional analysis and allocation, synthesis, and systems analysis and control. The component tasks are presented within each activity.

While this military standard is no longer in force, it is still used as a guide by many organizations and is the foundation for understanding the basics of today's systems engineering processes.

Three relevant commercial standards describe a systems engineering process: IEEE-1220, the EIA-STD-632, and the ISO-IEC-IEEE-STD-15288. As these three processes are presented, notice that each commercial standard blends aspects of a systems engineering process with the life cycle model describe above. The order that we present these three methods is important—they are presented in order of the level of convergence with the life cycle model of system development. And in fact, the military standard discussed above could be placed first in the sequence. In other words, MIL-STD-499B is the most divergent from the life cycle model. In contrast, ISO-15288 could easily be thought of as a life cycle model for system development.

Figure 4.8 presents the IEEE-1220 process. The main control activity is located in the middle of the graph. The general flow of activities is then clockwise, starting from the bottom left, beginning with "process inputs" and ending with "process outputs." This process could also be thought of as an expansion of the military standard—the four basic activities are present, with a verification or validation step in between.

Figure 4.9 presents the EIA-632 process. Actually, the EIA-632 standard presents a collection of 13 processes that are linked together. One can easily recognize the iterative and circular nature of these linkages. Although the general flow is top–down, the processes are repeated multiple times throughout the system life cycle.

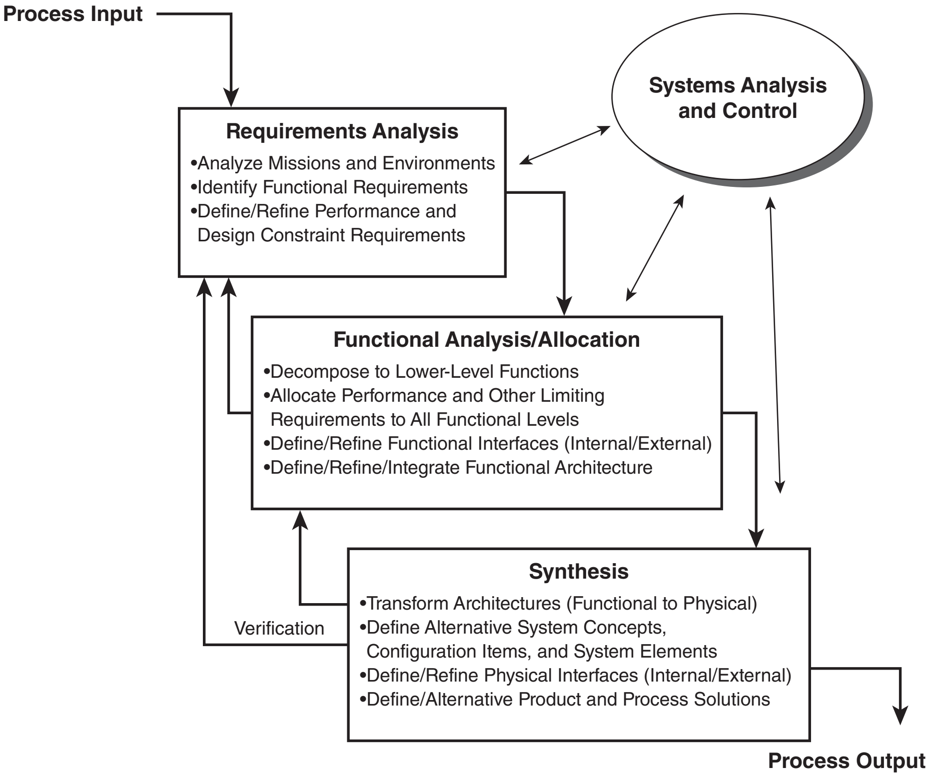

Figure 4.7. DoD MIL-STD499B.

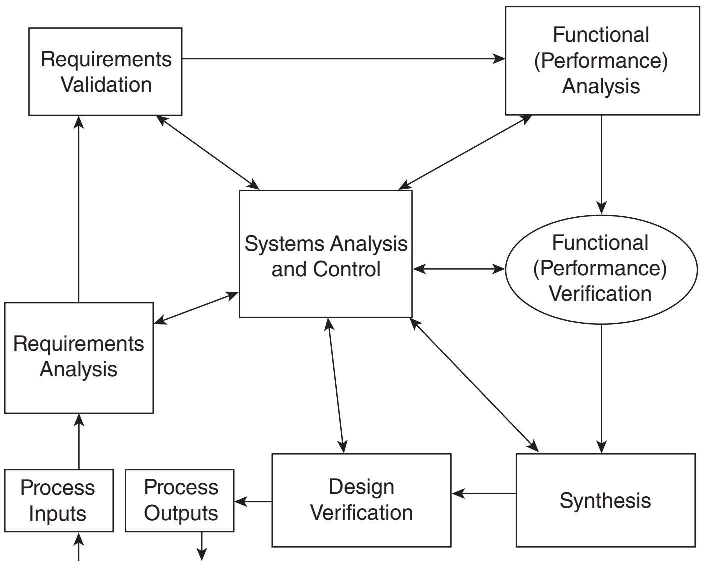

Figure 4.8. IEEE-1220 systems engineering process.

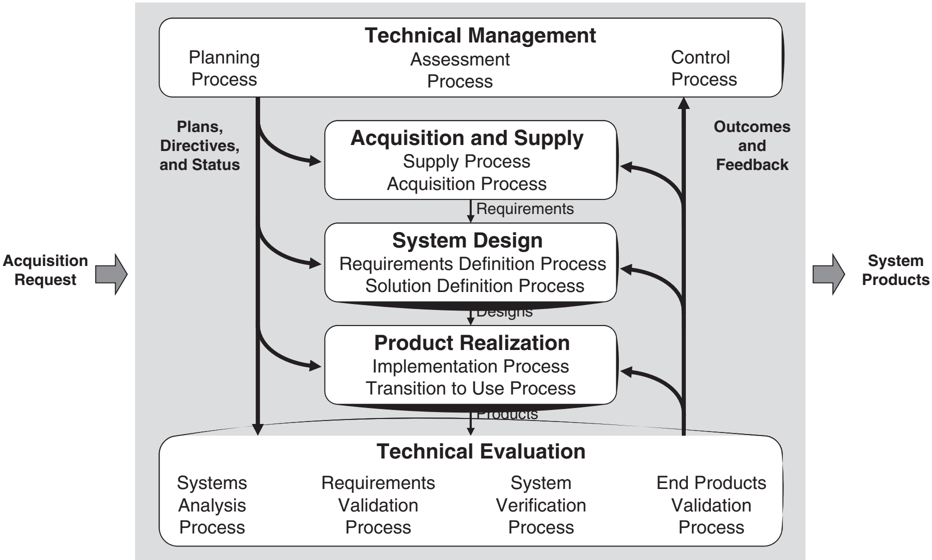

Figure 4.9. EIA-632 systems engineering process.

