1

INTRODUCTION TO SYSTEM ENGINEERING

This text deals with *system engineering*, or the orderly process of bringing a system into being and the subsequent effective and efficient operation and support of that system throughout its projected life cycle. It constitutes an interdisciplinary approach and means for enabling the realization and the follow-on deployment of a successful system.

A *system* comprises a complex combination of resources (in the form of human beings, materials, equipment, hardware, software, facilities, data, information, services, etc.), integrated in such a manner as to fulfill a specified operational requirement. A system is developed to accomplish a specific function, or a series of functions, with the objective of responding to some identified need. The various elements of a system must be directly tied to and supportive in the accomplishment of some given mission scenario or series of scenarios.

A system may be classified as a *natural system*, *human-made system*, *physical system*, *conceptual system*, *closed-loop system*, *open-loop system*, *static system*, *dynamic system*, and so on. This text addresses primarily human-made systems that are physical, dynamic, and open loop in structure. Further, the objective is to address the system in the context of its *whole* versus dealing with its components only. Of significant importance is the realization that ultimate system performance is dependent not only on the complete and timely integration of its various components, but also on establishing the proper interrelationships among these components. By accomplishing this through the application of system engineering principles and concepts, a value-added component can be realized.

A system may vary in form, fit, and/or function. One may be dealing with a group of aircraft accomplishing a mission at a specific geographical location; a cloud-based communication network for the processing of information on a worldwide basis;

a tightly integrated collection of integrated circuit chips, printed circuit boards, and higher-level modular electronics processing huge amounts of Internet and consumer mobile data for products in variety of vertical sectors; a power distribution capability involving waterways and electrical power-generating units; a healthcare capability including a group of hospitals and mobile units serving a given community; a manufacturing facility that produces *x* products in a designated time frame; or a small vehicle providing the transportation of certain cargo from one location to another. A system may also be contained within some overall hierarchy such as an aircraft within an airline system, which is within a larger regional transportation system, which is within a worldwide transportation capability, and so on. In this context, we may be dealing with *system of systems (SOS)*, a popular term currently being applied in describing highly complex systems within some higher-level structure. The objective is to be able to adequately define and describe the overall boundaries of the particular system being addressed and its interfaces (and interrelationships) across the board.

A system must have a purpose! It includes not only those basic elements that are directly related to accomplishing the mission itself (configuration items, subsystems, segments, components or parts) but also those enabling elements that are necessary for keeping the system in service or ending its service, processes or products used to enable a system development, test, production, training, deployment, support, and ultimate disposal. In other words, for a system to be able to accomplish its intended mission, it must also include its total maintenance and support infrastructure.

The objectives of this chapter are: to address the subject of systems in general, to define some key terms and the characteristics of systems, to identify the need for and the basic requirements for bringing systems into being and for later evaluating systems in terms of their effectiveness in a user's environment, and to provide an introduction to *system engineering* and the associated management activities inherent in and supportive of the system engineering process.

1.1 DEFINITION OF A SYSTEM

In order to ensure a good and common understanding of the material throughout this text, it seems appropriate to commence with a few definitions. As a start, one should first establish a basic definition for a *system*. Although this may appear to be overly simplistic, experience has indicated that people throughout the world tend to utilize the term rather loosely to describe many different situations and configurations. Further, there is a lack of consistency in the application of system engineering principles and concepts. Thus, it is important to first review a few terms to establish a baseline for further discussion.

1.1.1 The Characteristics of a System

The term *system* stems from the Greek *systēma*, meaning an "organized whole." *Merriam-Webster's Collegiate Dictionary* defines a system as "a regularly interacting or interdependent group of items forming a unified whole."[1] One of the early Military Standards on the subject, MIL-STD-499, defines a system as "a composite of equipment, skills, and techniques capable of performing and/or supporting an operational role. A complete system includes all equipment, related facilities, material, software, services, and personnel required for its operation and support to the degree that it can be considered a self-sufficient unit in its intended environment."[2] A more recent document, EIA/IS-632, defines a system as "an integrated composite of people, products, and processes that provide a capability to satisfy a stated need or objective."[3]

In the world of "semiconductor systems," integrated chips have become so complex in both design and manufacturing that they are called, "Systems-on-Chip (SoC)." The integrated circuit in a SoC may contain digital, analog, mixed-signal, and often radio-frequency functions all on a single chip substrate. Even in the early days of the SoC, the IEEE recognized that, "the definition of the 'system' design and manufactured on a chip has significantly changed and expanded as did the technology, skills, tools, and methodologies required to produce it."[4]

"Software systems" offer an example of the many and varied use of both the term "software" and "systems." System software operates and controls electronic hardware to provide a platform for running application software. System software can further be separated into categories of firmware/drivers, operating systems and applications.

Given the variations in the basic definition of a "system," the leadership of INCOSE (International Council on Systems Engineering) assigned the current Fellows of the Council to develop a consensus definition. After a few iterations, the following definition evolved:

> A "system" is a construct or collection of different elements that together produce results not obtainable by the elements alone. The elements, or parts, can include people, hardware, software, facilities, policies, and documents; that is, all things required to produce system-level results. The results include system-level qualities, properties, characteristics, functions, behavior, and performance. The value added by the system as a whole, beyond that contributed independently by the parts, is primarily created by the relationship among the parts; that is, how they are interconnected.[5]

In essence, a system constitutes a set of interrelated components working together with the common objective of fulfilling some designated need.

Although the preceding definitions reflect a good initial overview, a greater degree of detail and precision is required to provide a good working definition acceptable for

[1] Merriam-Webster's Collegiate Dictionary. 10th Ed. (Springfield, MA: Merriam-Webster, Inc. 1988).
[2] Military Standard, MIL-STD-499, System Engineering Management. (Department of Defense, July 17, 1969).
[3] EIA/IS-632, Processes for Engineering a System, Electronic Industries Association (EIA), Washington, D.C. December 1994.
[4] A.M. Rincon, W. R. Lee, and M. Slattery, "The Changing Landscape of System-on-a-Chip Design", Proceedings of the IEEE 1999.
[5] INCOSE, 7670 Opportunity Road, Suite 220, San Diego, CA 92111. This definition was developed in the Fall of 2001.

describing the principles and concepts of system engineering. To facilitate this objective, a system may be defined further in terms of the following general characteristics:

1. A system constitutes a *complex combination of resources* in the form of human beings, materials, equipment, hardware, software, facilities, data, money, and so on. To accomplish many functions often requires large amounts of personnel, equipment, facilities, and data (e.g., an airline or a manufacturing capability). Such resources must be combined in an *effective* manner, as it is too risky to leave this to chance alone.
2. A system is contained within some form of *hierarchy*. An airplane may be included within an airline, which is part of an overall transportation capability, which is operated in a specific geographic environment, which is part of the world, and so on. As such, the system being addressed is highly influenced by the performance of the higher-level system, and these external factors must be evaluated.
3. A system may be broken down into *subsystems* and related components, the extent of which depends on complexity and the function(s) being performed. Dividing the system into smaller units allows for a simpler approach relative to the initial allocation of requirements and the subsequent analysis of the system and its functional interfaces. A system is made up of many different components; these components *interact* with each other, and these interactions must be thoroughly understood by the system designer and/or analyst. Because of these interactions among components, it is impossible to produce an effective design by considering each component separately. One must view the system as a whole, break down the system into components, study the components and their interrelationships, and then put the system back together as an integrated whole.
4. A system must have a *purpose*. It must be functional, able to respond to some identified need, and able to achieve its overall objective in a cost-effective manner. There may be a conflict of objectives, influenced by the higher-level system in the hierarchy, and the system must be capable of meeting its stated purpose in the best way possible.

As a point of emphasis, a system must respond to an identified *functional need*. Thus, the elements of a system must include not only those items that relate directly to the accomplishment of a given scenario or mission profile, but also those elements of logistics and the maintenance and support infrastructure that have to be available and in place should a failure of a prime element(s) occur. In other words, if one is to ensure the successful completion of a mission, all of the supporting elements must be available, in place, and ready to respond to a given need.[6]

[6]In many instances, the logistics and maintenance support infrastructure is not addressed or included as an element of the system, or as a major subsystem, but is treated separately and "after the fact." The approach assumed throughout this text is that this is included in the context of a major subsystem and addressed as an inherent part of the whole.

It should be noted that in many instances the term "system" is rather loosely applied to other elements such as *software systems*, *semiconductor systems*, *systems-on-chips*, *firmware systems*, *hardware systems*, *embedded electronic systems*, and the like. In most cases, these elements should be considered as major "subsystems," forming a part of some larger "system" that directly supports some specific mission objective. Software, for example, is NOT a "system" in itself, and does not accomplish any type of a mission without being properly integrated with applicable hardware, personnel, facilities, and so on. It is important here to be "specific" in arriving at and utilizing given definitions, particularly with the objective of facilitating good communications across the board.

1.1.2 Categories of Systems

In defining systems in terms of the general characteristics presented, it readily becomes apparent that some degree of further classification is desirable. There are many different types of systems, and there are some variations in terms of similarities and dissimilarities. To provide some insight into the variety of systems in existence, a partial listing of categories follows:[7]

1. *Natural and man-made systems*. Natural systems include those that came into being through natural processes. Examples include a river system and an energy system. Man-made systems are those that have been developed by human beings, which results in a wide variety of capabilities. As all man-made systems are embedded in the natural world and there are numerous interfaces that must be addressed. For instance, the development and construction of a hydroelectric power system located on a river system creates impacts on both sides of the spectrum, and it is essential that the systems approach involving both the natural and man-made segments of this overall capability be implemented.
2. *Physical and conceptual systems*. Physical systems are those made up of real components occupying space. By contrast, conceptual systems can be an organization of ideas, a set of specifications and plans, a series of abstract concepts, and so on. Conceptual systems often lead directly into the development of physical systems, and there is a certain degree of commonality in terms of the type of processes employed. Again, the interfaces may be many, and there is a need to address these elements in the context of a higher-level system in the overall hierarchy.
3. *Static and dynamic systems*. Static systems include those that have structure, but without activity (as viewed in a relatively short period of time). A highway bridge and a warehouse are examples. A dynamic system is one that combines structural components with activity. An example is a production system combining a manufacturing facility, capital equipment, utilities, conveyors,

[7]This categorization follows the general format presented in B. Blanchard and W. Fabrycky, *Systems Engineering and Analysis*, 5th ed., Pearson Prentice Hall, 2011. These categories represent only a few possibilities.

workers, transportation vehicles, data, software, managers, and so on. Although there may be specific points in time when all system components are static in nature, the successful accomplishment of system objectives does require activity and the dynamic aspects of system operation do prevail throughout a given scenario.

4. *Closed and open-loop systems*. A closed system is one that is relatively self-contained and does not significantly interact with its environment. The environment provides the medium in which the system operates; however, the impact is minimal. A chemical equilibrium process and an electrical circuit (with a built-in feedback and control loop) are examples. Conversely, open-loop systems interact with their environments. Boundaries are crossed (through the flow of information, energy, and/or matter), and there are numerous interactions both among the various system components and up and down the overall system hierarchical structure. A system/product logistic support capability is an example.

These categories are presented to stimulate further thought relative to the definition of a system. It is not easy to classify a system as being either closed or open, and the precise relationships between natural and man-made systems may not be well defined. However, the objective here is to gain a greater appreciation for the many different considerations required in dealing with system engineering and its process. This text tends to deal mainly with *man-made* systems that are *physical* by nature, *dynamic* in operation, and of the *open-loop* variety.

The systems addressed herein may include a wide variety of *functional* entities. There are transportation systems, communication systems, manufacturing systems, information processing systems, logistics and supply-chain systems, and so on, as indicated in Figure 1.1. In each instance, there are *inputs*, there are *outputs*, there are external *constraints* imposed on a system, and there are the required *mechanisms* necessary to realize the desired results. Within the framework of the system, there are products and processes.

A system is composed of many different elements, including those that are directly utilized in the actual accomplishment of a mission (e.g., prime equipment, embedded electronic hardware, operating and control software, operating personnel, facilities, data) and the elements of maintenance support (e.g., maintenance personnel, test equipment, maintenance facilities, spares and repair parts and inventories). Although the support infrastructure is not often considered an element of a system per se, the system may not be able to complete its designated function in its absence. Thus, the support infrastructure is addressed as a major system element, presented in the context of the system life cycle. Figure 1.2 identifies the major elements of a system.

With the objective of providing some additional emphasis relative to the definition of a "system," Figure 1.3 is presented to convey that a "system" must include not only those elements which are directly related to accomplishing a given mission scenario, but also those enabling system elements that are required in addition and are necessary to support the intended objective.

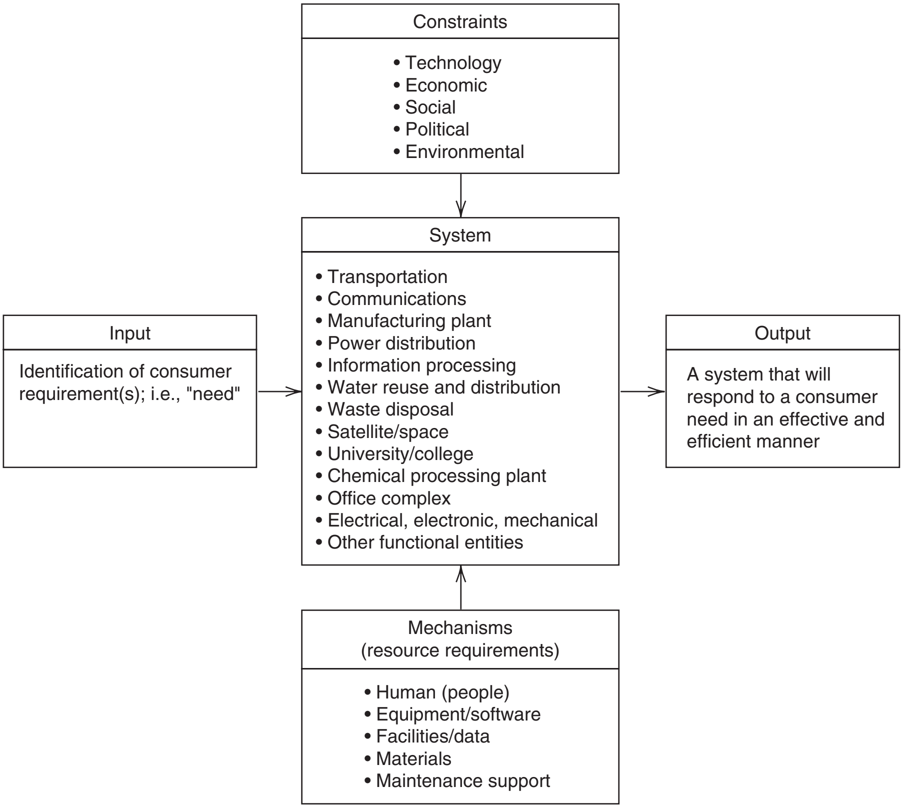

FIGURE 1.1 The system.

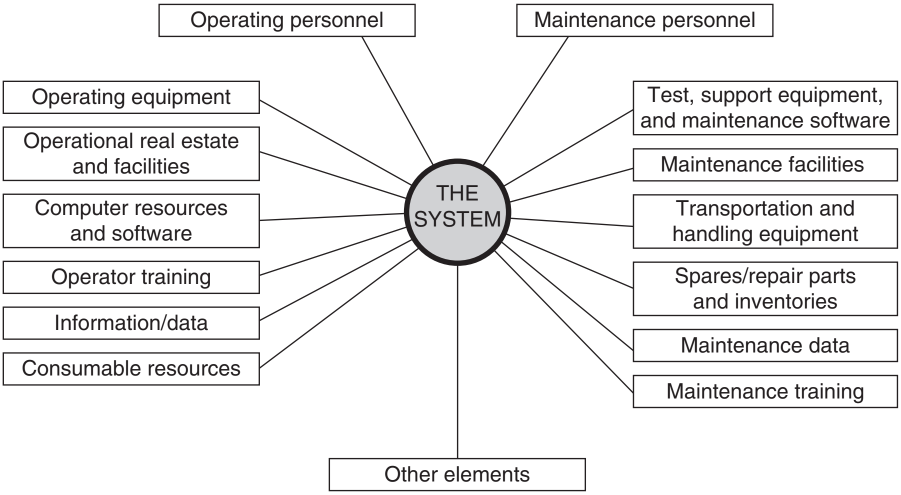

FIGURE 1.2 Major elements of a system.