The 13 processes are further categorized into five sets: technical management, acquisition and supply, system design, product realization, and technical evaluation. The first and last process sets occur almost continuously throughout the system development life cycle. Planning, assessment, and control do not stop after the initial development phases, and systems analysis, requirements validation, system verification, and end-product validation commence well before a physical product is available. The three middle sets occur linearly, but with feedback and iterations.

Figure 4.10 presents the ISO-15288 process. This standard presents processes for both the system life cycle and systems engineering activities. In addition, the philosophy behind this standard is based on the systems engineer's and the program manager's ability to tailor the processes presented into a sequence of activities that is applicable to the program. Thus, no specific method is presented that sequences a subset of processes.

Our Systems Engineering Method

The *systems engineering method* can be thought of as the systematic application of the scientific method to the engineering of a complex system. It can be considered as consisting of four basic activities applied successively, as illustrated in Figure 4.11:

1. requirements analysis,
2. functional definition,
3. physical definition, and
4. design validation.

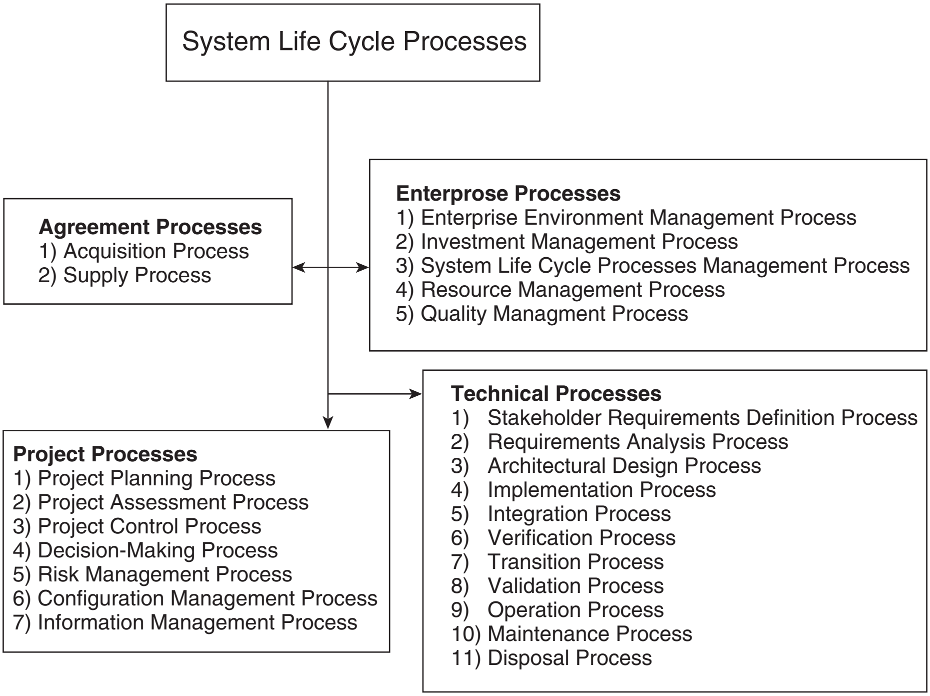

Figure 4.10. ISO-15288 Systems engineering process.

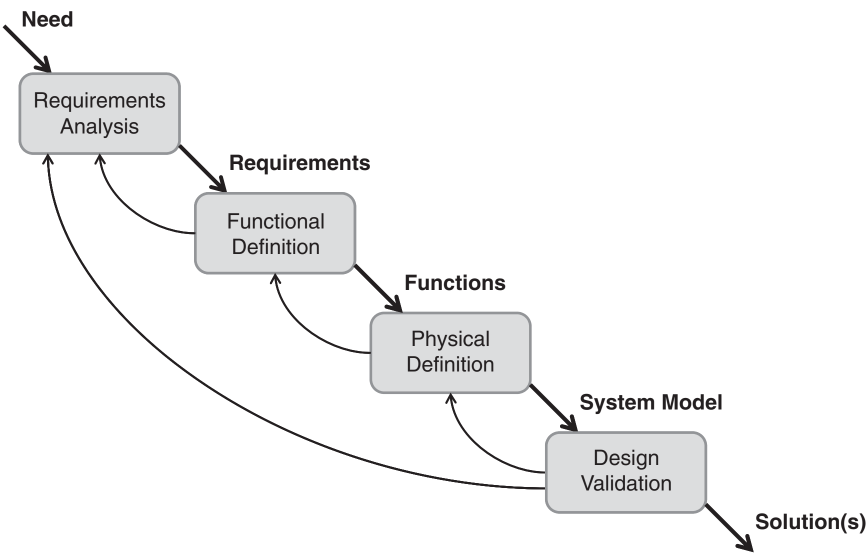

Figure 4.11. Systems engineering method top-level flow diagram.

These steps will vary in their specifics depending on the type of system and the phase of its development. However, there is enough similarity in their operating principles that it is useful to describe the typical activities of each step in the method. Such brief descriptions of the activities in the four steps are listed below.

Requirements Analysis (Problem Definition). Typical activities include

- assembling and organizing all input conditions, including requirements, plans, milestones, and models from the previous phase;
- identifying the "whys" of all requirements in terms of operational needs, constraints, environment, or other higher-level objectives;
- clarifying the requirements of what the system must do, how well it must do it, and what constraints it must fit; and
- correcting inadequacies and quantifying the requirements wherever possible.

Functional Definition (Functional Analysis and Allocation). Typical activities include

- translating requirements (why) into functions (actions and tasks) that the system must accomplish (what),
- partitioning (allocating) requirements into functional building blocks, and
- defining interactions among functional elements to lay a basis for their organization into a modular configuration.

Physical Definition (Synthesis, Physical Analysis, and Allocation). Typical activities include

- synthesizing a number of alternative system components representing a variety of design approaches to implementing the required functions, and having the most simple practicable interactions and interfaces among structural subdivisions;
- selecting a preferred approach by trading off a set of predefined and prioritized criteria (measures of effectiveness [MOE]) to obtain the best "balance" among performance, risk, cost, and schedule; and
- elaborating the design to the necessary level of detail.

Design Validation (Verification and Evaluation). Typical activities include

- designing models of the system environment (logical, mathematical, simulated, and physical) reflecting all significant aspects of the requirements and constraints;
- simulating or testing and analyzing system solution(s) against environmental models; and
- iterating as necessary to revise the system model or environmental models, or to revise system requirements if too stringent for a viable solution until the design and requirements are fully compatible.

The elements of the systems engineering method as described above are displayed in the form of a flow diagram in Figure 4.12, which is an expanded view of Figure

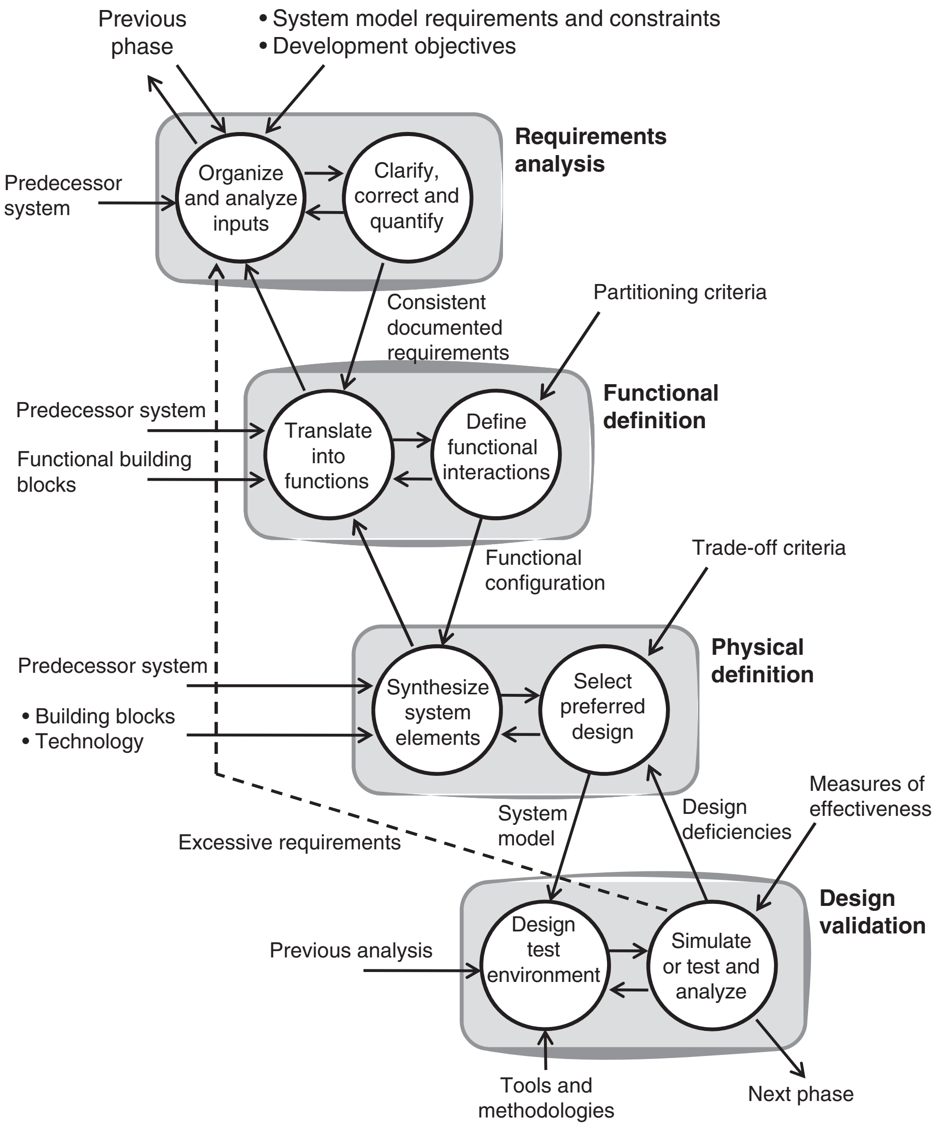

Figure 4.12. Systems engineering method flow diagram.

4.11. The rectangular blocks are seen to represent the above four basic steps in the method: requirements analysis, functional definition, physical definition, and design validation. At the top are shown inputs from the previous phase, which include requirements, constraints, and objectives. At the left of each block are shown external inputs, such as the predecessor system, system building blocks, and previous analyses. At the upper right of the top blocks and at the very bottom are inputs from systems engineering methodology.

The circles inside each block are simplified representations of key processes in that step of the method. The interfacing arrows represent information flow. It is seen that there are feedbacks throughout the process, iteration within the elements as well as to prior elements, and indeed all the way back to the requirements.

Each of the elements of the method is described more fully in the remainder of this section.

Requirements Analysis (Problem Definition)

In attempting to solve any problem, it is first necessary to understand exactly what is given, and to the extent that it appears to be incomplete, inconsistent, or unrealistic, to make appropriate amplifications or corrections. This is particularly essential in the system development process, where a basic characteristic of systems engineering is that everything is not necessarily what it seems and that important assumptions must be verified before they are accepted as being valid.