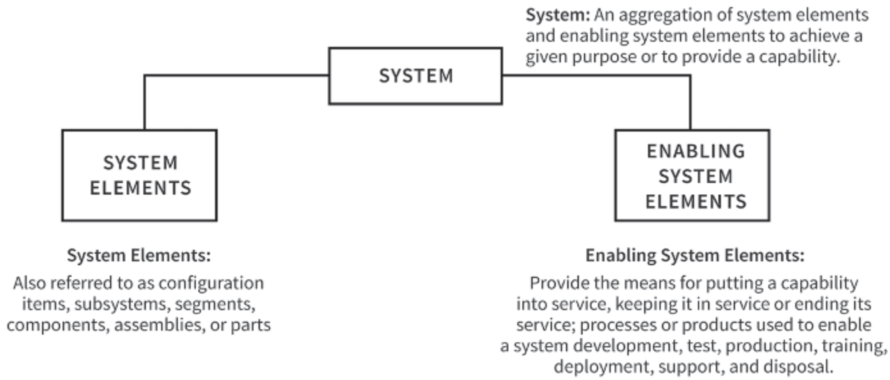

FIGURE 1.3 System, system elements, and enabling system elements. *Source: Defense Acquisition Guide*, Chapter 4, Department of Defense 2014.

1.1.3 System of Systems (SOS)

A system may be contained within some form of *hierarchy*, as shown in Figure 1.4, where there are different layers of systems within an overall configuration. For example, there is an aircraft system, within an air transportation system (e.g., commercial airline), within an overall regional transportation capability, and so on. Often, and particularly in dealing with large-scale systems, such a configuration may be referred to as a *system of systems (SOS)*. Basically, an SOS may be defined as:[8]

> a collection of component systems that produce results unachievable by the individual systems alone. Each system in the SOS structure is likely to be operational in its own right, as well be contributing in the accomplishment of some higher-level mission requirement. The life cycles of the individual systems may vary somewhat as there will be additions and deletions at different times, as long as the mission requirements for any given system are met. Thus, there may be some new developments in progress at the same time as other elements are being retired for disposal.

Referring to Figure 1.4, the question is—are we addressing a transportation system that includes many different types of air and ground vehicles, or an airline that includes many different aircraft, or a specific aircraft with its crew and all of its support? It is not uncommon for a group of individuals to get together and discuss a particular issue, each having a different perception as to the *system* being addressed. One person's system of interest can be viewed as an element (or subsystem) in another person's system of interest.

In defining the requirements for a system, one must be careful in relating such to a specific functional objective, establishing the appropriate hierarchical relationships, defining the boundaries for each system in the hierarchy, and identifying some of the

[8]INCOSE-TP-2003–002–03.2.2, *Systems Engineering Handbook*, Version 3, International Council On Systems Engineering, 7670 Opportunity Rd, Ste 220, San Diego, CA 92111, 2011.

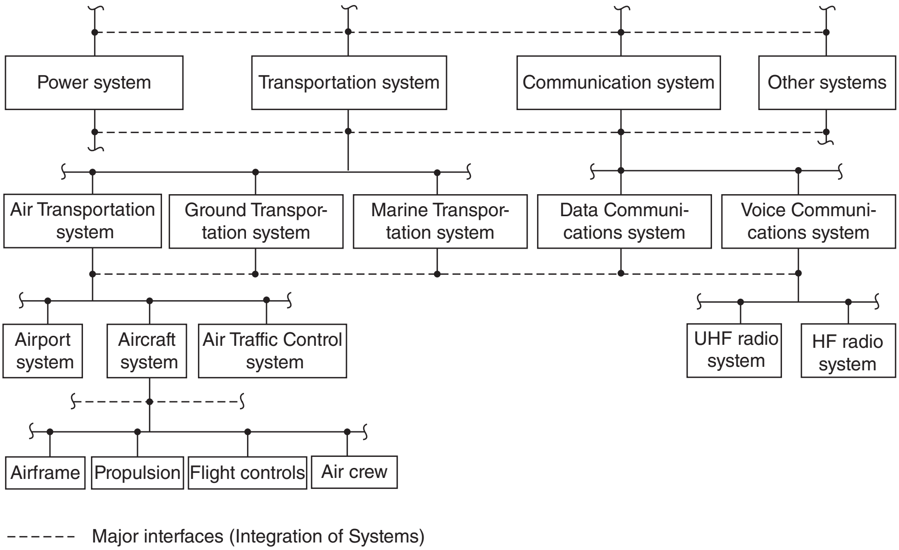

FIGURE 1.4 Multiple systems (system of systems).

interrelationships that exist throughout. In regard to the systems shown in Figure 1.4, both upward and downward impacts must be considered. Decisions pertaining to the aircraft system may have an upward impact on the air transportation system (e.g., the airline) and certainly will have a downward impact on the aircraft's airframe, propulsion and so on. For example, the maintenance support infrastructure for the aircraft system may have to be compatible with the maintenance concept specified for the higher-level air transportation system (e.g., the airline). In addition, this concept may also be imposed as a constraint in the design of the airframe and its components. In any event, these interaction effects may be significant and must be addressed.

1.2 THE CURRENT ENVIRONMENT: SOME CHALLENGES

Having a good understanding of the overall *environment*, and some of the challenges ahead, is certainly a prerequisite to the successful implementation of system engineering principles and concepts. Although individual perceptions will differ, depending on what various individuals observe, there are a few trends that appear to be significant. These trends, as summarized here and illustrated in Figure 1.5, are all interrelated and need to be addressed in total, and as an integrated set in determining the ultimate requirements for systems and in the implementation of the system engineering process:

1. *Constantly changing requirements*. The requirements for new systems are frequently changing because of the dynamic conditions worldwide, changes in mission thrusts and priorities, and the continuous introduction of new technologies. Further, it is often difficult to define the "real" requirements for new

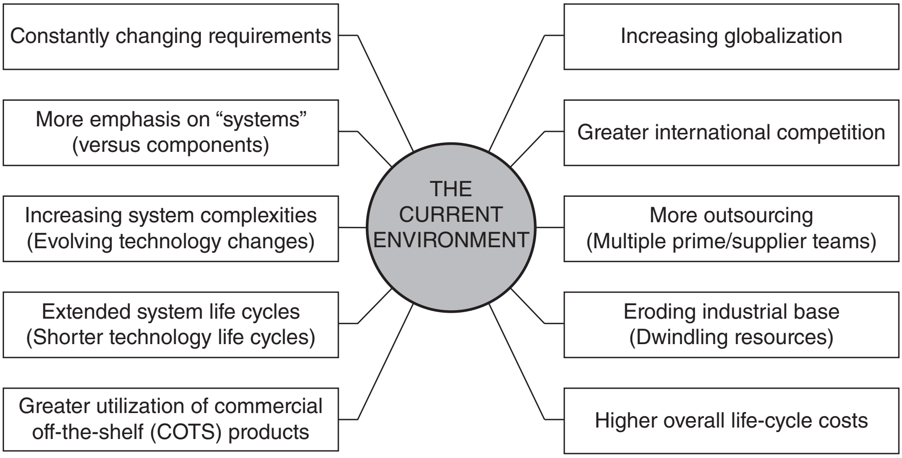

FIGURE 1.5 The current environment.

systems because of the lack of a good definition of the problem(s) to be solved and the subsequent lack of good communications between the ultimate user and the system developer from the beginning.

2. *More emphasis on systems*. There is a greater degree of emphasis on total *systems* versus the *components* of a system. One must look at the system in total, and throughout its entire life cycle, to ensure that the functions that need to be performed are being accomplished in an effective and efficient manner. At the same time, components need to be addressed within the context of some overall system configuration.
3. *Increasing system complexities*. It appears that the structures of many systems are becoming more complex with the introduction of evolving new technologies. Further, the interaction effects between different systems, within a higher-level SOS configuration, often lead to added complexities. It will be necessary to design systems so that changes can be incorporated quickly, efficiently, and without causing a significant impact on the overall configuration of the system. An *open-architecture* approach in design will be required.
4. *Extended system life cycles—shorter technology life cycles*. The life cycles of many of the systems in use today are being extended for one reason or another while, at the same time, the life cycles of most technologies are relatively much shorter. It will be necessary to design systems (with an *open-architecture* approach in mind) so that the incorporation of a new technology can be accomplished easily and efficiently (this trend, of course, closely relates to item 3).
5. *Greater utilization of commercial off-the-shelf (COTS) products and hardware-software intellectual property (IP)*. With current goals pertaining to lower initial costs and shorter and more efficient procurement and acquisition cycles, there has been a greater emphasis on the utilization of best

commercial practices, processes, COTS equipment, and hardware-software IP. As a result, there is a greater need for a good definition of requirements from the beginning, and there is a greater emphasis on the design of systems (and their major subsystems) versus the design of components.

6. *Increasing globalization*. The "world is becoming smaller" (as they say), and there is more trading and dependency on different countries (and manufacturers) throughout the world than ever before. This trend, of course, is being facilitated through the introduction of rapid and improved communications practices, the availability of quicker and more efficient packaging and transportation methods, the application of electronic commerce (EC) methods for expediting procurement and related processes, and so on. Design team collaboration is a critical element in successful system development.
7. *Greater international competition*. Along with the noted trend toward increasing globalization, there is more international competition than ever before. This, of course, is facilitated not only through improvements in communications and transportation methods, but through the greater utilization of COTS items (and hardware-software IP), and the establishment of effective partnerships worldwide.
8. *More outsourcing*. There is more outsourcing and procurement of COTS items (equipment, hardware, software, processes, IP, services) from external sources of supply than ever before. Thus, there are more suppliers associated with any given program. This trend, in turn, requires greater emphasis on the early definition and allocation of system-level requirements, the development of a good and complete set of specifications, and a closely coordinated and integrated level of activity throughout the system development and acquisition process.
9. *Eroding industrial base*. The aforementioned trends (increasing globalization, more outsourcing, and greater international competition), combined with some decline in available resources worldwide, have resulted in a decrease in the number of available manufacturers of many products. In the design of systems, it is necessary to take care to select and utilize components for which there are stable and reliable sources of supply for at least the duration of the life cycle for the system in question. The supply-chain requirements for each major system are increasing, internationally.
10. *Higher overall life-cycle costs*. In general, experience indicates that the life-cycle costs of many of the systems in use today are increasing. Although a great deal of emphasis has been placed on minimizing the costs associated with the procurement and acquisition of systems, little attention has been paid to the costs of system operation and support. In the design of systems, it is important to view *all* decisions in the context of *total* cost if one is to properly assess the risks associated with the decision in question.

Although these and related trends have evolved over time and have had a direct impact on our day-to-day activities, we often tend to ignore some of the changes that have taken place and continue with a business-as-usual approach by implementing

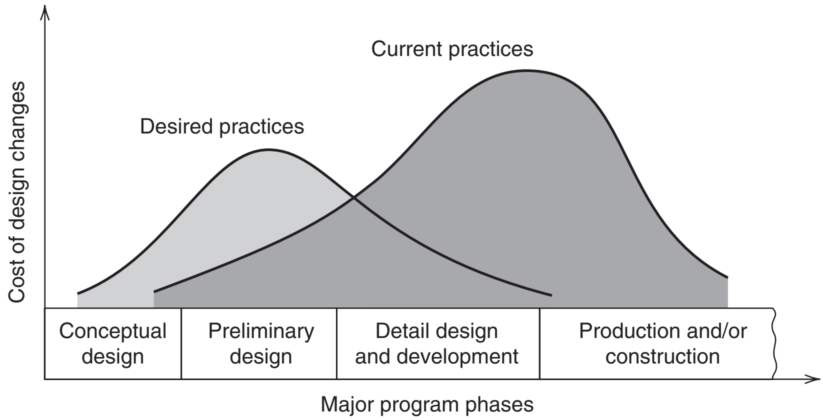

FIGURE 1.6 The cost impact due to changes.

some past practices that ultimately have had a negative impact on the systems we have developed. From past experience, it is clear that many of the problems noted have been the direct result of not applying a *disciplined* “systems approach” to meet the desired objectives. The overall requirements for the system in question were not well defined from the beginning; the perspective in terms of responding to a consumer (user) need was a relatively “short-term” focus, and, in many instances, the approach followed was to *design it now and fix it later!* In essence, the system design and development process has suffered somewhat from a lack of good early planning and the subsequent definition and allocation of requirements in a complete and methodical manner.

In regard to *requirements*, the trend has been to keep things “loose” in the beginning by developing a system-level specification that is very general (vague) in content, providing an opportunity for the introduction of the “latest and greatest” changes in technology developments just prior to going into the construction/production stage. Traditionally, many engineers do not want to be forced into design-related commitments any earlier than necessary, and the basis for defining lower-level requirements is often very “fluid” from the beginning. Thus, there are a lot of last-minute changes in design, and many of these late changes are introduced in haste and without concern for any form of configuration management. Furthermore, sometimes these changes are actually incorporated at a later stage. In any event, the introduction of late changes and the lack of good configuration control from the beginning can be rather costly. Figure 1.6 provides a comparison of the cost impact from the incorporation of changes early in the design process versus those incorporated later.[9]

These and related past practices have had a great impact on the overall cost of systems. In fact, in recent years and for many systems, there has been an *imbalance* between the “cost” side of the spectrum and the “effectiveness” side, as illustrated

[9]Source: B. S. Blanchard, D. Verma, and E. Peterson, *Maintainability: A Key to Effective Serviceability and Maintenance Management* (New York: John Wiley & Sons, Inc. 1995).

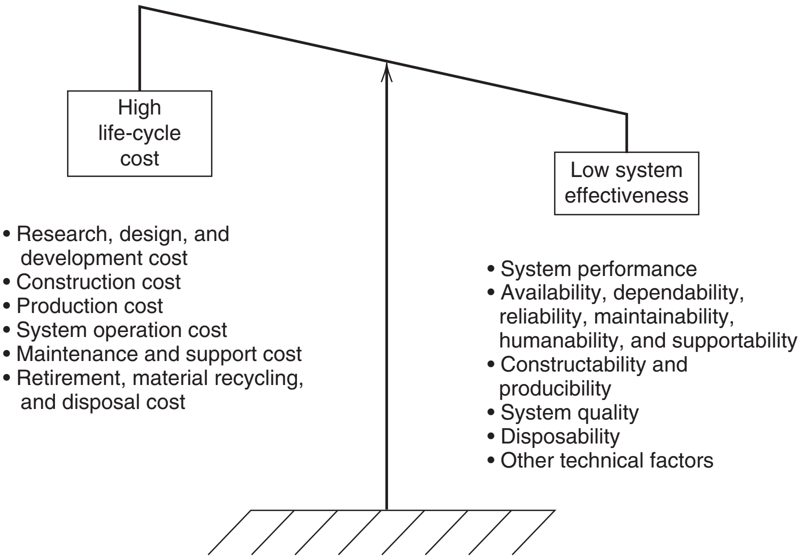

FIGURE 1.7 The imbalance between system cost and effectiveness factors.

in Figure 1.7. Many systems have grown in complexity, and although there has been an increase in emphasis on some *performance* factors, the resultant reliability and quality have been decreasing. At the same time, the overall long-term costs have been increasing. Thus, there is a need to provide the proper balance in the development of systems in the future, as any specific design decision will have an impact on both sides of the balance and the interaction effects can be significant.

In addressing the aspect of economics, one often finds that there is a lack of total cost visibility, as illustrated by the "iceberg" in Figure 1.8. For many systems, design and development costs (and production costs) are relatively well known; however, the costs associated with the sustaining management of IP, system operation and maintenance support, and the like are somewhat hidden. In essence, the design community has been successful in dealing with the short-term aspects of cost but has not been very responsive to the long-term effects. At the same time, experience has indicated that a large segment of the life-cycle cost for a given system is associated with the operational and maintenance support activities accomplished downstream in the life cycle (e.g., up to 75% of the total cost in some instances). Thus, although our budgeting and current practices tend to heed the short-term cost impacts, we cannot adequately assess the risks associated with the ongoing decision-making process without projecting these decisions in the context of the entire system life cycle. In other words, we may wish to make a design decision based on some short-term aspect of cost, but it is important to address the *life-cycle* implications prior to finalizing the decision.

Moreover, in considering cause-and-effect relationships, it has been determined that a major portion of the projected life-cycle cost for a given system stems from the consequences of decisions made during the early stages of advance planning and

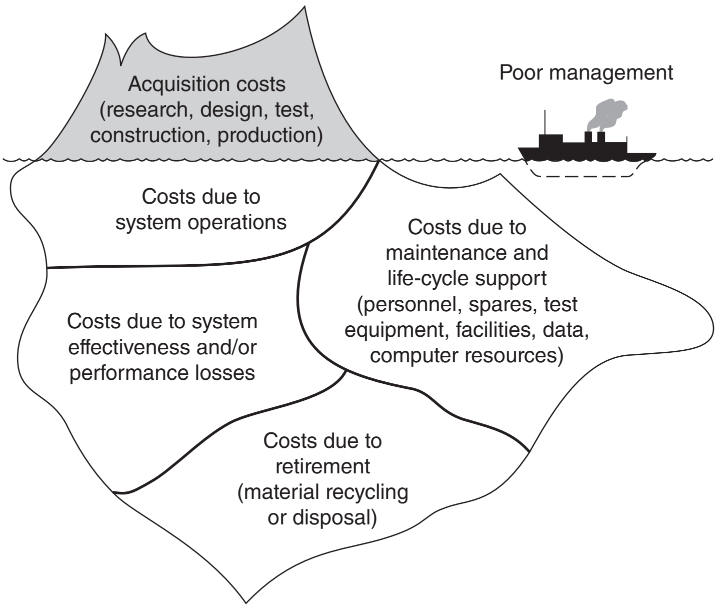

FIGURE 1.8 Total cost visibility.

system conceptual design. Such decisions, which can have a significant impact on downstream costs, relate to the definition of operational requirements (the number of consumer sites assumed, the selection of a given mission profile, specified utilization factors, the assumed life cycle), maintenance and support policies (two versus three levels of maintenance, levels of repair, in-house versus third-party maintenance support), allocations associated with manual versus automation applications, equipment packaging schemes and diagnostic routines, hardware versus software applications, the selection of materials, the selection of a manufacturing process, whether a COTS item (or hardware-software IP) should be selected versus pursuing a new design approach, and so on. In Figure 1.9 it can be seen that the greatest opportunity for influencing life-cycle cost can be realized in the early stages of system design and development. In other words, early design decisions should be evaluated on the basis of *total life-cycle cost*.

Given the environment of constantly changing requirements, greater utilization of COTS items and hardware-software IP, increased globalization and more outsourcing, and so forth, there is an ever-increasing need to review our current practices for bringing new systems into being (or for modifying existing systems). A highly disciplined approach must be pursued in the design and development of new systems, with the objective of providing the consumer (user) with a high-quality system that is cost-effective, considering the proper balance among the factors identified in Figure 1.7. In addition, there must be more emphasis on *systems*, from a life-cycle perspective, which must be established from the beginning, as

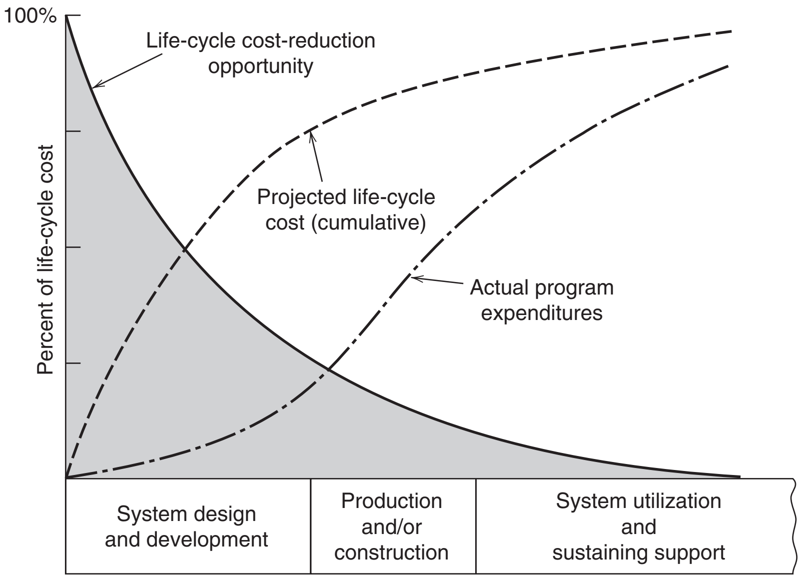

FIGURE 1.9 Commitment of life-cycle cost.

illustrated in Figure 1.9. For systems already in use, it is critical that we establish a systematic approach to reviewing their requirements and subsequently implementing an effective *evaluation and continuous product/process improvement methodology*. In any event, the current environment, as highlighted herein, is certainly conducive to the implementation of the principles and concepts discussed throughout this text.

1.3 THE NEED FOR SYSTEM ENGINEERING

The trends and concerns conveyed in Section 1.2 are only a sample of the major issues that need to be addressed. The challenge is to be more effective and efficient in the development and acquisition of new systems (i.e., any time that there is a newly identified need and a new system requirement has been established), as well as in the operation and support of those systems already in use. This can be accomplished through the proper implementation of *system engineering* concepts, principles, and methods.

In exploring topics such as *systems*, *system engineering*, *system analysis*, and the like, one will find a variety of approaches in existence. The specific terms may be defined somewhat differently, depending on individual backgrounds, experiences, and on the organizational interests of practitioners in the field. Thus, with the objective of providing some clarification relative to the material throughout this text, it seems appropriate to consider a few additional concepts and definitions at this point.

1.3.1 The System Life Cycle

As shown in Figure 1.10, the *life cycle* includes the entire spectrum of activity for a given system, commencing with the identification of need and extending through system design and development, production and/or construction, operational use and sustaining maintenance and support, and system retirement and material disposal. As the activities in each phase interact with the activities in other phases, it is essential to consider the overall life cycle in addressing system-level issues, particularly if one is to properly assess the risks associated with the decision-making process throughout.

Although the life-cycle phases conveyed in Figure 1.10 reflect a more generic sequential approach, the specific activities (and the duration of each) may vary somewhat, depending on the nature, complexity, and purpose of the system. Needs may change, obsolescence may occur, and the levels of activity may be different, depending on the type of system and where it fits in the overall hierarchical structure of activities and events. In addition, the various phases of activity may overlap somewhat, as illustrated in the two examples presented in Figure 1.11.

Figure 1.11 shows how an airplane, a ground transportation vehicle, or an electronic device may progress through conceptual design, preliminary design, detail design, production, and so on, as reflected through the series of activities for Example A. When this example is evaluated further, the top row of activities is applicable to those elements of the system that relate directly to the accomplishment of the mission (e.g., an automobile). At the same time, there are two closely related life cycles of activity that must also be considered. The design, construction, and operation of the production capability, which can have a significant impact on the operation of the prime elements of the system, should be addressed concurrently along with the system maintenance and support activity. Further, these activities must be addressed early during the conceptual and preliminary design of those prime elements represented by the top row. Although all of these activities may be presented through an illustrated single flow, as conveyed in Figure 1.10, the breakout in Figure 1.11 is intended to emphasize the importance of addressing *all* aspects of the total system process and the various interactions that may occur.

Example B in Figure 1.11 is presented to cover the major phases associated with a manufacturing plant, a chemical processing plant, or a satellite ground tracking

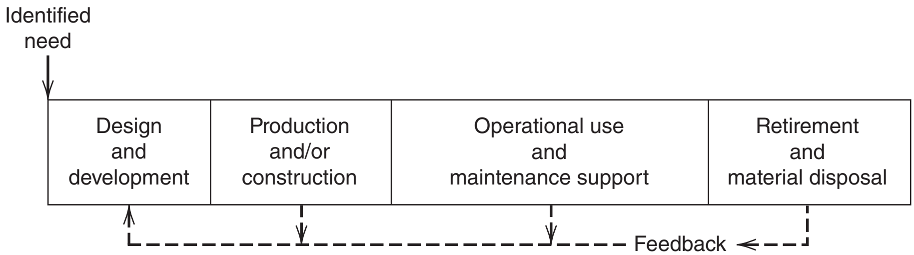

FIGURE 1.10 The system life cycle.

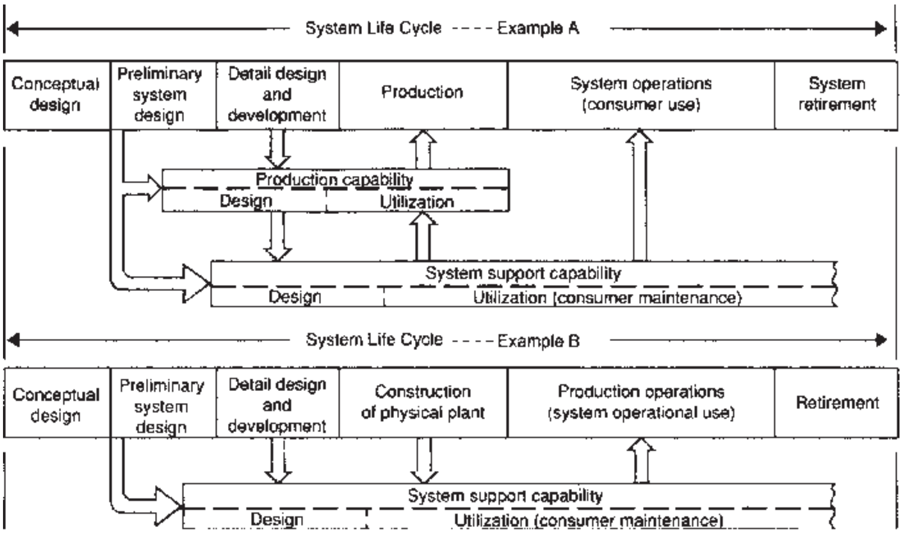

FIGURE 1.11 Examples of a system life cycles.

facility, where the construction of a "one-of-a-kind" system configuration is required. Again, the maintenance and support capability is identified separately in order to indicate degree of importance and to suggest that there are many interaction effects that must be considered.

Although there may be variations in approach, the nomenclature used, the duration of the different phases, and so on, it is still appropriate that systems be viewed in the context of their respective life cycles. This is further complicated in the SOS situation, where each of the identified systems within a given SOS structure will likely have a different life cycle. Nevertheless, a total *life-cycle approach* must be assumed in the decision-making process. The past is replete with examples in which major decisions have been made in the early stages of system acquisition based on the "short-term" only. In other words, in the design and development of a new system, the considerations for *production/construction, maintenance and support*, and/or *retirement and disposal* for that system were inadequate. These activities were considered later and, in many instances, the consequences of this 'after-the-fact' approach were costly, as discussed in Section 1.2.[10]

[10]Referring to Figure 1.11, the emphasis as presented addresses the three life cycles, including (1) the life cycle pertaining to the mission-related elements of the system, (2) the production capability, and (3) the maintenance and support capability. There is a fourth life cycle that is equally important but not highlighted in the figure, and this pertains to the design and implementation of the retirement and material recycling/disposal capability. One needs to design for producibility, design for supportability/serviceability, and design for recyclability and disposability.

1.3.2 Definition of System Engineering

System engineering may be defined in a number of ways, depending on one's background and personal experience. The inaugural issue of *Systems Engineering*, published by the International Council on Systems Engineering (INCOSE), describes a variety of approaches.[11] However, there is a basic theme throughout that deals with a top-down process, which is life-cycle-oriented, involving the integration of functions, activities, and organizations.

The International Council on Systems Engineering (INCOSE) defines it as follows:[12]

> Systems engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem. Systems engineering considers both the business and technical needs of all customers with the goal of providing a quality product that meets the user needs.

The Department of Defense (DOD) defines system engineering as follows:

> An approach to translate approved operational needs and requirements into operationally suitable blocks of systems. The approach shall consist of a top-down, iterative process of requirements analysis, functional analysis and allocation, design synthesis and verification, and system analysis and control. Systems engineering shall permeate design, manufacturing, test and evaluation, and support of the product. Systems engineering principles shall influence the balance between performance, risk, cost, and schedule.

Wikipedia defines Systems Engineering as:[13]

> An interdisciplinary field of engineering that focuses on how to design and manage complex systems over their life cycles.—Systems Engineering ensures that all likely aspects of a project or system are considered, and integrated into a whole.

More specifically:

> The systems engineering process shall:[14]

1. Transform approved operational needs and requirements into an integrated system design solution through concurrent consideration of all life-cycle needs (i.e., development, manufacturing, test and evaluation, deployment, operations, support, training, and disposal), and

[11] Inaugural Issue "System Engineering" *Journal of the International Council on Systems Engineering*. vol. 1, no. 1 (July/September 1994); INCOSE, 7670 Opportunity Road, Ste 220, San Diego, CA 92111.
[12] INCOSE-TP-2003–002–03.2.2, *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, Version 3.2.2, (San Diego, CA: INCOSE, 2011).
[13] http://en.wikipedia.org/wiki/Systems_engineering
[14] Department of Defense Regulation 5000.2R. "Mandatory Procedures for Major Defense Acquisition Programs (MDAPS) and Major Automated Information Systems (MAIS) Acquisition Programs." Chapter 5. Paragraph C5.2 (April 2002).

2. Ensure the interoperability and integration of all operational, functional, and physical interfaces. Ensure that system definition and design reflect the requirements for all system elements to include hardware, software, facilities, people, and data, and
3. Characterize and manage technical risks.

> The key systems engineering activities that should be performed are requirements analysis, functional analysis/allocation, design synthesis and verification, and system analysis and control.

A slightly different definition (preferred by the author) states that system engineering is:

> The application of scientific and engineering efforts to: (1) transform an operational need into a description of system performance parameters and a system configuration through the use of an iterative process of definition, synthesis, analysis, design, test and evaluation, and validation; (2) integrate related technical parameters and ensure the compatibility of all physical, functional, and program interfaces in a manner that optimizes the total definition and design; and (3) integrate reliability, maintainability, usability (human factors), safety, producibility, supportability, sustainability, disposability, and other such factors into a total engineering effort to meet cost, schedule, and technical performance objectives.[15]

Basically, system engineering is *good* engineering with certain designated areas of emphasis, a few of which are noted as follows:

1. A top-down approach is required, viewing the system as a *whole*. Although bottom-up engineering activities in the past have very adequately covered the design of various system components, the necessary *overview* and an understanding of how these components effectively fit together has not always been present.
2. A *life-cycle* orientation is required, addressing all phases to include system design and development, production and/or construction, distribution, operation, sustaining maintenance and support, and retirement and material phase-out. Emphasis in the past has been placed primarily on system design activities, with little (if any) consideration given to their impact on production, operations, support, and disposal.
3. A better and more complete effort is required relative to the initial *identification of system requirements*, relating these requirements to specific design goals, the development of appropriate design criteria, and the follow-on analysis effort to ensure the effectiveness of early decision making in the design process. In the past, the early "front-end" analysis effort, as applied to many new systems, has been minimal. This, in turn, has required greater individual design efforts downstream in the life cycle, many of which were not well integrated with other design activities and have required modification later on.

[15]This is a slightly modified version of the definition of systems engineering that was included in the original version of MIL-STD-499. Systems Engineering (Washington, DC: Department of Defense, July 1969).

4. An *interdisciplinary* collaborative effort (or team approach) is required throughout the system design and development process to ensure that all design objectives are met in an effective manner. This necessitates a complete understanding of the many different design disciplines and their interrelationships, particularly for large projects. Due to the global nature of system development, careful attention to the supply chain structure is also required.
5. *Interface management* is the key method for highlighting problems and for monitoring the "goodness" of the system design and integration effort. Managing the design of complex technical systems requires an understanding of many topics, including interface-related issues, resource-margin allocation, and technical performance measurement (TPM) methods.[16]

Inherent within the system engineering process is a "top-down/bottom-up" development approach, as illustrated in the "Vee-Diagram" in Figure 1.12. The emphasis throughout this text is represented by the shaded area; that is, the front-end requirements analysis activity. Traditionally, the requirements have not been well defined from the beginning, resulting in some rather extensive and costly efforts during the final integration and test activity.

Figure 1.13 presents an extension of the basic life-cycle phases shown in Figure 1.10, describing typical activities that occur in each phase, identifying various

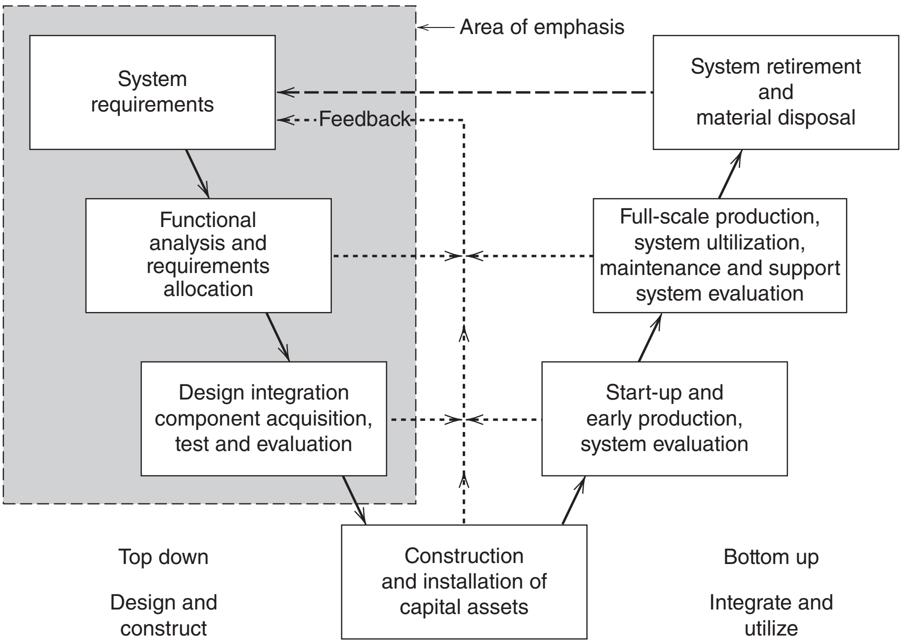

FIGURE 1.12 Top-down/bottom-up system development process.

[16]"Interface Management," *IEEE Instrumentation & Measurement Magazine*, Volume 7, Issue 1, March 2004.