Thus, in a system development project, it is the responsibility of systems engineering to thoroughly analyze all requirements and specifications, first in order to understand them vis-à-vis the basic needs that the system is intended to satisfy, and then to identify and correct any ambiguities or inconsistencies in the definition of capabilities for the system or system element being addressed.

The specific activities of requirements analysis vary as the system development progresses, as the inputs from the previous phase evolve from operational needs and technological opportunities (see Fig. 4.3) to increasingly specific representations of requirements and system design. The role of systems engineering is essential throughout, but perhaps more so in the early phases, where an understanding of the operational environment and the availability and maturity of applicable technology are most critical. In later phases, environmental, interface, and other interelement requirements are the special province of systems engineering.

Organization and Interpretation. In a well-structured acquisition process, a new phase of the system life cycle begins with three main inputs, which are defined during or upon completion of the previous phase:

1. the system model, which identifies and describes all design choices made and validated in the preceding phases;
2. requirements (or specifications) that define the design, performance, and interface compatibility features of the system or system elements to be developed during the next phase; these requirements are derived from previously developed higher-level requirements, including any refinements and/or revisions introduced during the latest phase; and
3. specific progress to be achieved by each component of the engineering organization during the next phase, including the identification of all technical design data, hardware/software products, and associated test data to be provided; this information is usually presented in the form of a series of interdependent task statements.

Clarification, Correction, and Quantification. It is always difficult to express objectives in unambiguous and quantitative terms, so it is therefore common that stated requirements are often incomplete, inconsistent, and vague. This is especially true if the requirements are prepared by those who are unfamiliar with the process of converting them to system capabilities, or with the origins of the requirements in terms of operational needs. In practice, the completeness and accuracy of these inputs can be expected to vary with the nature of the system, its degree of departure from predecessor systems, the type of acquisition process employed, and the phase itself.

The above analysis must include interaction with the prospective users of the system to gain a first-hand understanding of their needs and constraints and to obtain their inputs where appropriate. The result of the analysis may be modifications and amplifications of the requirements documents so as to better represent the objectives of the program or the availability of proposed technological improvements. The end objective is to create a firm basis from which the nature and location of the design changes needed to meet the requirements may be defined.

Functional Definition (Functional Analysis and Allocation)

In the systems engineering method, functional design precedes physical or product design to ensure a disciplined approach to an effective organization (configuration) of the functions and to the selection of the implementation that best balances the desired characteristics of the system (e.g., performance and cost).

Translation into Functions. The system elements that may serve as functional building blocks are briefly discussed in Chapter 3. The basic building blocks are at the component level representing elements that perform a single significant function and deal with a single medium, that is, either signals, data, material, or energy. They, in turn, consist of subelements performing lower-level functions and aggregate into functional subsystems. Thus, functional design can be thought of as selecting, subdividing, or aggregating functional elements appropriate to the required tasks and level of system materialization (see Table 4.1).

Decomposition and allocation of each iterative set of requirements and functions for implementation at the next lower level of system definition is a prime responsibility of systems engineering. This first takes place during the concept development stage as follow-on to the definition of the system architecture. It includes identification and description of all functions to be provided, along with the associated quantitative requirements to be met by each subsystem, in order that the prescribed system-level capabilities can in fact be achieved. This information is then reflected in *system functional specifications*, which serve as the basis for the follow-on engineering development stage.

As part of the advanced development phase, these top-level subsystem functions and requirements are further allocated to individual system components within each subsystem. This, as noted earlier, is the lowest level in the design hierarchy that is of direct concern to systems engineering, except in special cases where lower-level elements turn out to be critical to the operation of the system.

Trade-Off Analysis. The selection of appropriate functional elements, as all aspects of design, is an inductive process, in which a set of postulated alternatives is examined, and the one judged to be best for the intended purpose is selected. The systems engineering method relies on making design decisions by the use of trade-off analysis. Trade-off analysis is widely used in all types of decision making, but in systems engineering, it is applied in a particularly disciplined form, especially in the step of physical definition. As the name implies, trade-offs involve the comparison of alternatives, which are superior in one or more required characteristics, with those that are superior in others. To ensure that an especially desirable approach is not overlooked, it is necessary to explore a sufficient number of alternative implementations, all defined to a level adequate to enable their characteristics to be evaluated relative to one another. It is also necessary that the evaluation be made relative to a carefully formulated set of criteria or "MOE." Chapters 8 and 9 contain more detailed discussions of trade-off analysis.

Functional Interactions. One of the single most important steps in system design is the definition of the functional and physical interconnection and interfacing of its building blocks. A necessary ingredient in this activity is the early identification of all significant functional interactions and the ways in which the functional elements may be aggregated so as to group strongly interacting elements together and to make the interactions among the groups as simple as possible. Such organizations (architectures) are referred to as "modular" and are the key to system designs that are readily maintainable and capable of being upgraded to extend their useful life. Another essential ingredient is the identification of all external interactions and the interfaces through which they affect the system.

Physical Definition (Synthesis or Physical Analysis and Allocation)

Physical definition is the translation of the functional design into hardware and software components, and the integration of these components into the total system. In the concept development stage, where all design is still at the functional level, it is nevertheless necessary to visualize or imagine what the physical embodiment of the concept would be like in order to help ensure that the solution will be practically realizable. The process of selecting the embodiment to be visualized is also governed by the general principles discussed below, applied more qualitatively than in the engineering development stage.

Synthesis of Alternative System Elements. The implementation of functional design elements requires decisions regarding the specific physical form that the implementation should take. Such decisions include choice of implementation media, element form, arrangement, and interface design. In many instances, they also offer a choice of approaches, ranging from exploiting the latest technology to relying on proven techniques. As in the case of functional design, such decisions are made by the use of trade-off analysis. There usually being more choices of different physical implementations than functional configurations, it is even more important that good systems engineering practice be used in the physical definition process.

Selection of Preferred Approach. At various milestones in the system life cycle, the selection of a preferred approach, or approaches, will need to be made. It is important to understand that this selection process changes depending on the phase within the life cycle. Early phases may require selecting a several approaches to explore, while later phases may require a down-select to a single approach. Additionally, the level of decisions evolves. Early decisions relate to the system as a whole; later decisions focus on subsystems and components.