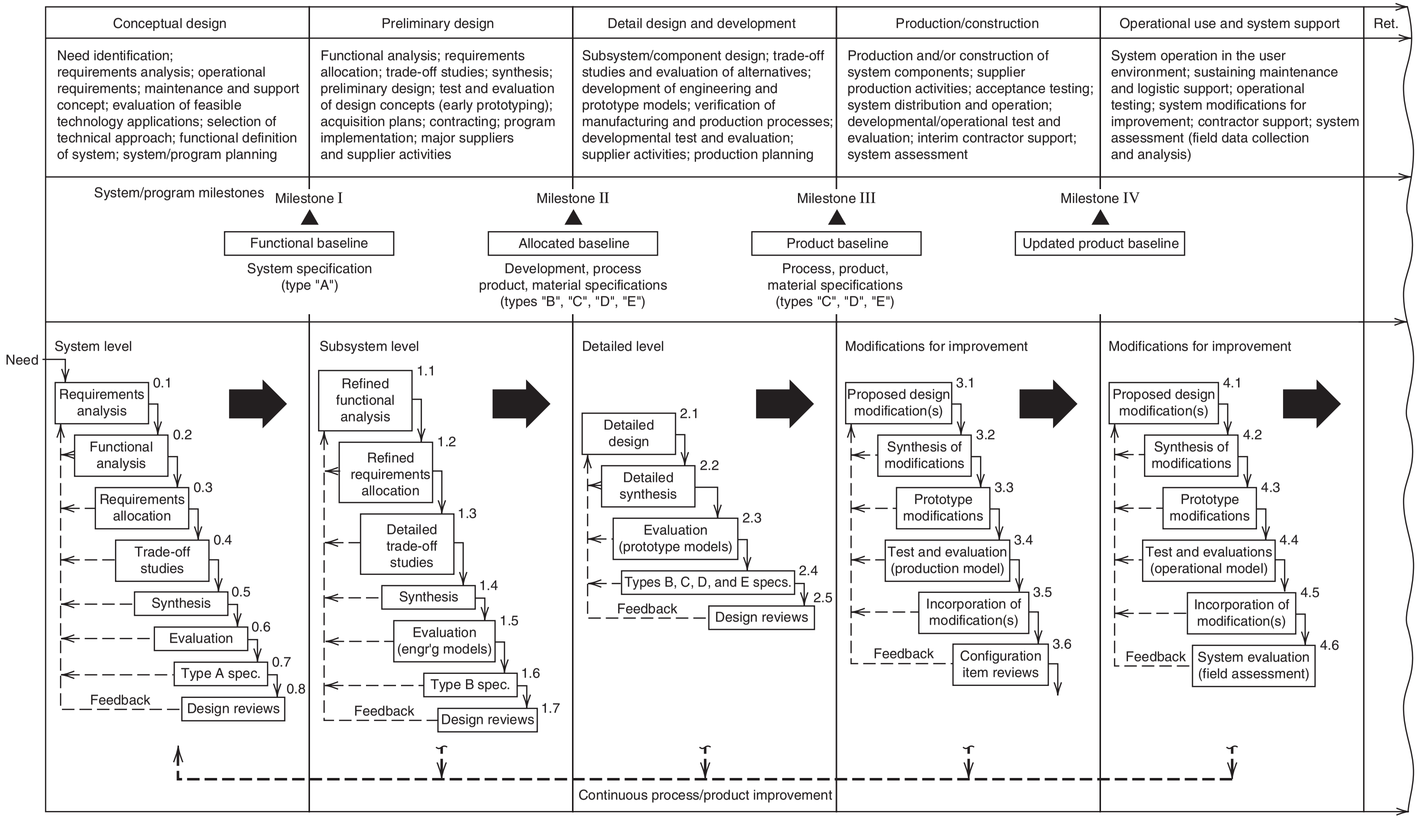

FIGURE 1.13 System engineering within the acquisition process. *Source:* B. S. Blanchard, D. Verma, and E. Peterson, *Maintainability: A Key to Effective Serviceability and Maintenance Management* (New York: John Wiley & Sons, Inc., 1995). This material is used by permission of John Wiley & Sons, Inc.

configuration *baselines* that should be established as one progresses from the initial identification of need to the development of a fully operational system, and including the iterative steps inherent within the system engineering process. Although the presentation of information in the figure may lead the reader to believe that the system acquisition process is very complex, the objective is to show this as a *process* in itself. Every time there is a newly identified *need*, there are certain steps through which a design engineer should evolve—that is, conceptual design, preliminary design, and so on. Even if the effort (in terms of the resources expended) is minimal, there is still the requirement for design activities at the *system level* and on down. The objective is to view these phase-related activities as a process within itself and to identify the baselines where the design evolves from one level of definition to the next. Tailoring the activities in Figure 1.13 to the system in question is essential for the successful implementation of the system engineering process.

The system engineering process per se includes the basic steps of requirements analysis, functional analysis, requirements allocation, design optimization and trade-offs, synthesis, evaluation, and so on (refer to blocks 0.1, 0.2, 0.3, etc., in Figure 1.13). These steps are *iterative* by nature, evolving from the system-level definition to the subsystem level, detailed level, and on down to the component. Further, these steps are not necessarily accomplished in a serial sequence, but are interactive with the appropriate *feedback* provisions at each step in the process. Although the requirements may vary somewhat from program to program, the purpose of this figure is to provide a baseline for future reference as different topics are presented throughout this text.

In Block 0.2 (Figure 1.13), the accomplishment of the *functional analysis* will lead to the identification of resources in terms of the need for hardware, software, people, facilities, data, and the like. The functional analysis identifies the "*whats*" from a requirements perspective, and this leads to the accomplishment of trade-offs and the description of the "*hows*" pertaining to the completion of functions. Figure 1.14 illustrates the identification of hardware, software, and human requirements (from the functional analysis), and the subsequent life cycles associated with the development of each of these resources. One of the goals of system engineering is to justify these resource requirements through a top-down approach and to ensure the proper development of each through a fully integrated system as one progresses through the design of its various elements. Measuring and monitoring the day-to-day interaction effects between the three life cycles is critical.

Figure 1.15 presents the system engineering approach from a different perspective. As one progresses through the life cycle, there is a need to ensure the full traceability of requirements from the system level and on down to the component. As *technical performance measures* (TPMs), or the applicable *metrics*, are established for the system, these measures must be allocated or apportioned to the next level, appropriate *design criteria* are identified, and these criteria must be reflected and supportive from the top down. Further, the appropriate methods/tools must be applied in the design process to ensure that the overall objectives of the system are met. Inherent within the system engineering process is the need to ensure that this traceability is maintained and to cause the integration of the appropriate techniques/methods/tools to facilitate the development process in an effective and efficient manner.

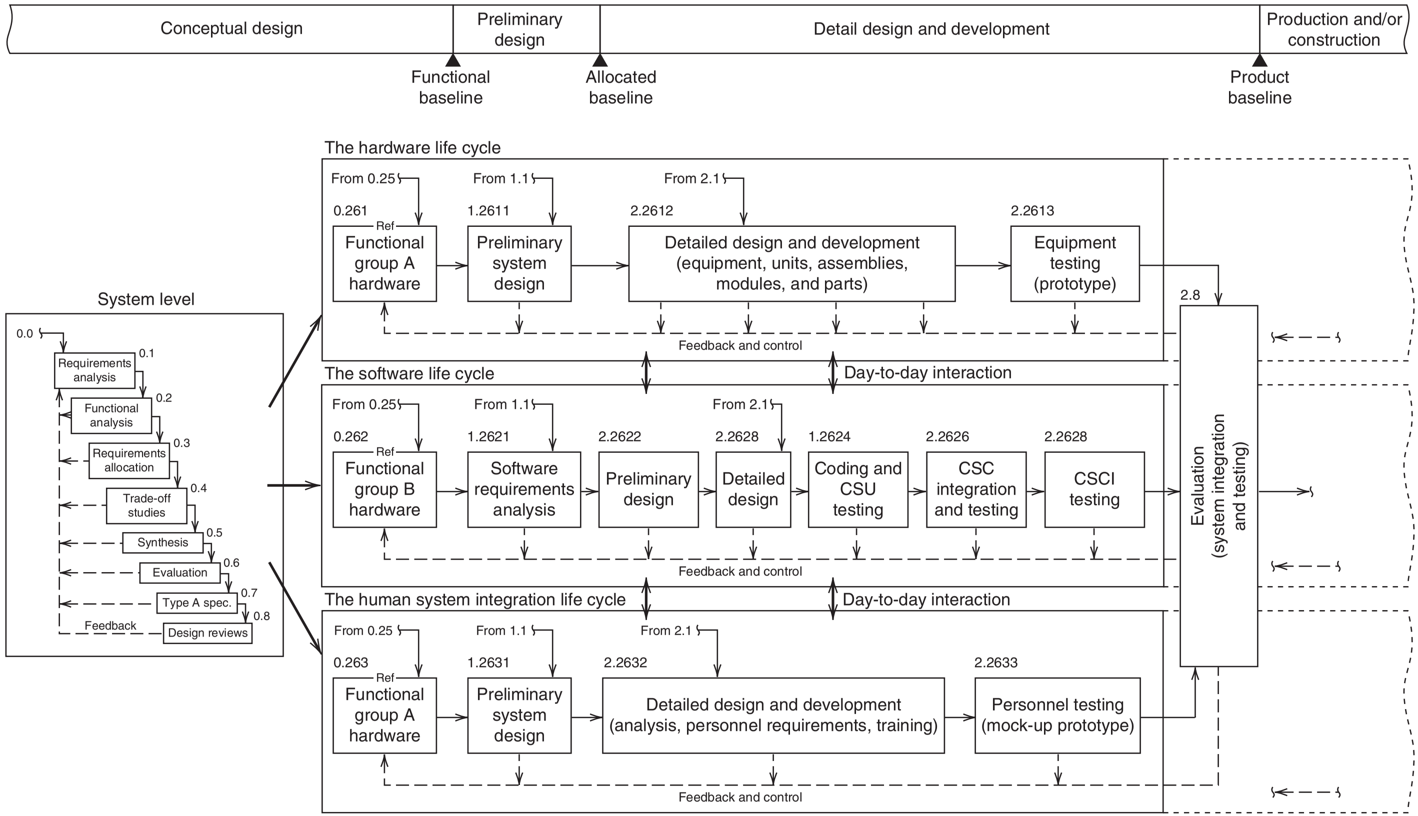

FIGURE 1.14 The integration of the hardware, software, and human life cycles.

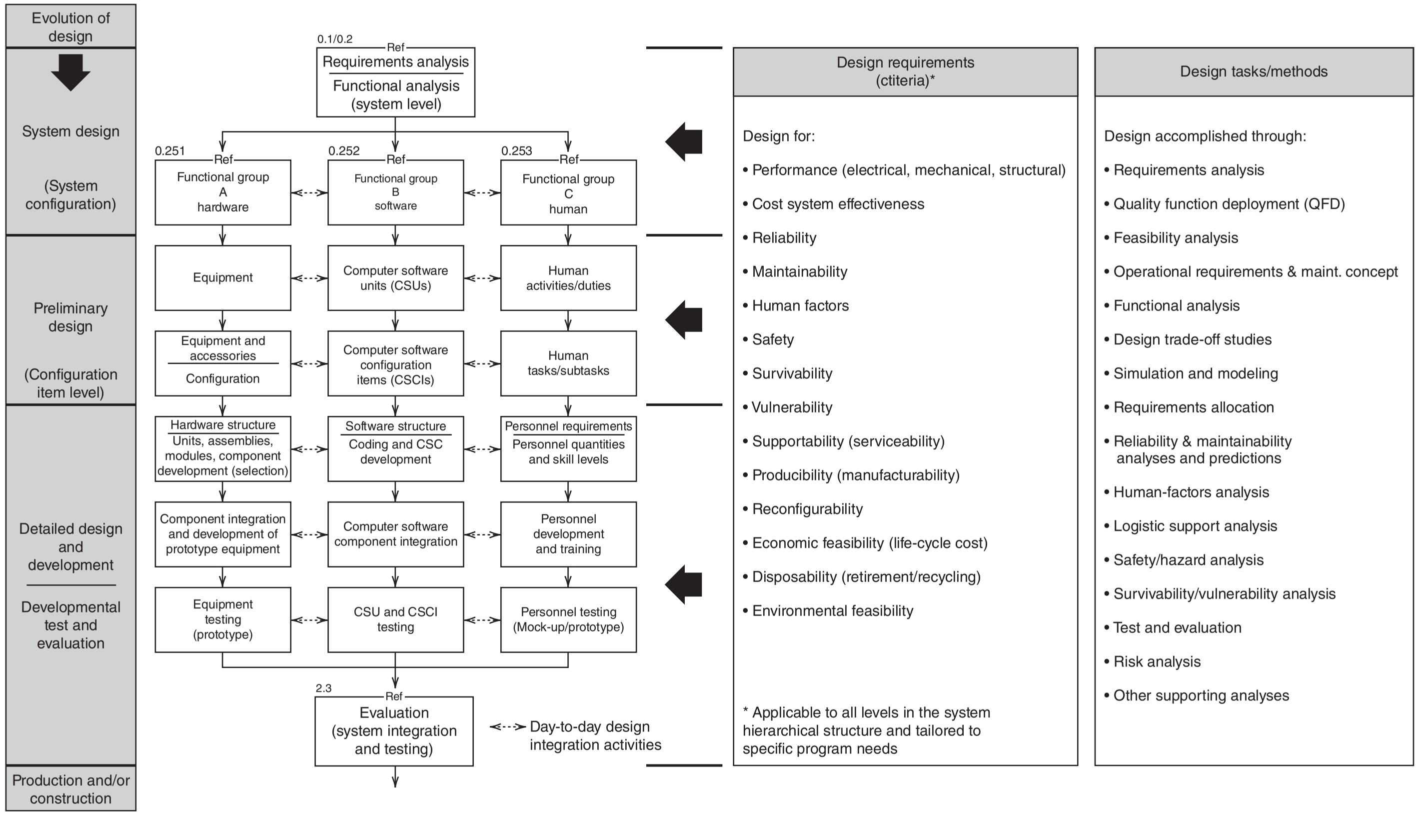

FIGURE 1.15 The top-down traceability of requirements.

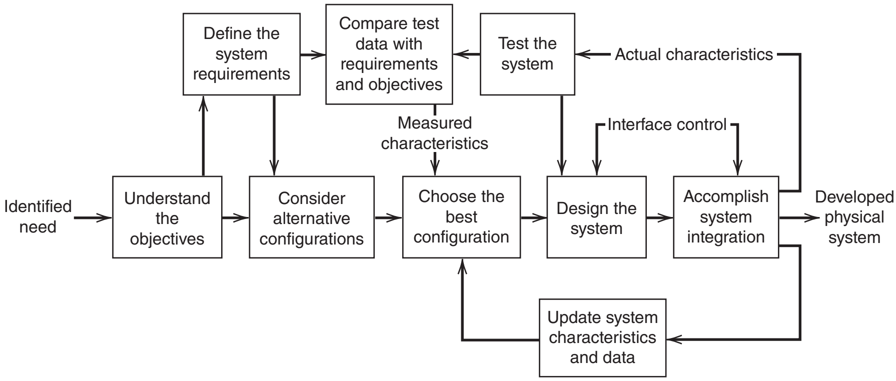

FIGURE 1.16 Feedback in the system engineering process.

In summary, the system engineering process is continuous, iterative, and incorporates the necessary feedback provisions to ensure convergence. Figure 1.16 illustrates the *feedback* capability that must be built into the process, applied at the system level, to the subsystem level, and so on, as illustrated in Figure 1.13.

System engineering per se is not considered an engineering discipline in the same sense as civil engineering, mechanical engineering, reliability engineering, or any other design specialty area. Actually, system engineering involves efforts pertaining to the overall design and development process employed in the evolution of a system from the point when a need is first identified, through production and/or construction and the ultimate installation of that system for consumer use. The objective is to meet the requirements of the consumer in an effective and efficient manner. The system engineering process is covered further in Chapter 2. Finally, the concepts and principles associated with system engineering are not necessarily new or novel. A review of the literature in Appendix F indicates that many of the principles identified herein were being promoted back in the 1950s and early 1960s. However, in many instances, the system engineering process has not been implemented very well (if at all). Yet, at this point in time, there is a need to emphasize these concepts more than ever.

1.3.3 Requirements for System Engineering

A primary objective in the implementation of system engineering is to evolve through the *process* illustrated in Figure 1.13, stemming from the initial identification of a *problem* and the subsequent definition of a *need* for a new system to accomplish certain specific functions that are not currently being performed. Every time that a new system requirement is identified, it is essential that we progress through a series of steps that will logically and effectively lead to an end solution. Hopefully, following the general steps described in Section 1.3.2 will facilitate this objective. This is not to indicate that the process in Figure 1.13 is complex by nature; it is the *thought process* (i.e., the "way of thinking") that is important. Whether one is dealing with a large complex system or a relatively small system, the same basic process should be followed. In other words, there is a requirement for the implementation of system engineering (and its process) for any type of system, whether large or small. This, in turn, should enable the development of a system that is both timely and cost-effective in its operation and support.

Traditionally, a designer will start with a bunch of existing system components, determine just how they might be improved to respond to a new system requirement, add some new items, modify these, and basically assume a trial-by-error approach in arriving at some configuration that will do something! In other words, only a bottom-up approach has been pursued which, in turn, has resulted in a rather costly outcome. The requirement here is to follow a more *top-down/bottom-up approach* as shown in Figure 1.12.

1.3.4 System Architecture

In general, systems are composed of many different interacting elements. Although each is unique in itself and has its own capabilities and characteristics, the composite of these must be arranged and integrated into some framework intended to accomplish the required function(s) or mission(s); that is, the ultimate system configuration. This framework of elements represents the system's *architecture*. More specifically, *architecture* can be defined as follows:[17]

> The fundamental organization of a system, embodied in its components, their relationships to each other and to the environment, and the principles guiding its design and evolution.

An architecture deals with a top-level description of system structure (configuration), its operational interfaces, anticipated utilization profiles (mission scenarios), and the environment within which it is to operate; it then describes how these various requirements for the system interact. This, in turn, leads into a description of the *functional architecture*, which evolves from the functional analysis and its description of the system in "functional" terms. From this analysis, and through the requirements allocation process and the definition of the various resource requirements necessary for the system to accomplish its mission, the system's *physical architecture* is defined. Through application of this process, one is able to evolve from the *whats* to the *hows*.

Referring to Figure 1.13, system architecture is basically described at the "system" level through the requirements analysis (system operational requirements and the maintenance and support concept), functional analysis, requirements allocation, trade-off analysis, and design synthesis (the steps shown by blocks 0.1 through 0.5).[18]

1.3.5 System Science

Often, in addressing the subject of system engineering, one uses the terms *system science* and *system engineering* interchangeably. For the purposes of this text, *system*

[17]IEEE 1471–2000, IEEE Recommended Practice for Architectural Description of Software-Intensive Systems (New York: IEEE), 2000.

[18]Two good references that include coverage of architecting, architecture, system architecture, etc., are: (1) E. Rechtin and M. Maier. *Systems Architecting: Creating and Building Complex Systems*, 2nd Ed. (Boca Raton, FL: CRC Press 2000); and (2) L. Bellagamba. *Systems Engineering and Architecting: Creating Formal Requirements* (CRC Press, 2012).

science deals primarily with the observation, identification, description, experimental investigation, and theoretical explanation of facts, physical laws, inter relationships, and so on, associated with natural phenomena. Science deals with basic concepts and principles that help to explain how the physical world behaves. In the sense that they are applied sciences, the disciplines of biology, chemistry, and physics cover many of these relationships. In any event, system engineering includes the application of scientific principles throughout the system design and development process.[19]

1.3.6 System Analysis

Inherent within the system engineering process is an ongoing analytical effort. In a somewhat puristic sense, analysis refers to a separation of the whole into its component parts, an examination of these parts and their interrelationships, and a follow-on decision relative to a future course of action.

More specifically, throughout system design and development there are many different alternatives (or trade-offs) requiring an evaluation effort in some form. For instance, there are alternative system operational scenarios, alternative maintenance and support concepts, alternative equipment packaging schemes, alternative diagnostic routines, alternative manual versus automation applications, and so on. The process of investigating these alternatives, and the evaluation of each in terms of certain criteria, constitute an ongoing analytical effort.

To accomplish this activity effectively, the engineer (or analyst) relies on the use of available analytical techniques/tools to include operations research methods such as simulation, linear and dynamic programming, integer programming, optimization (constrained and unconstrained), and queuing theory to help solve problems. Further, mathematical models are used to help facilitate the quantitative analysis process.

More recently, the system design and development process has been facilitated through the application of numerous standalone digital and related models to address various detailed aspects of system design. These individual models are combined and integrated into an overall model for the system as a whole, describing system characteristics and interrelationships. The utilization of these models throughout the process illustrated in Figure 1.13 is accomplished by applying the concepts associated with *Model-Based Systems Engineering (MBSE)*. MBSE emphasizes the application of rigorous analysis methods applied throughout the overall system design and development process.

In essence, *system analysis* includes that ongoing analytical process of evaluating various system design alternatives, employing the application of computerized mathematical models and associated analytical tools as appropriate. Analytical methods and models are discussed further in Chapter 4.[20]

[19]Systems science is a major subject by itself, and adequate coverage is not included (or possible) herein. Two excellent references are: (1) G. M. Sandquist. *Introduction to System Science* (Pearson Prentice Hall, 1985); and (2) R. L. Ackoff, S. Gupta, and J. Minas. *Scientific Method: Optimizing Applied Research Decisions* (John Wiley & Sons, 1962).

[20]System analysis is covered further in a number of references listed in the Bibliography in Appendix F. Some of the operations research tools utilized in accomplishing system analyses are included in B. S. Blanchard and W. J. Fabrycky. *Systems Engineering and Analysis*, 5th Ed, Pearson Prentice Hall, 2011.

1.3.7 Some Additional System Models

In the early 1980s, when the makeup of systems became more *software intensive*, there were a number of models developed with the objective of portraying the system life cycle. The *waterfall model* is probably the oldest and most widely used of the system development models in this category at the time.[21] This model, shown in Figure 1.17, is based on a top-down approach for software development and includes the steps of initiation, requirements analysis, design, testing, and so on.

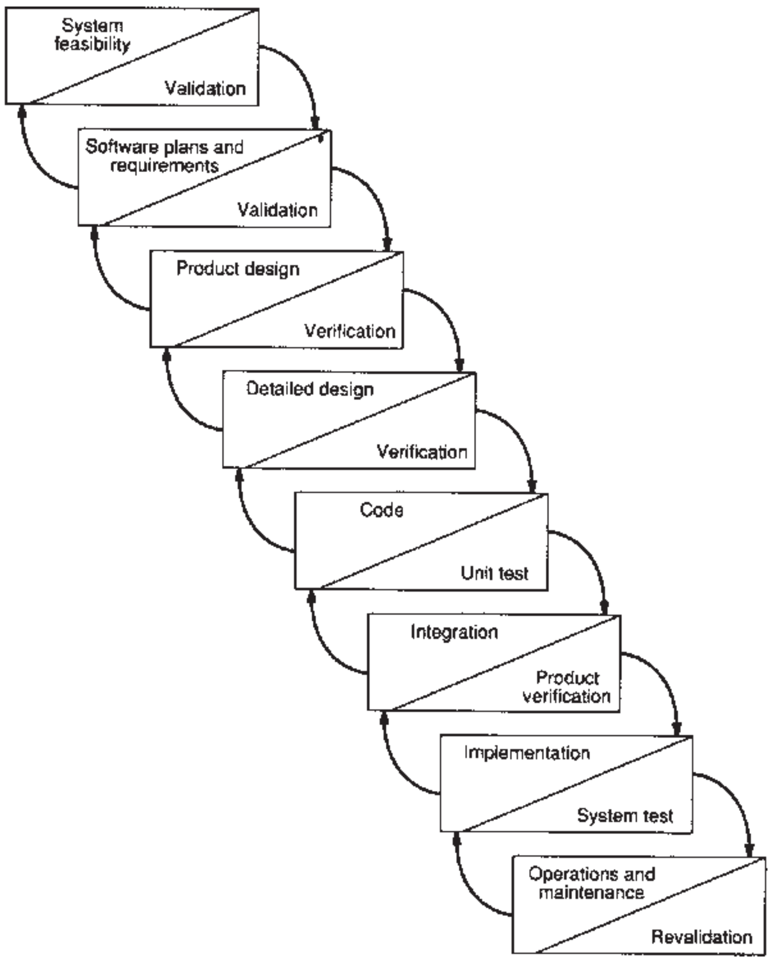

FIGURE 1.17 Waterfall model of the software life cycle.

[21] B. W. Boehm. *Software Engineering Economics* (Pearson Prentice Hall, 1981, p. 36).

Often, in its implementation, the steps were viewed as being relatively independent from one another and were to be executed in a strict sequence, and the feedback effects were not emphasized. In addition, the required interfaces with the other elements of the system (e.g., hardware, the human factor, facilities, data) were not usually considered.

Recently, the application of *agile engineering* methods has resulted in a more iterative approach in the development of software-based systems. The emphasis has been on continuous feedback, stressing individuals and interactions over processes and tools, working software in lieu of comprehensive documentation, customer collaboration, and rapidly responsive to the incorporation of in process changes as appropriate. Agile engineering is discussed further in Section 3.4.2.[22]

In the mid-1980s, a generic *spiral model* was developed for software-intensive systems.[23] In this method, the analyst continually examines objectives, strategies, design alternatives, and validation methods. System development results through several iterations of this model. Figure 1.18 illustrates a modified version of the original generic approach, evolving from a prototype model. Note that rapid prototyping is used in each cycle and that the model emphasizes risk analysis. This approach is particularly useful in high-risk developments because design sometimes evolves as detailed requirements emerge.

The *Vee model*, introduced in the early 1990s, reflects a top-down and bottom-up approach to system development. In Figure 1.19, the left side of the Vee represents the evolution of user requirements into preliminary and detail design, and the right side represents the integration and verification of system components through subsystem and system testing. This model most nearly reflects the approach conveyed in Figure 1.12 (Section 1.3.2).

Figure 1.20 represents an extension of the Vee model concept. Of particular note is the interface between the system and the software subsystem. Quite often, individuals refer to *software systems*. Although software may be predominant within the structure of a system, it is *not* the "system" per se. It does not fulfill a functional requirement by itself. Software requirements are identified through functional analysis and are subsequently developed through the steps illustrated in Figure 1.13. Figure 1.20 emphasizes that there are system engineering activities that lead into the software development process.

Numerous models have been introduced through the past few decades, with the objective of providing a logical approach to the overall process of system design and development. The few identified here are only representative of the total population. Most of these models are directed primarily to the system acquisition process only

[22]Refer to "Agile Engineering" by Adam Beaver, 2014 – http://mccadedesign.com/bsb/

[23]The generic spiral model was presented by B. W. Boehm, "A Spiral Model of Software Development," in *Software Engineering Project Management* by R. H. Thayer and M. Dorfman (IEEE Computer Society Press, 1988). This was modified in Figure 1.18 and is included in A. P. Sage, *Systems Engineering* (John Wiley & Sons, 1992, pp. 53–54).

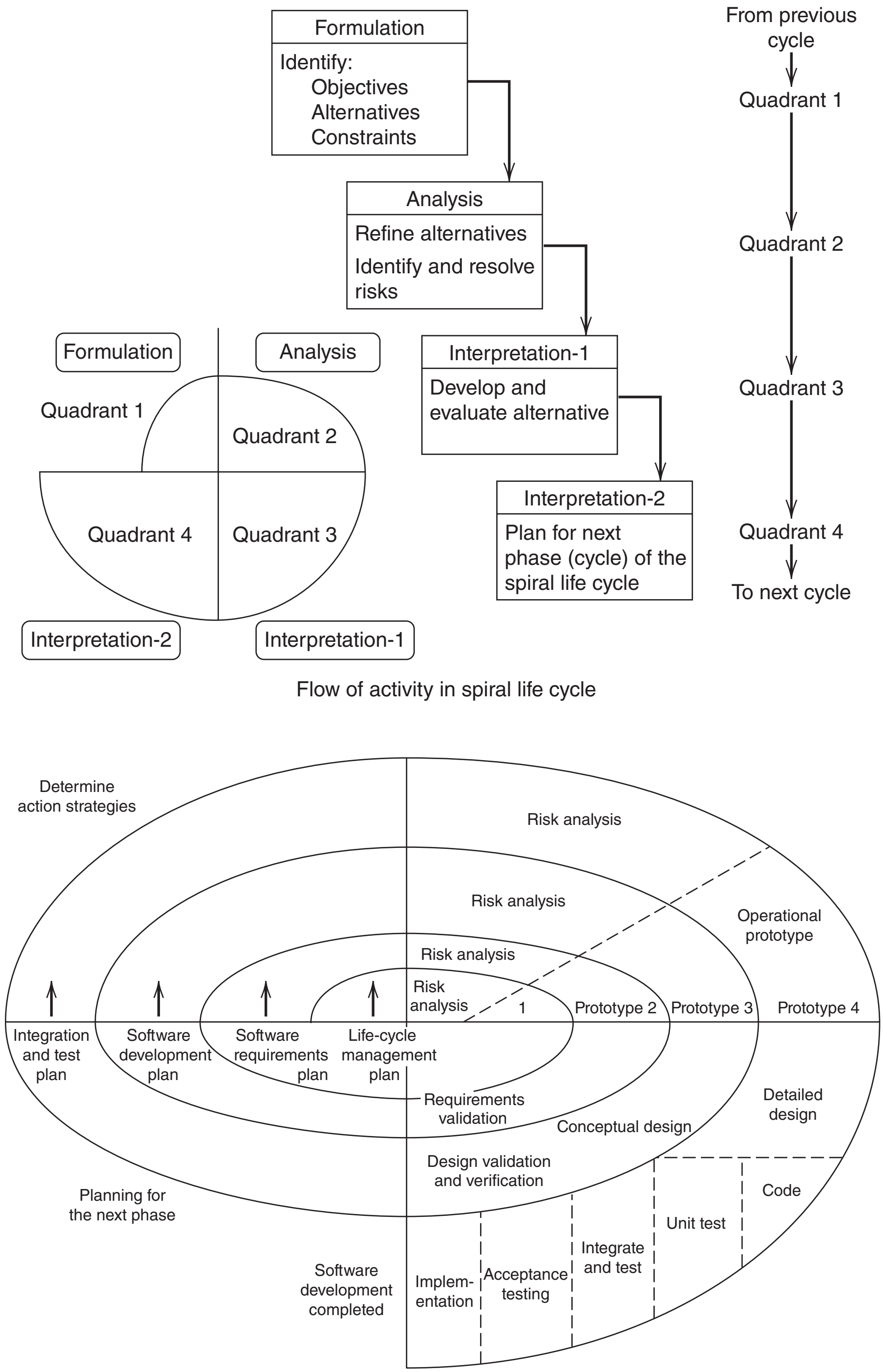

FIGURE 1.18 The spiral model for the software life cycle. *Source:* A.P. Sage, *Systems Engineering* (New York: John Wiley & Sons, Inc., 1992). This material is used with permission.

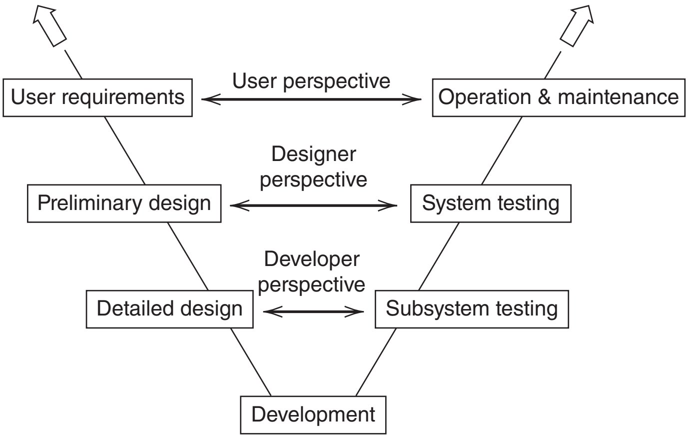

FIGURE 1.19 Generic "Vee" developmental model.

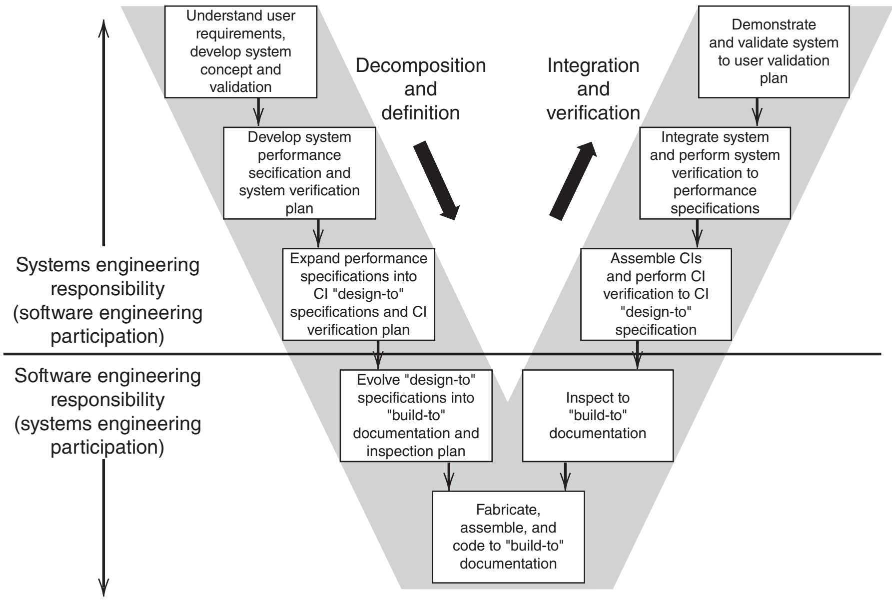

FIGURE 1.20 The systems versus software engineering boundary. *Source:* B. G. Downward, "A Brave New World: Molding Systems and Software Engineering," *Proceedings of the Symposium of the International Council on Systems Engineering* (Seattle, WA: INCOSE, 1991), 157.

and/or to some element of the system (such as software); hence, they lack a certain degree of completeness. If implemented properly, they are excellent in terms of accomplishing their intended objectives. However, it should be recognized that their application may be limited unless utilized within the broader spectrum of system engineering described in Section 1.2.5.[24]

1.3.8 System Engineering in the Life Cycle (Some Applications)

The system engineering process is applicable in all phases of the life cycle, as illustrated in Figure 1.13. In the early stages of conceptual design, the emphasis is on understanding the true needs of the consumer (user) and in developing the actual requirements for the system. These requirements, which constitute the baseline that needs to be established from the beginning, must be traceable from the top and on down to the component level as necessary. This top-down approach (with the appropriate feedback incorporated), reflected in the left side of the Vee diagram in Figure 1.12, is critical for the successful implementation of a system engineering program. It is the establishment of these early requirements that has a great impact on the ultimate life-cycle cost for a given system (see Figure 1.9).