As stated previously, to make a meaningful choice among design alternatives, it is necessary to define a set of evaluation criteria and to establish their relative priority. Among the most important variables to be considered in the physical definition step is the relative affordability or cost of the alternatives and their relative risk of successful accomplishment. In particular, early focus on one particular implementation concept should be avoided.

Risk as a component of trade-off analysis is basically an estimate of the probability that a given design approach will fail to produce a successful result whether because of deficient performance, low reliability, excessive cost, or unacceptable schedule. If the component risk appears substantial, the risk to the overall project must be reduced (risk abatement) by either initiating an intensive component development effort, by providing a backup using a proven but somewhat less capable component, by modifying the overall technical approach to eliminate the need for the particular component that is in doubt, or, if these fail, by relaxing the related system performance specification. Identifying significantly high-risk system elements and determining how to deal with them are an essential systems engineering responsibility. Chapter 5 discusses the risk management process and its constituent parts.

Proper use of the systems engineering method thus ensures that

1. all viable alternatives are considered;
2. a set of evaluation criteria is established; and
3. the criteria are prioritized and quantified where practicable.

Whether or not it is possible to make quantitative comparisons, the final decision should be tempered by judgment based on experience.

Interface Definition. Implicit in the physical definition step is the definition and control of *interfaces*, both internal and external. Each element added or elaborated in the design process must be properly connected to its neighboring elements and to any external inputs or outputs. Further, as the next lower design level is defined, adjustments to the parent elements will inevitably be required, which must in turn be reflected in adjustments to their previously defined interfaces. All such definitions and readjustments must be incorporated into the model design and interface specifications to form a sound basis for the next level of design.

Design Validation (Verification and Evaluation)

In the development of a complex system, even though the preceding steps of the design definition may have been carried out apparently in full compliance with requirements, there still needs to be an explicit validation of the design before the next phase is undertaken. Experience has shown that there are just too many opportunities for undetected errors to creep in. The form of such validation varies with the phase and degree of system materialization, but the general approach is similar from phase to phase.

Modeling the System Environment. To validate a model of the system, it is necessary to create a model of the environment with which the system can interact to see if it produces the required performance. This task of modeling the system environment extends throughout the system development cycle. In the concept development stage, the model is largely functional, although some parts of it may be physical, as when an experimental version of a critical system component is tested over a range of ambient conditions.

In later stages of development, various aspects of the environment may be reproduced in the laboratory or in a test facility, such as an aerodynamic wind tunnel or inertial test platform. In cases where the model is dynamic, it is more properly called a simulation, in which the system design is subjected to a time-varying input to stimulate its dynamic response modes.

As the development progresses into the engineering development stage, modeling the environment becomes increasingly realistic, and environmental conditions are embodied in system and component test equipment, such as environmental chambers, or shock and vibration facilities. During operational evaluation testing, the environment is, insofar as is practicable, made identical to that in which the system will eventually operate. Here, the model has transitioned into greater reality.

Some environments that are of great significance to system performance and reliability can only be imperfectly understood and are very difficult to simulate, for example, the deep ocean and exoatmospheric space. In such cases, defining and simulating the environment may become a major effort in itself. Even environments that were thought to be relatively well understood can yield surprises, for example, unusual radar signal refraction over the Arabian Desert.

At each step, the system development process requires a successively more detailed definition of the requirements that the system must meet. It is against these environmental requirements that the successive models of the system are evaluated and refined. A lesson to be learned is that the effort required to model the environment of a system for the purpose of system T&E needs to be considered at the same level of priority as the design of the system itself and may even require a separate design effort comparable to the associated system design activity.

Tests and Test Data Analysis. The definitive steps in the validation of the system design are the conduct of tests in which the system model (or a significant portion of it) is made to interact with a model of its environment in such a way that the effects can be measured and analyzed in terms of the system requirements. The scope of such tests evolves with the degree of materialization of the system, beginning with paper calculations and ending with operational tests in the final stages. In each case, the objective is to determine whether or not the results conform to those prescribed by the requirements, and if not, what changes are required to rectify the situation.

In carrying out the above process, it is most important to observe the following key principles:

1. All critical system characteristics need to be stressed beyond their specified limits to uncover incipient weak spots.
2. All key elements need to be instrumented to permit location of the exact sources of deviations in behavior. The instruments must significantly exceed the test articles in precision and reliability.
3. A test plan and an associated test data analysis plan must be prepared to assure that the requisite data are properly collected and are then analyzed as necessary to assure a realistic assessment of system compliance.
4. All limitations in the tests due to unavoidable artificialities need to be explicitly recognized and their effect on the results compensated or corrected for, as far as possible.
5. A formal test report must be prepared to document the degree of compliance by the system and the source of any deficiencies.

The test plan should detail each step in the test procedure and identify exactly *what information* will be recorded prior to, during, and at the conclusion of each test step, as well as *how* and *by whom* it will be recorded. The test data analysis plan should then define how the data would be reduced, analyzed, and reported along with specific criteria that will be employed to demonstrate system compliance.

To the extent that the validation tests reveal deviations from required performance, the following alternatives need to be considered:

1. Can the deviation be due to a deficiency in the environmental simulation (i.e., test equipment)? This can happen because of the difficulty of constructing a realistic model of the environment.
2. Is the deviation due to a deficiency in the design? If so, can it be remedied without extensive modifications to other system elements?
3. Is the requirement at issue overly stringent? If so, a request for a deviation may be considered. This would constitute a type of feedback that is characteristic of the system development process.

Preparation for the Next Phase

Each phase in the system development process produces a further level of requirements or specifications to serve as a basis for the next phase. This adds to, rather than replaces, previous levels of requirements and serves two purposes:

1. It documents the design decisions made in the course of the current phase.
2. It establishes the goals for the succeeding phase.

Concurrent with the requirements analysis and allocation activity, systems engineering, acting in concert with project management, is also responsible for the definition of specific technical objectives to be met, and for the products (e.g., hardware/software components, technical documentation, and supporting test data) that will be provided in response to the stated requirements for inputs to the next phase. These identified end products of each phase are also often accompanied by a set of intermediate technical milestones that can be used to judge technical progress during each particular design activity.

The task of defining these requirements or specifications and the efforts to be undertaken in implementing the related design activities is an essential part of system development. Together, these constitute the official guide for the execution of each phase of the development.

It must be noted, however, that in practice, the realism and effectiveness of this effort, which is so critical to the ultimate success of the project, depends in large part on good communication and cooperation between systems engineering and project management on the one hand, and on the other, the design specialists who are ultimately the best judges of what can and cannot be reasonably accomplished given the stated requirements, available resources, and allotted time scale.

Since the nature of the preparation for the next phase varies widely from phase to phase, it is not usually accorded the status of a separate step in the systems engineering method; most often, it is combined with the validation process. However, this does not diminish its importance because the thoroughness with which it is done directly affects the requirements analysis process at the initiation of the next phase. In any event, the definition of the requirements and tasks to be performed in the next phase serves an important interface function between phases.

Systems Engineering Method over the System Life Cycle

To illustrate how the systems engineering method is applied in successive phases of the system life cycle, Table 4.3 lists the primary focus of each of the four steps of the method for each of the phases of the system life cycle. As indicated earlier in Table 4.1, it is seen that as the phases progress, the focus shifts to more specific and detailed (lower-level) elements of the system until the integration and evaluation phase.

The table also highlights the difference in character of the physical definition and design validation steps in going from the concept development to the engineering development stage. In the concept development stage (left three columns), the defined concepts are still in functional form (except where elements of the previous or other systems are applied without basic change). Accordingly, physical implementation has not yet begun, and design validation is performed by analysis and simulation of the functional elements. In the engineering stage, implementation into hardware and software proceeds to lower and lower levels, and design validation includes tests of experimental, prototype, and finally production system elements and the system itself.

TABLE 4.3. Systems Engineering Method over Life Cycle

	Phase					
	Concept development			*Engineering development*		
Step	Needs analysis	Concept exploration	Concept definition	Advanced development	Engineering design	Integration and evaluation
Requirements analysis	Analyze needs	Analyze operational requirements	Analyze performance requirements	Analyze functional requirements	Analyze design requirements	Analyze tests and evaluation requirements
Functional definition	Define system objectives	Define subsystem functions	Develop functional architecture component functions	Refine functional architecture subcomponent functions140	Define part functions	Define functional tests
Physical definition	Define system capabilities; visualize subsystems, ID technology	Define system concepts, visualize components	Develop physical architecture components	Refine physical architecture; specify component construction	Specify subcomponent construction	Define physical tests; specify test equipment and facilities
Design validation	Validate needs and feasibility	Validate operational requirements	Evaluate system capabilities	Test and evaluate critical subsystems	Validate component construction	Test and evaluate system

In interpreting both Tables 4.3 and 4.1, it should be borne in mind that in a given phase of system development, some parts of the system might be prototyped to a more advanced phase to validate critical features of the design. This is particularly true in the advanced development phase, where new potentially risky approaches are prototyped and tested under realistic conditions. Normally, new software elements are also prototyped in this phase to validate their basic design.

While these tables present a somewhat idealized picture, the overall pattern of the iterative application of the systems engineering method to successively lower levels of the system is an instructive and valid general view of the process of system development.

Spiral Life Cycle Model

The iterative nature of the system development process, with the successive applications of the systems engineering method to a stepwise materialization of the system has been captured in the so-called spiral model of the system life cycle. A version of this model as applied to life cycle phases is shown in Figure 4.13. The sectors representing the four steps in the systems engineering method defined in the above section are shown separated by heavy radial lines. This model emphasizes that each phase of the development of a complex system necessarily involves an iterative application of the systems engineering method and the continuing review and updating of the work performed and conclusions reached in the prior phases of the effort.

4.5 TESTING THROUGHOUT SYSTEM DEVELOPMENT

Testing and evaluation are not separate functions from design but rather are inherent parts of design. In basic types of design, for example, as of a picture, the function of T&E is performed by the artist as part of the process of transferring a design concept to canvas. To the extent that the painting does not conform to the artist's intent, he or she alters the picture by adding a few brushstrokes, which tailor the visual effect (performance) to match the original objective. Thus, design is a closed-loop process in which T&E constitutes the feedback that adjusts the result to the requirements that it is intended to meet.

Unknowns

In any new system development project, there are a great many unknowns that need to be resolved in the course of producing a successful product. For each significant departure from established practice, the result cannot be predicted with assurance. The project cost depends on a host of factors, none of them known precisely. The resolution of interface incompatibilities often involves design adjustment on both sides of the interface, which frequently leads to unexpected and sometimes major technical difficulties.