Given the basic requirements, the emphasis then shifts to an iterative process of synthesis, analysis, design optimization, and validation. Trade-off studies are conducted, with the objective of providing a well-balanced system design. There are many different design objectives that must be met, some of which may be somewhat conflicting, and the role of system engineering is to identify, prioritize, integrate, and to cause the development of a system configuration that will meet all customer requirements in a timely, effective, and efficient manner. Such an ultimate configuration must consider the system in total to include the development of the production, maintenance and support, and retirement/material recycling capabilities, as shown in Figure 1.11.

System engineering activities continue through the construction and/or production phase to ensure that the designed system configuration is compliant with the initially specified requirements. Next, there is an ongoing and iterative process of *assessment* (and validation) throughout the operational use and maintenance support phase, and during the system retirement and material recycling stage. This assessment, with the proper feedback, is important to ensure not only that the initially specified system requirements being met, but also that any changes in requirements that take place in the user environment are properly reflected back into the design process (through redesign, reengineering, etc.). In other words, there is a continuous product/process improvement feedback loop, included at the bottom in Figure 1.13, which is critical in the implementation of system engineering.

[24]There are numerous other models, including prototype models, the Sashimi Model, the Scrum Model, the Handcuff Model, the Hollywood Model, the Evolutionary Development Model, and so on. A good reference covering some of these models (in a summary manner) is R. S. Scotti and S. S. Gulu Gambhir, "A Conceptual Framework for a Customer-Centered System Development Life-Cycle Model," Proceedings of the 6th Annual Symposium (San Diego, CA: International Council on Systems Engineering, INCOSE, 1996), p. 547. The reader may also refer to the Vee model, Tufts Systems Engineering Process Model. Plowman's Model, and INCOSE's model, described in *INSIGHT* vol. 5, no. 1, published by INCOSE (April 2002) pp. 7–16. Additionally, for more current information on models, in general, one should review the various issues of *Systems Engineering: The Journal of the International Council on Systems Engineering* (INCOSE), published quarterly by John Wiley & Sons, Hoboken, NJ.

Experience related to the evaluation and assessment of a system, which is operational and being maintained in the user's environment, must be captured. A baseline configuration (with the appropriate metrics) must be established for the purposes of *benchmarking* and the initiation of possible changes for improvement. This, of course, requires that a good comprehensive data collection, analysis, and evaluation capability be implemented to provide the necessary feedback.

This knowledge of what really is happening to the system in the user's environment is critical but is often lacking because there is not a good assessment capability in place. Hence, we often end up introducing the same mistakes again and again as we design new systems. Assessment is an inherent part of the system engineering process. Hardware and virtual software prototypes have greatly helped in establishing the user's experience early in the system and product life cycles.

Finally, when changes are being initiated (whether for corrective action or for product/process improvement), the consequences of such changes must be evaluated from a top system-level perspective; that is, assessing the impact of a change on the overall system. The principles of configuration management and change control must be implemented to ensure that the end results are consistent with the basic requirements in terms of both effectiveness and life-cycle cost (i.e., both sides of the spectrum in Figure 1.7). Such changes may be applicable to the prime mission-related elements of the system, the construction/production capability, the maintenance and support infrastructure, and/or the retirement and material recycling capability. The interaction effects, both upward and downward, must be properly addressed in a *systems* context.

The system engineering process is applicable in all phases of the life cycle, and the successful implementation of such is dependent on many different organizational groups working in a cooperative and integrated manner. Although a single organizational entity may ultimately be responsible for overall leadership in fulfilling system engineering objectives, accomplishing many of the various individual required tasks may be the responsibility of the different participating organizations. In other words, an integrated *team* approach is required.

The traditional systems engineering activities must be "tailored" to every organization, to every task, and to every phase of the life cycle. Each program implementation of systems engineering has special aspects that require some adaptations of the general approach described herein. Large organizations performing critical tasks will require a more robust implementation of the process than a small "start-up" capability. Many large companies have acknowledged this need to "tailor" their development approach by creating internal "start-up" incubators within the organization.[25]

Additionally, this process is applicable to any type of system such as a communication network, a system-on-chip (SoC) or embedded electronics capability, commercial shopping plaza, defense system, information processing network, healthcare capability, transportation system, space system, power distribution network, supply chain network, environmental control capability, and so on. Further, this process can be applied at any level in the overall hierarchal structure presented in Figure 1.4.

[25]"EDA Inflections on Technology Innovation," published in *Chip Design* magazine, vol. 3, May 2011.

1.4 RELATED TERMS AND DEFINITIONS

The preceding framework for the basics in describing the principles and concepts associated with system engineering is now extended to include a few additional but related concepts. Figure 1.21 presents the system development process and life-cycle activities in a slightly different context. System design activities, emphasized in Figure 1.12 and in the first three columns in Figure 1.13, are addressed in blocks 1. 2, and 3; construction and/or production activities are included in blocks 3 and 4, and then there are the system operational and support activities reflected in blocks 5, 6, 7, and 8.

In Figure 1.21, it should be noted that there is a *forward* flow of activities covering not only the design and development of the system, but also the construction and/or production of the system and its elements, the transportation and distribution of these elements to the various operational sites, and the subsequent installation of the system for sustaining operational use. Throughout this flow, there are production and logistics-related activities that are essential if the ultimate system is to accomplish its

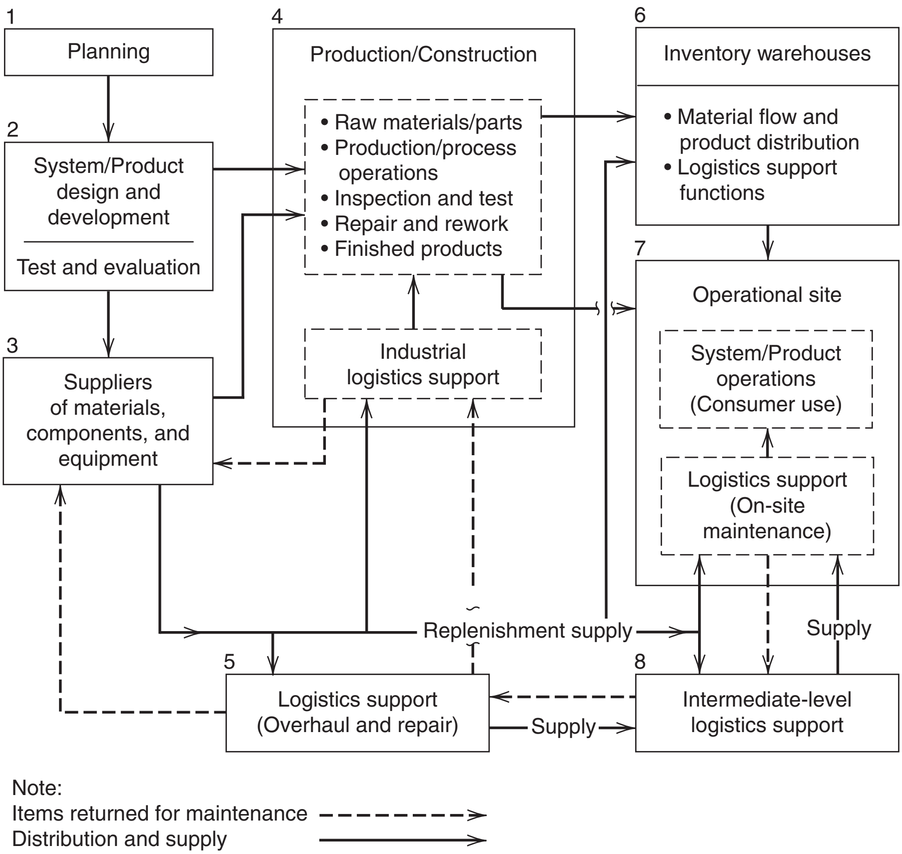

FIGURE 1.21 System operational and maintenance flow.

objectives. For instance, the lack of an effective and efficient transportation capability, the absence of a supplier-produced component in a timely manner, or the lack of appropriate information may preclude the successful accomplishment of a mission requirement(s). Thus, having the appropriate "logistics" available where and when needed is critical.

At the same time, there is a *reverse* (or backward) flow of activities that deal with the maintenance and support of the system throughout its life cycle, which may be necessary in the event of a system failure. An unreliable system that is nonoperational when needed will obviously not be able to perform its designated mission, and there must be an effective and efficient infrastructure that is readily available and in place that can respond to problems, with the objective of repairing the system and returning it to full operational status in a timely manner. The lack of a needed spare part, an available transportation capability, a necessary item of test equipment, required maintenance software, an appropriate repair facility, or the right data, may prevent the system from performing its intended function(s). Thus, there are a variety of logistics and maintenance support functions needed, which are inherent within this reverse flow of activities (as illustrated in blocks 3, 4, 5, 7, 8, and the dotted lines in Figure 1.21).

These activities, associated with both the *forward* and *reverse* flows in Figure 1.21, are characteristic and applicable for any system. However, they are usually addressed downstream and "after the fact" in the life cycle (constituting one of the problems highlighted earlier). In the past, there has been little emphasis on the *design for reliability and maintainability*, *design for producibility*, *design for packaging*, *design for transportation and handling*, *design for supportability and serviceability*, *design for disposability and recyclability*, *design for sustainability and the environment*, and so on. Yet these factors should be considered as critical system-level parameters, along with *the design for performance*, and emphasized in the early phases of the system engineering process illustrated in Figure 1.12.

Given this background information, a number of terms and definitions are discussed further with the objective of strengthening the understanding of systems engineering before addressing the process described in Chapter 2.

1.4.1 Concurrent/Simultaneous Engineering

In the mid-1980s the term *concurrent engineering* became popular, with the objective of placing additional emphasis on "concurrency" as it applies to the design and development of the prime mission-related elements of a system, the construction and/or production capability, the maintenance and support infrastructure, and the retirement and material recycling capability. In Figure 1.11, the various life cycles should be viewed on a concurrent basis, which is directly supportive of system engineering objectives.

One of the first formal definitions resulted from a Department of Defense study, in which *concurrent engineering* is defined as "a systematic approach to the integrated, concurrent design of products and their related processes, including manufacture and support. This approach is intended to cause the developers, from the outset, to consider all elements of the product life cycle from conception through disposal, including quality, cost, schedule, and user requirements."[26] As such, concurrent engineering should be included within the system engineering process.

1.4.2 Some Major Supporting Design Disciplines

Throughout the life cycle, and particularly in system design and development, there are many different individual disciplines that contribute in providing the ultimate system/product configuration. An objective in system engineering is to integrate all of these various design disciplines and specialty groups into a team effort by creating a structured process that proceeds from conception, through development and production, and into system operation. An abbreviated description of nine of these critical disciplines follows:

1. *Software engineering*. From its beginning, software engineering has influenced modern system engineering practices to a great degree and, in some instances, the development of software has proceeded down an independent path without considering the other elements of a system (refer to Figure 1.14 and the software life cycle). Although the element of *software* often constitutes a major part of a system's overall hierarchical structure, it must be properly integrated with the applicable system hardware, people, facilities, information/data, elements of support, etc., on a timely basis. Also, refer to *Agile Engineering*, which constitutes a process utilized in the development of software-based systems.[27]
2. *Reliability engineering*. This design discipline has the objective of ensuring that a system will meet the customer's expectations for good performance throughout its life cycle. Reliability is inherent within and is a major factor in the overall *availability* of a system, and can be measured in terms of probability of success, mean time between failure (MTBF), and failure rate (λ).
3. *Maintainability engineering*. This design discipline has the objective of ensuring that a system incorporates the necessary inherent built-in characteristics such that it can be maintained with ease, accuracy, and economy in the performance of both corrective and preventive maintenance actions as required. Maintainability is the *ability* of a system to be maintained, whereas maintenance constitutes those actions taken to restore a system to (or retain a system in) a specified operating condition. Some typical measures include mean time between maintenance (MTBM), maintenance downtime (MDT), mean corrective maintenance time ($\overline{M}$ ct), mean preventive maintenance time ($\overline{M}$ pt), maintenance labor hours per operating hour (MLH/OH), and cost per maintenance action ($/MA). As such, maintainability (along with reliability) is inherent within and a major factor in the overall *availability* of a system.
4. *Human factors and safety engineering*. This design discipline addresses the *human being* as a major element of a system (refer to Figure 1.14 and the human system integration life cycle). It deals with the anthropometric, human sensory,

[26]R. I. Winner, J. P. Pennell, H. E. Bertrand, and M. M. Slusarczuk, *The Role of Concurrent Engineering in Weapons Systems Acquisition*, Report R-338 (Alexandria, VA: Institute of Defense Analysis, 1988).
[27]Refer to: http://mccadedesign.com/bsb/

physiological, and psychological factors in system design. Safety engineering is a design discipline with the objective of ensuring that a system is *safe* to both operate and maintain throughout its life cycle. Some good measures include the number of successful operational task sequences (in the accomplishment of operator, maintenance, and support tasks) without error, human error rates, personnel safety/hazardous rates, number of training sequences and rates, and so on.

5. *Security engineering*. With the advent of and increased emphasis on *terrorism*, there is a need to design a system such as to preclude the introduction of faults/failures (whether unintentional or through planned actions), which, in turn, will cause serious damage to equipment and facilities, cause the loss of human life, and/or result in some other catastrophic system degradation. The objective is to incorporate good condition monitoring and diagnostic capabilities to ensure that the system is secure in its operation and supported in accomplishing its intended mission(s). Knowing the status of a system's condition at all times is essential. Several measures may include hours/days/years of successful operation without violating system security, number of system faults per unit of time resulting in system damage, loss of life, threat to the environment and society, and so on.
6. *Manufacturing and production engineering*. This discipline addresses both (a) the design of the prime mission-related elements of a system to ensure that they can be produced and/or constructed effectively and efficiently; and (b) design of the production and/or construction capability to ensure that these prime elements can be easily processed as planned (refer to Figure 1.11 and the applicable life cycles as shown). The objective is to incorporate good industrial engineering, reliability and maintainability, human factors, and safety characteristics inherent and within the overall production/construction process. Some good measures include the overall availability of the production process, the number of items produced in a given time frame, the unit cost of a produced item, and the overall equipment effectiveness (OEE) rate.
7. *Logistics and supportability engineering*. Logistics and the system maintenance support infrastructure have generally been considered after the fact and as a separate entity, independent of the basic system that it is supporting. However, if a system is to successfully accomplish its intended mission, this support infrastructure must be in place and available, in a timely manner, in the event of need (refer to Figure 1.21, blocks 4–8). If a specific time frame (i.e., date) is established for system operation, then the appropriate logistics support must be available before the fact (and throughout) in order for this to happen; if a system failure occurs during the period of system utilization, then the right maintenance support infrastructure must be available and responsive in order for the system to complete its objective, and so on. To ensure that all system requirements are fulfilled, the basic mission-related elements of a given system must be *designed for supportability*, and the logistics and maintenance support infrastructure must also be designed such that it can both effectively and efficiently support these elements. In other words, this infrastructure must be considered as a major element of the system, and supportability requirements

must be "designed in" along with reliability, maintainability, human factors, safety, and other related factors (Figure 1.21, blocks l–3).

8. *Disposability engineering*. This discipline primarily addresses the design of a system and its components such that, in the event of obsolescence, they can be either recycled for other applications or processed for disposal. In the event of disposal, the results should not have any degradation or negative impact on the environment (i.e., not result in any residual solid waste, toxicity, noise pollution, water pollution, or the equivalent).[28]
9. *Environmental engineering*. The *environment*, as defined herein, refers primarily to ecological considerations such as air pollution, water pollution, noise pollution, radiation, and solid waste. The design objective concerns itself both with the impact of the new system design configuration on those factors in the environment (i.e., the creation of negative impacts on the external environment), and the impact of similar factors from other outside sources on the new system being introduced (the external impact of these negative conditions on the system itself). Such concerns must be addressed in the system design process from the beginning.