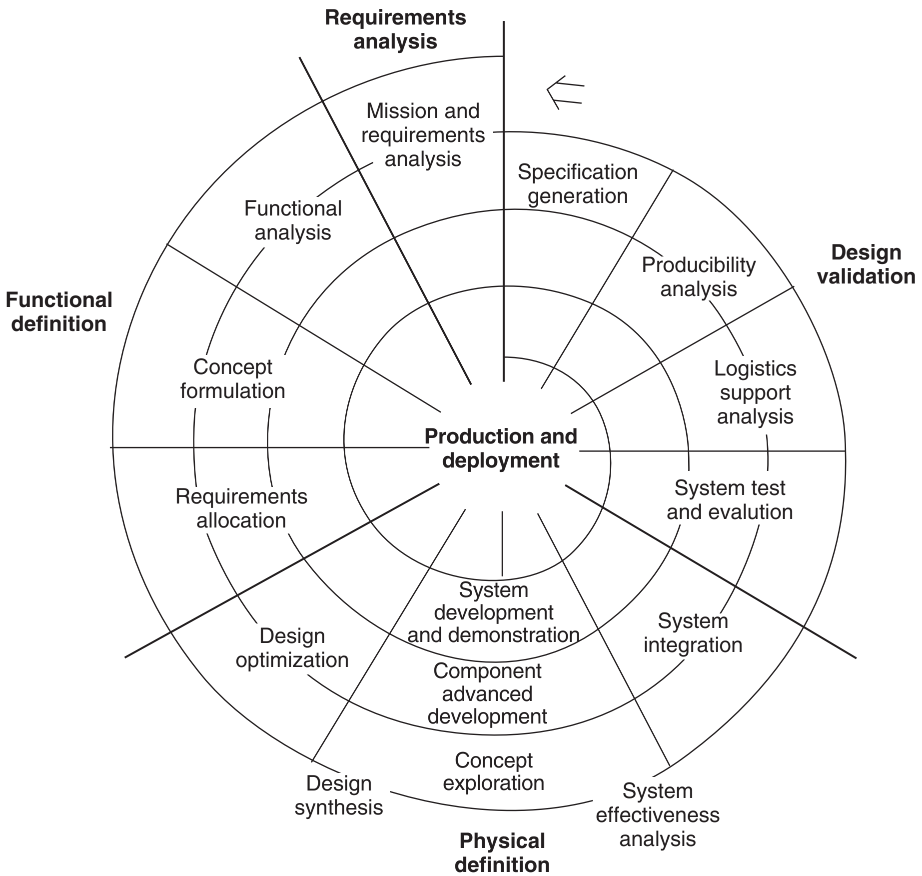

Figure 4.13. Spiral model of the system life cycle.

An essential task of systems engineering is to guide the development of the system so that the unknowns are turned into knowns as early in the process as possible. Any surprises occurring late in the program can prove to be many times more costly than those encountered in its early phases.

Many unknowns are evident at the beginning, and may be called "known unknowns." These are identified early as potential problem areas and are therefore singled out for examination and resolution. Usually, this can be accomplished through a series of critical experiments involving simulations and/or experimental hardware and software. However, many other problem areas are only identified later when they are discovered during system development. These unanticipated problems are often identified as unknown unknowns or "unk-unks" to distinguish them from the group of known unknowns that were recognized at the outset and dealt with before they could seriously impact the overall development process.

Transforming the Unknown into the Known

The existence of unk-unks makes the task of attempting to remove all the unknowns far more difficult. It forces an active search for hidden traps in the favored places of technical problems. It is the task of the systems engineer to lead this search based on experience gathered during previous system developments and supported by a high degree of technical insight and a "What if …?" attitude.

Since every unknown poses an uncertainty in the accomplishment of the final objective, it represents a potential risk. In fact, unknowns present the principal risks in any development program. Hence, the task of risk assessment and integration is one and the same as that of identifying unknowns and resolving them.

The tools for resolving unknowns are analysis, simulation, and test, these being the means for discovering and quantifying critical system characteristics. This effort begins during the earliest conceptual stages and continues throughout the entire development, only changing in substance and character and not in objective and approach.

In designing a new system or a new element of a system that requires an approach never attempted before under the same circumstances (as, e.g., the use of new materials for making a highly stressed design element), the designer faces a number of unknowns regarding the exact manner in which the new design when implemented will perform (e.g., the element made of a new material may not be capable of being formed into the required shape by conventional tools). In such cases, the process of testing serves to reveal whether or not the unknown factors create unanticipated difficulties requiring significant design changes or even abandonment of the approach.

When a new design approach is undertaken, it is unwise to wait until the design is fully implemented before determining whether or not the approach is sound. Instead, testing should first be done on a theoretical or experimental model of the design element, which can be created quickly and at a minimum cost. In doing so, a judgment must be made as to the balance between the potential benefit of a greater degree of realism of the model and the time and cost of achieving it. This is very often a system-level rather than a component-level decision, especially if the performance of the element can have a system impact. If the unknowns are largely in the functional behavior of the element, then a computational model or a simulation is indicated. If, on the other hand, the unknowns are concerned with the material aspects, an experimental model is required.

Systems Engineering Approach to Testing

The systems engineering approach to testing can be illustrated by comparing the respective views of testing by the design engineer, the test engineer, and the systems engineer. The design engineer wants to be sure that a component passes the test, wanting to know, "Is it OK?" The test engineer wants to know that the test is thorough so as to be sure the component is stressed enough. The systems engineer wants to be sure to find and identify all deficiencies present in the component. If the component fails a test, the systems engineer wants to know why, so that there will be a basis for devising changes that will eliminate the deficiency.

It is evident from the above that the emphasis of systems engineering is not only on the test conditions but also on the acquisition of data showing exactly how the various parts of the system did or did not perform. Furthermore, the acquisition of data itself is not enough; it is necessary to have in hand procedures for analyzing the data. These are often complicated and require sophisticated analytical techniques, which must be planned in advance.

It also follows that a systems engineer must be an active participant in the formulation of the test procedures and choice of instrumentation. In fact, the prime initiative for developing the test plan should lie with systems engineering, working in close cooperation with test engineering. To the systems engineer, a test is like an experiment is to a scientist, namely, a means of acquiring critical data on the behavior of the system under controlled circumstances.

System T&E

The most intensive use of testing in the system life cycle takes place in the last phase of system development, integration and evaluation, which is the subject of Chapter 13. Chapter 10 also contains a section on T&E during the advanced development phase.

4.6 SUMMARY

Systems Engineering through the System Life Cycle

A major system development program is an extended complex effort to satisfy an important user need. It involves multiple disciplines and applies new technology, requires progressively increasing commitment of resources, and is conducted in a stepwise manner to a specified schedule and budget.

System Life Cycle

The system life cycle may be divided into three major stages.

Concept Development. Systems engineering establishes the system need, explores feasible concepts, and selects a preferred system concept. The concept development stage may be further subdivided into three phases:

1. *Needs Analysis:* defines and validates the need for a new system, demonstrates its feasibility, and defines system operational requirements;
2. *Concept Exploration:* explores feasible concepts and defines functional performance requirements; and
3. *Concept Definition:* examines alternative concepts, selects the preferred concept on the basis of performance, cost, schedule, and risk, and defines system functional specifications (A-Spec).