The nine disciplines mentioned represent only an example of a few critical areas that require some emphasis in system design and development, along with the more traditional engineering disciplines such as aerospace and aeronautical engineering, chemical engineering, civil engineering, electrical engineering, industrial engineering, mechanical engineering, and others, as applicable. A key objective of system engineering is to create a "team" approach through the proper integration of these and other needed disciplines in a timely manner.

1.4.3 Logistics and Supply-Chain Management (SCM)

The subject of *logistics* is introduced from an engineering design perspective in Section 1.4.2, but it is essential that a review of the overall scope of logistics be included as well. The term *logistics* can be described somewhat differently, depending on application (e.g., *commercial* versus *defense*), the companies and organizations involved, and one's personal background and experience. In the commercial sector, *logistics* is often defined as that part of the supply chain process that *plans, implements*, and *controls* the efficient, effective forward and reverse flow and storage of goods, services, and related information between the point of origin and the point of consumption in order to meet customer requirements. The *supply chain (SC)* refers to that group of organizations and activities pertaining to the overall flow of materials and services from various supplier sources to the ultimate customer(s).

For many years, emphasis was primarily directed to the physical aspects of supply, materials handling, and transportation and distribution, as shown in Figure 1.22. However, during the past several decades, the area of *commercial* or *business* logistics has been expanded significantly with the introduction of the latest

[28] Additional material on the "design for producibility and disposability" may be found in B. S. Blanchard and W. J. Fabrycky, *Systems Engineering and Analysis*, 5th ed. (Upper Saddle River, NJ: Pearson Prentice-Hall, 2011), Chapter 16, pp. 541–564.

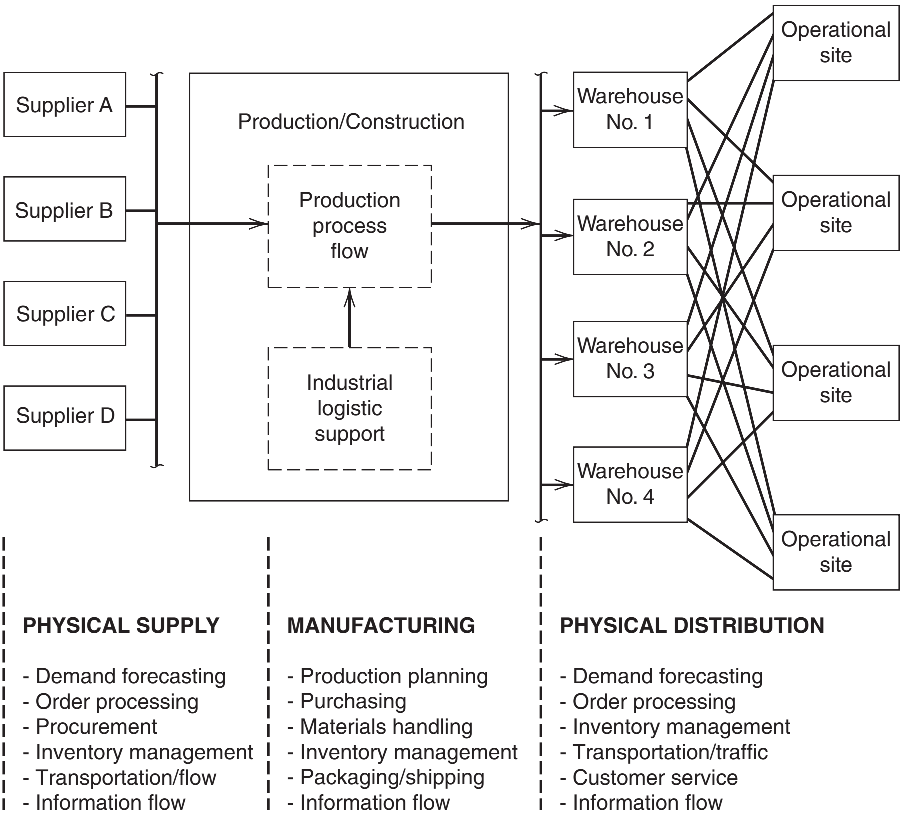

FIGURE 1.22 Logistics activities in the production process.

electronic commerce (EC) methods, information technology (IT), electronic data interchange (EDI), development of radio-frequency identification (RFID) tags and global positioning system (GPS) technology, and the application of good business processes to the activities shown in Figure 1.22. Further, the trends toward more outsourcing, greater international competition, and increased globalization have resulted in the need for the establishment of coalitions and industry/government partnerships worldwide. This, in turn, has resulted in the currently popular concepts associated with *supply-chain management (SCM)*. SCM pertains to *management of the supply chain, or group of supply chains, with the objective of providing the required customer services, both effectively and efficiently*. It requires a highly integrated approach, employing the appropriate resources (e.g., transportation, warehousing, inventory control, and information) and implementing the necessary business processes to ensure complete satisfaction. The Council of Supply Chain Management Professionals (CSCMP) has adopted this definition.[29]

[29]This definition was developed by the Council of Supply Chain Management Professionals (CSCMP), 2805 Butterfield Road, Oak Brook, ILB.

In any event, no matter how one may wish to define it, the associated concepts and magnitude of SCM are certainly growing in recognition and importance. SCM, as described herein, basically includes those *forward* flow of activities reflected by blocks 3, 4, 6, and 7 in Figure 1.21.[30]

1.4.4 Integrated System Maintenance and Support

In Figure 1.21, there is also a *reverse* flow relating to the activities in blocks 3,4, 5,7, and 8, with the dotted lines indicating a path whereby faulty (or obsolete) items are sent for maintenance and repair (or disposal) as necessary. Figure 1.23 shows an expansion of this, presented in the form of a maintenance and support infrastructure.[31]

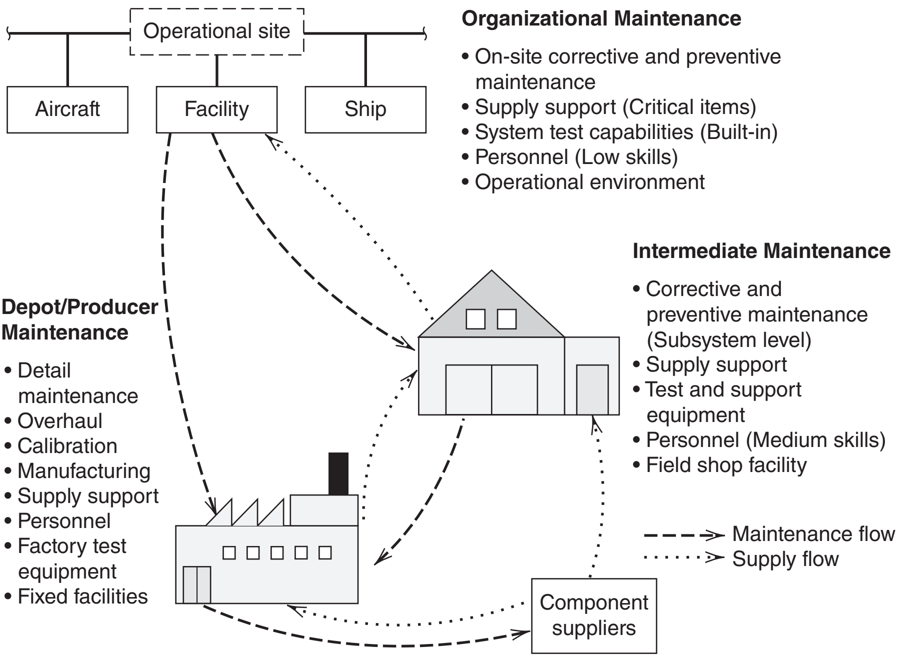

FIGURE 1.23 System maintenance and support infrastructure.

[30]It should be noted that the material presented thus far emphasizes "logistics" in the commercial (business) sector and not the entire spectrum of logistics as practiced in the defense sector. Historically, logistics as it applies for defense systems, has been covered within the context of integrated logistic support (ILS) which, in turn, has included not only what is described in Section 1.4.3 but also the maintenance and support infrastructure that is discussed further in Section 1.4.4. Recently, the term acquisition logistics has become popular within the Department of Defense (DOD) and includes the principles and concepts of SC/SCM and system maintenance and support, as applicable in each and every phase of the system life cycle.

[31]The system maintenance and support infrastructure (i.e., the maintenance concept and showing major repair policies) is discussed further in Chapter 2, Section 2.5.

Figure 1.23 illustrates an example of a basic *three-levels-of-maintenance* approach (i.e., organizational maintenance, intermediate-level maintenance, and depot/ producer/supplier maintenance). Depending on the type of system, the mission to be accomplished, the system's complexity and reliability, the geographical location and where utilized, customer desires, overall cost, and so on, there may be some variation in repair policies and the infrastructure may include only two levels of maintenance. Further, there may be one or more third-party maintenance contractors involved in the overall network, and this configuration may well change as the system ages and the demand for support changes. In any event, there must be some form of a maintenance and support infrastructure, in place and readily available, to ensure that the system will continue to be operational when required.

Referring to the network in Figure 1.24, it can be seen that the functional elements of support include maintenance and support planning, maintenance personnel, supply support (spares/repair parts and associated inventories), test and support equipment, packaging and transportation, maintenance facilities, computer resources, technical data, and related management requirements. These various elements must be properly integrated, with the support of an effective management information capability.

Traditionally, the subject of maintenance has been addressed after the fact and downstream in the system life cycle, and the maintenance and support infrastructure has not been considered as an element of a system but rather as a separate and

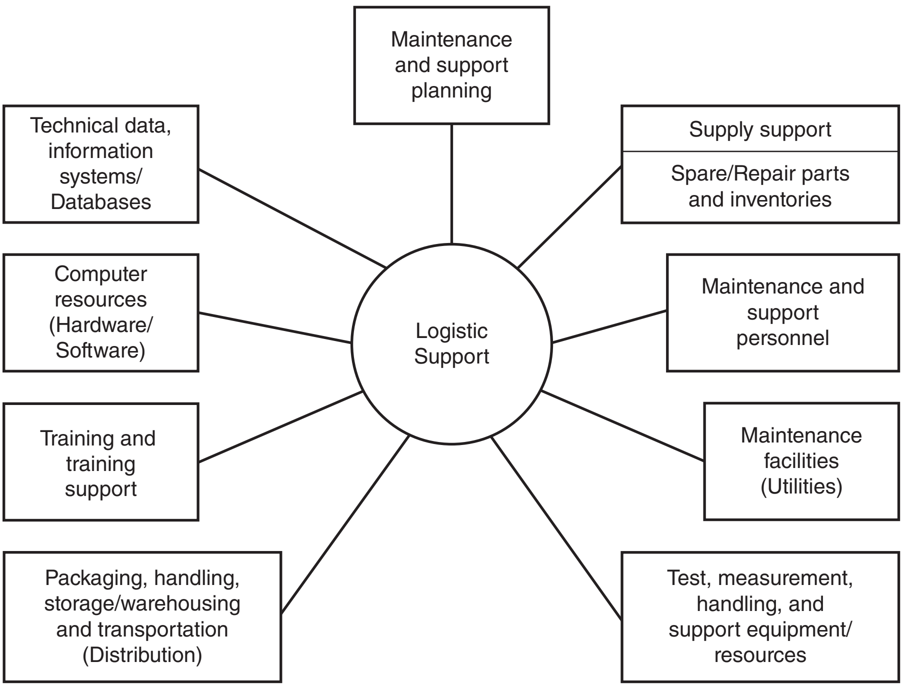

FIGURE 1.24 Functional elements of logistics.

somewhat unrelated entity. Through the years, the results of such a practice have been rather costly, with a large portion of the total life-cycle cost for many systems being attributed to maintenance activities accomplished downstream (refer to Figures 1.7 and 1.8). As a result and in response, there have been some efforts leading to the recognition of the maintenance and support infrastructure early in the system life cycle and as an inherent element of a system (refer to Figure 1.9).

In the defense sector, the DOD initiated the concept of *integrated logistic support* (ILS) in the mid-1960s. ILS is a management function that provides the initial planning, funding, and controls that help to ensure that the ultimate consumer (or user) will receive a system that will not only meet performance requirements, but can be supported expeditiously and economically throughout its programmed life cycle. According to Wikipedia, ILS can be defined as *an integrated and iterative process for developing material and a support strategy that optimizes functional support, leverages existing resources, and guides the system engineering process to quantify and lower life cycle cost and decrease the logistics footprint (demand for logistics), making the system easier to support. Although originally developed for military purpose, it is also widely used in commercial product support or customer service organizations*. A major ILS objective is to ensure the proper and timely integration of the elements of support, as shown in Figure 1.24.[32]

More recently, in the interest of economy, the DOD has been emphasizing (1) increased reliance on the utilization of best commercial logistics and supply chain practices in meeting the support requirements for defense systems, and (2) an increased emphasis on the *design for supportability/sustainability* and consideration of the maintenance and support infrastructure within the context of the systems engineering process. Relative to the first area, there has been a great deal of growth in adopting many of the principles of supply chain management in the defense sector, particularly in view of the growth in information technology (IT) and electronic commerce (EC) methods. In regard to the second area of emphasis, the concept of *acquisition logistics* has become popular, and its focus is segmented into three interrelated parts: (1) designing the system for support, (2) designing the support system, and (3) acquiring the support elements.[33] In essence, logistics in the defense sector has become an integrated mix of the activities depicted in Figures 1.22 and 1.23.

When addressing the overall issue of maintenance, another consideration must be properly integrated within this overall spectrum of system support. In the commercial sector, and primarily oriented to the maintenance of equipment in a typical manufacturing plant, the concept of *total productive maintenance (TPM)* has become quite popular and is currently being implemented worldwide. TPM, a concept originally developed by the Japanese in the late 1960s and early 1970s, is a system-oriented,

[32]http://en.wikipedia.org/wiki/Integrated_logistics_support

[33]Mil-HDBK-502A, Product Support Analysis, Department of Defense, March 2013, which supersedes MIL-HDBK-502, Acquisition Logistics, 1997. This requirement supports the system engineering process.

life-cycle approach to maintenance, with the objective of maximizing productivity in the commercial manufacturing plant. TPM:[34]

1. Promotes the overall effectiveness and efficiency of equipment in a factory. It includes *maintenance prevention* (MP) and *maintainability improvement* (MI), which consider the appropriate incorporation of reliability and maintainability characteristics in design.
2. Establishes a complete preventive maintenance program for factory equipment based on life-cycle criteria (similar to the reliability-centered maintenance approach used in establishing preventive maintenance requirements).
3. Is implemented on a "team" basis, involving various departments to include engineering, production operations, and maintenance.
4. Involves every employee in the company, from the top management to the workers on the shop floor. Even equipment operators are responsible for the care and maintenance of the equipment they operate.
5. Is based on the promotion of preventive maintenance through "motivational management" (the establishment of autonomous small-group activities for the maintenance and support of equipment).

The introduction of TPM was motivated by the high costs of producing products in the factory, combined with the fact that a good percentage of these high costs was attributed to equipment maintenance on the production line. The objective is to reduce the costs of producing products by minimizing maintenance costs in the factory. The implementation of TPM, which can be measured in terms of *overall equipment effectiveness (OEE)*, has become quite popular, particularly during the past several decades.

As a final note, it should be reemphasized that the logistics and maintenance support infrastructure (i.e., the overall structure described in Sections 1.4.3 and 1.4.4) must be considered and included as a major "element" of a system, and that this element must be properly integrated, along with the other system elements, within the system engineering process from the beginning.

1.4.5 Data and Information Management

Characteristic of most programs is the large amount of data and information associated with the design and development, operation, and the sustaining maintenance and support of systems throughout their respective life cycles. Included in such data requirements are specifications and plans, engineering drawings and associated design data, system test and evaluation data, logistics and maintenance data, technical data (in the form of system operating instructions, maintenance and overhaul

[34]The concept of TPM was initiated in Japan, through the Japan Institute of Plant Maintenance, in the late 1960s. Refer to Appendix F for additional references (see S. Nakajima, et al).

manuals, and component parts lists), engineering change data, subcontractor and supplier data, and so on. Additionally, there is a vast amount of information (e-mail, Internet communications, Twitter, and various categories of messages) that is generated and distributed daily among the various individuals and organizations that are involved throughout the different stages of the system life cycle.

The challenge is to properly plan for, coordinate, and integrate such data/information into a unified data package, from the beginning. Such a data package not only includes a historical basis for what has been accomplished and why it was done, but also reflects the proper configuration of a system at any given point in time. The generation of too much data, or too little data, can be very costly. Thus, it is critical that the right amount of data/information be available when required and in a timely manner, not too early or too late. This, in turn, requires some coordination and integration covering all aspects of system-level activities and across organizational lines (including the customer, major contractor, subcontractor, and supplier organizations).