Engineering Development. Systems engineering validates new technology, transforms the selected concept into hardware and software designs, and builds and tests production models. The engineering development stage may be further subdivided into three phases:

1. *Advanced Development:* identifies areas of risk, reduces these risks through analysis, development, and test, and defines system development specifications (B-Spec);
2. *Engineering Design:* performs preliminary and final design and builds and tests hardware and software components, for example, configuration items (CIs); and
3. *Integration and Evaluation:* integrates components into a production prototype, evaluates the prototype system, and rectifies deviations.

Postdevelopment. Systems engineering produces and deploys the system and supports system operation and maintenance. The postdevelopment stage is further subdivided into two phases:

1. *Production:* develops tooling and manufactures system products, provides the system to the users, and facilitates initial operations; and
2. *Operations and Support:* supports system operation and maintenance, and develops and supports in-service updates.

Evolutionary Characteristics of the Development Process

Most new systems evolve from predecessor systems—their functional architecture and even some components may be reusable.

A new system progressively "materializes" during its development. System descriptions and designs evolve from concepts to reality. Documents, diagrams, models, and products all change correspondingly. Moreover, key participants in system development change during development; however, systems engineering plays a key role throughout all phases.

The Systems Engineering Method

The systems engineering method involves four basic steps:

1. *Requirements Analysis*—identifies why requirements are needed,
2. *Functional Definition*—translates requirements into functions,
3. *Physical Definition*—synthesizes alternative physical implementations, and
4. *Design Validation*—models the system environment.

These four steps are applied repetitively in each phase during development. Application of the systems engineering method evolves over the life cycle—as the system progressively materializes, the focus shifts from system level during needs analysis down to component and part levels during engineering design.

Testing throughout System Development

Testing is a process to identify unknown design defects in that it verifies resolution of known unknowns and uncovers unknown unknowns (unk-unks) and their causes. Late resolution of unknowns may be extremely costly; therefore, test planning and analysis is a prime systems engineering responsibility.

PROBLEMS

4.1 Identify a recent development (since 2000) of a complex system (commercial or military) of which you have some knowledge. Describe the need it was developed to fill and the principal ways in which it is superior to its predecessor(s). Briefly describe the new conceptual approach and/or technological advances that were employed.

4.2 Advances in technology often lead to the development of a new or improved system by exploiting an advantage not possessed by its predecessor. Name three different types of advantages that an advanced technology may offer and cite an example of each.

4.3 If there is a feasible and attractive concept for satisfying the requirements for a new system, state why it is important to consider other alternatives before deciding which to select for development. Describe some of the possible consequences of failing to do so.

4.4 The space shuttle was an example of an extremely complicated system using leading edge technology. Give three examples of shuttle components that you think represented unproven technology at the time of its development, and which much have required extensive prototyping and testing to reduce operational risks to an acceptable level.

4.5 What steps can the systems engineer take to help ensure that system components designed by different technical groups or contractors will fit together and interact effectively when assembled to make up the total system? Discuss in terms of mechanical, electrical, and software system elements.

4.6 For six of the systems listed in Tables 1.1 and 1.2, list their "predecessor systems." For each, indicate the main characteristics in which the current systems are superior to their predecessors.

4.7 Table 4.2 illustrates the evolution of system models during the system development process. Describe how the evolution of requirements documents illustrates the materialization process described in Table 4.1.

4.8 Look up a definition of the "scientific method" and relate its steps to those postulated for the systems engineering method. Draw a functional flow diagram of the scientific method parallel to that of Figure 4.11.

4.9 Select one of the household appliances listed below:

- automatic dishwasher
- washing machine

- television set
  (a) State the *functions* that it performs during its operating cycle. Indicate the primary medium (signals, data, material, or energy) involved in each step and the basic function that is performed on this medium.
  (b) For the selected appliance, describe the physical elements involved in the implementation of each of the above functions.

FURTHER READING

S. Biemer and A. Sage. Chapter 4: Systems engineering: Basic concepts and life cycle. In *Agent-Directed Simulation and Systems Engineering*, L. Yilmaz and T. Oren, eds. John Wiley & Sons, 2009.

B. Blanchard. *System Engineering Management*, Third Edition. John Wiley & Sons, 2004.

B. Blanchard and W. Fabrycky. *System Engineering and Analysis*, Fourth Edition. Prentice Hall, 2006.

D. Buede. *The Engineering Design of Systems: Models and Methods*, Second Edition. John Wiley & Sons, 2009.

H. Chesnut. *System Engineering Methods*. John Wiley, 1967.

P. DeGrace and L. H. Stahl. *Wicked Problems, Righteous Solutions*. Yourdon Press, Prentice Hall, 1990.

H. Eisner. *Computer-Aided Systems Engineering*. Prentice Hall, 1988, Chapter 10.

H. Eisner. *Essentials of Project and Systems Engineering Management*, Second Edition. John Wiley & Sons, 2002.

A. D. Hall. *A Methodology for Systems Engineering*. Van Nostrand, 1962, Chapter 4.

M. Maier and E. Rechtin. *The Art of Systems Architecting*, Third Edition. CRC Press, 2009.

J. N. Martin. *Systems Engineering Guidebook: A Process for Developing Systems and Products*. CRC Press, 1997, Chapters 2–5.

E. Rechtin. *Systems Architecting: Creating and Building Complex Systems*. Prentice Hall, 1991, Chapters 2 and 4.

N. B. Reilly. *Successful Systems for Engineers and Managers*. Van Nostrand Reinhold, 1993, Chapter 3.

A. P. Sage. *Systems Engineering*. McGraw Hill, 1992, Chapter 2.

A. P. Sage and J. E. Armstrong, Jr. *Introduction to Systems Engineering*. Wiley, 2000, Chapter 2.

A. Sage and S. Biemer. Processes for system family architecting, design and integration. *IEEE Systems Journal*, 2007, 1, 5–16.

S. M. Shinners. *A Guide for System Engineering and Management*. Lexington Books, 1989, Chapter 1.

R. Stevens, P. Brook, K. Jackson, and S. Arnold. *Systems Engineering, Coping with Complexity*. Prentice Hall, 1998, Chapters 7 and 8.