The initial determination of data requirements stems from the early development of system-level requirements in conceptual design. These data requirements are then expanded as the system is further defined through the functional analysis and allocation of requirements (refer to Figure 1.13). From this point on, there are likely to be many different requirements, in different forms, throughout a given program. Thus, it is important that the data/information integration and management function be included within the overall spectrum of system engineering.[35]

1.4.6 Configuration Management (CM)

Configuration management (CM) is a management approach that includes identifying, documenting, and auditing the functional and physical characteristics of an item, recording the configuration of the item, and controlling the changes to the item and its documentation. The purpose is to provide a complete audit trail of design decisions and system modifications. CM is a concept of *baseline* management, which includes the definition of the *functional* baseline for a system, the *allocated* baseline, and the *product* baseline identified in Figure 1.13. Successful fulfillment of system engineering requirements is heavily dependent on a good disciplined approach to baseline management. This is particularly true in considering the current trends toward evolutionary design and the introduction of new technologies into a system configuration on a continuing basis.[36]

[35]The objective is to develop just the right amount and right type of data, in a timely manner, so as to enable a full "description" of the system configuration at any specific time throughout the life cycle. The tendency at times is to avoid the preparation of essential data until later and then cause the loss of traceability from one configuration to the next. It should be noted that, more recently, the application of Model- Based System Engineering (MBSE) concepts and principles have been implemented with the objective of reducing the amount of data required.

[36]The important issue includes the proper management of "changes." Refer to MIL-HDBK-61A (SE), Configuration Management Handbook, February 2001.

1.4.7 Total Quality Management (TQM)

Total quality management (TQM) can be described as a totally integrated management approach that addresses system/product quality during all phases of the life cycle and at each level in the overall system hierarchy. It provides a before-the-fact orientation to quality, and it focuses on system design and development activities as well as manufacturing and production, maintenance and support, and related functions. TQM is a unification mechanism linking human capabilities to engineering, production, and support processes. The emphasis is on total customer satisfaction, the iterative practice of "continuous improvement," and a total integrated organizational approach. As part of the initial system design and development effort, consideration must be given to (1) the design of the processes that will be used to manufacture and produce the components of the system and (2) the design of the support infrastructure that will provide the necessary ongoing maintenance of that system throughout its planned life cycle. In this regard, the principles of TQM must be inherent within the system engineering process.

1.4.8 Total System Value and Life-Cycle Cost (LCC)

A system should be measured in terms of its total value to the consumer. For the purposes of discussion, it is necessary to consider both sides of the balance, as illustrated in Figure 1.25; that is, the *technical factors* and the *economic factors*. Of particular interest within the domain of system engineering is the issue of *life-cycle cost* (LCC). LCC includes all costs associated with the system life cycle, which can be broken down as follows:

1. *Research and development (R&D) cost*. This includes the cost of feasibility studies; developing operational and maintenance requirements; system analyses; detail design and development; fabrication, assembly, and test of engineering models; initial system test and evaluation: and associated documentation.
2. *Production and construction cost*. Included here are the cost of fabrication, assembly, and test of operating systems (production models); operation and the sustaining maintenance and support of the manufacturing capability; facility construction; and the acquisition of an *initial* system support capability (e.g., test and support equipment, spare/repair parts, and technical documentation).
3. *Operation and maintenance cost*. This includes the cost of system operation and the sustaining maintenance and support of the system through its planned life cycle (e.g., manpower and personnel, spare/repair parts and related inventories, test and support equipment, transportation and handling, facilities, software, modifications, and technical data).
4. *System retirement and phase-out cost*. This final expense is the cost of phasing the system and its components out of the inventory because of obsolescence or wearing out, recycling of items for further use, condemnation, and the disposal of materials.

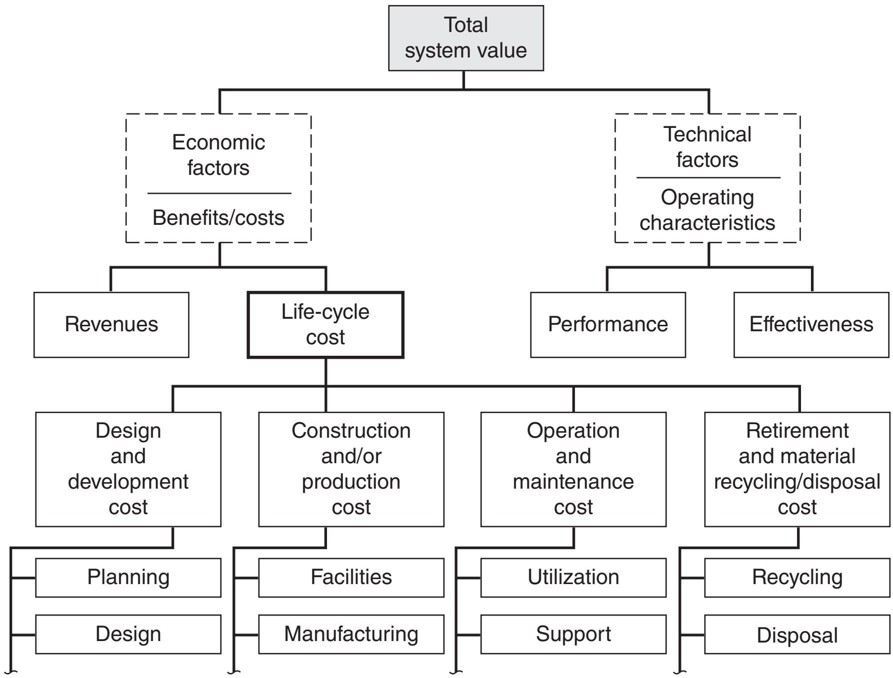

FIGURE 1.25 Total system value.

Life-cycle costs can be categorized many different ways depending on the type of system and the sensitivities desired in cost-effectiveness measurement. The objective is to ensure *total cost visibility* (see Figure 1.8).This is necessary if one is to be able to properly assess the risks associated with each of the major design and management decisions made throughout the life cycle. Life-cycle cost (LCC) is a major theme throughout this text, and the process for conducting a life-cycle analysis is highlighted in Appendix B.

1.4.9 Some Additional Terms And Definitions

Most system engineering activities in industry are closely tied to *Product Life-Cycle Management (PLM)* tools and processes. The focus of PLM is on the technical aspects of product design, development, and product launches. In this way, PLM provides the product information backbone for the organization; that is, the master product data management across various product configurations and derivatives. Further, companies concerned with "knowledge" retirement, especially in the aerospace and defense industries, try to retain this experience by capturing project requirements in modern PLM tools and processes.

As commercial product life cycles have, in general, shrunk from 5 years to as low as 18 months or less, design-focused PLM tools have become closely integrated with various "business counterparts." *Enterprise Resource Planning (ERP)* deals with the execution of all business aspects of manufacturing (to include all transactional data from customer orders, translating the same to a manufacturing order, shipping, and invoicing). The growth of virtual digital floors combined with the more traditional physical processes have contributed to the rise of the *Manufacturing Execution System (MES)* capability. MES is utilized to manage a company's shop floor, controlling the equipment as well as scheduling production runs across the different routings within a production fabrication run.

In summary, these and related techniques and methods must be properly integrated, along with other comparable procedures, through the proper implementation of the higher-level *System Engineering Management (SEM)* requirements.

1.5 SYSTEM ENGINEERING MANAGEMENT

The successful realization of system engineering principles and concepts is dependent not only on the *technology* issues and the process for implementing such, but on the *management* issues as well. As illustrated in Figure 1.26, there are two sides of the spectrum, and each is highly dependent on the other. The best tools/models may be available to implement the process shown in Figure 1.13. However, there is no guarantee for success unless the proper organizational environment has been created and an effective and efficient management structure is in place. Top management must first *believe in* and then *provide the necessary support* to enable the application

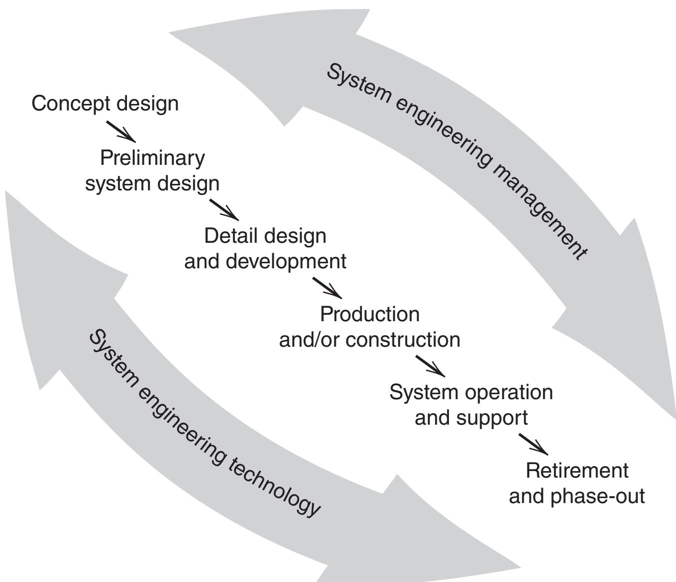

FIGURE 1.26 Management and technology application to the system engineering process.

of system engineering methods to all applicable projects, both in-house and external. Specific objectives must be defined, policies and procedures must be developed and properly implemented, and an effective review and reward structure must be supportive. This structure must prevail throughout the customer, prime contractor, and down through the various applicable supplier organizations as required. The challenge is that of *proper implementation*.

Although there are variations from one program to the next, Figure 1.27 presents a baseline for discussion. The major program phases and milestones are noted, along with a few selected activities and events that are considered to be significant from a system engineering perspective. It should be noted that the emphasis in the figure is primarily on the system acquisition (procurement) process, and not on the downstream operation, maintenance and support, and retirement phases of the life cycle. These phases of activity, which are discussed in detail in subsequent chapters, are briefly summarized as follows:[37]

1. During the early stages of conceptual design, it is essential that good communications between the producer and the consumer(s) be established from the beginning. Defining the *real* need, conducting feasibility analyses, developing operational requirements and the maintenance concept, and identifying specific quantitative and qualitative requirements at the system level are critical. These requirements must be properly conveyed through a well-prepared system specification (Type A). This top-level system specification constitutes the most important *technical* document, from which all lower-level specifications evolve. Without a good foundation from the beginning, all subsequent lower-level requirements may be questionable (refer to Chapter 3).
2. During the latter stages of conceptual design, a comprehensive System Engineering Management Plan (SEMP), or System Engineering Plan (SEP), must be developed to ensure the implementation of a program that will lead to a well-coordinated and integrated product output. The SEMP, which evolves from the top-level Program Management Plan (PMP), integrates all lower-level planning documents. It includes the design-related tasks necessary to enhance the day-to-day system development effort, the implementation of concurrent engineering methods, and the integration of the appropriate organizational entities into a "team" approach. The SEMP must directly support the requirements in the system specification (Type A) from a *management* perspective, and the two documents must "talk to each other." The SEMP is addressed in detail in Chapter 6.
3. During the latter stages of conceptual design, a Test and Evaluation Master Plan (TEMP), or equivalent, must be developed for the purposes of assessment and ultimate validation. As requirements are initially specified in the system

[37]The process and milestones presented in Figure 1.27 are representative for a relatively large-scale system. The important issue is understand the concepts and principles described and to "tailor" these requirements for the system of your choice.

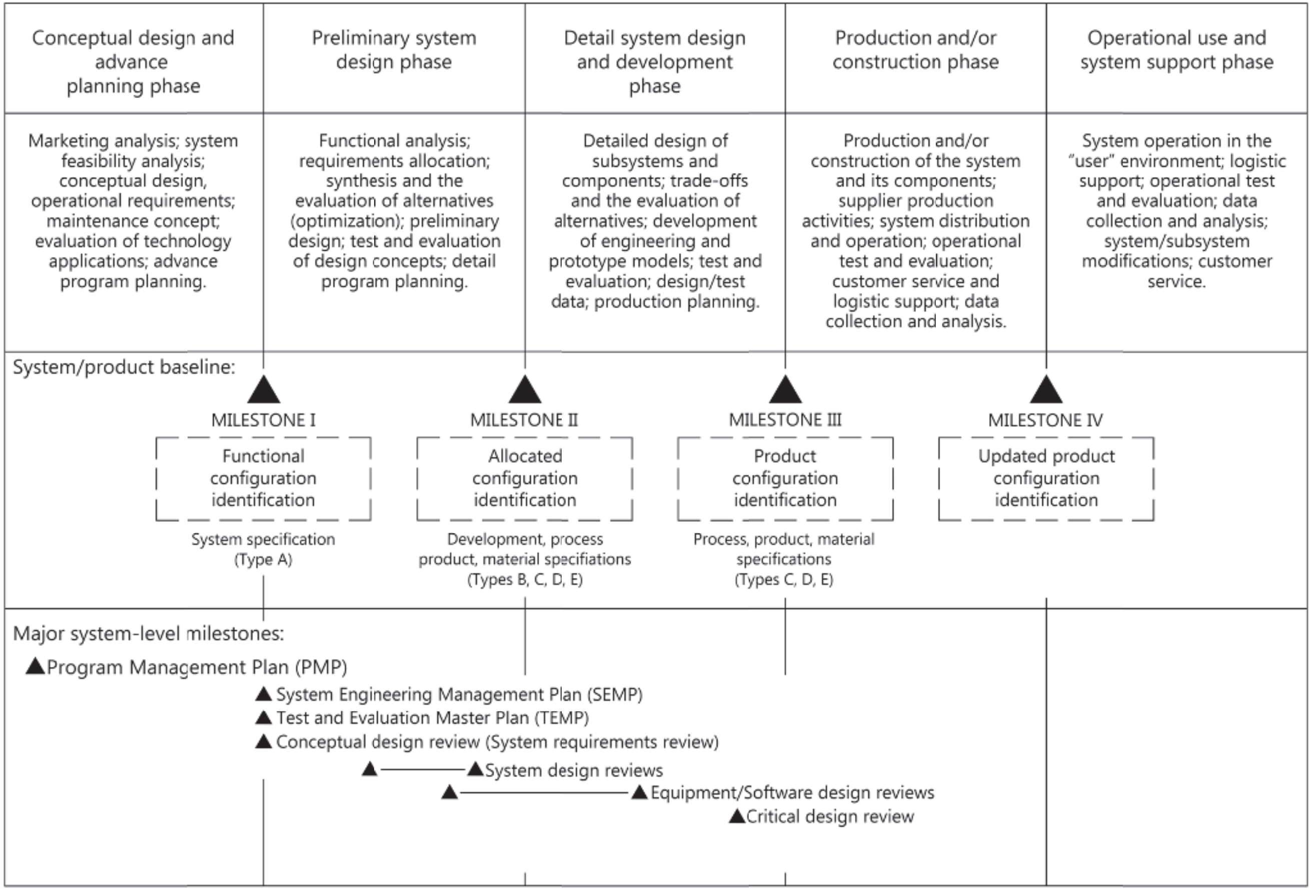

FIGURE 1.27 The system acquisition process and major milestones.

specification (Type A) and planned through the tasks described in the SEMP, the methods/techniques to be used for measuring and evaluating the system to ensure compliance with these requirements must be described. This plan must address test and evaluation activities on a fully integrated basis, employing the appropriate combination of simulation and other analytical tools, mock-ups, laboratory models, and prototype models. Test and evaluation are covered further in Chapter 2.

4. As system design and development progresses, there is a need to schedule a series of formal design reviews at discrete points where the design configuration evolves from one level of definition to another; that is, conceptual, system, equipment/software, and critical design reviews. The purpose of these reviews is to ensure that the specified requirements are being met prior to entering into a subsequent phase of effort, and to ensure that the necessary communications exist across organizational lines. See Chapter 5 for further discussion of design reviews and evaluation requirements.
5. Toward the latter stages of detail design, throughout the construction/production phase, and during the operational use and maintenance support phase, there is a need to provide ongoing assessment and validation of the system. The objective is to ensure that the consumer requirements are being met and to establish a "baseline" for the purposes of benchmarking and the initiation of *a continuous process improvement* activity. Design changes are initiated as required to correct any noted deficiencies.

The successful implementation of system engineering principles is highly dependent on proper management of the simplified process depicted in Figure 1.28. Inherent in this process is the application of different technologies employed to facilitate the steps of requirements analysis, functional analysis and allocation, synthesis, design optimization, and validation.

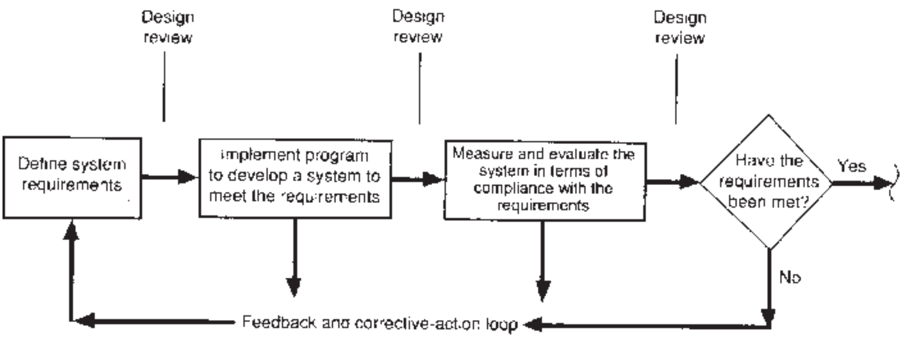

FIGURE 1.28 The basic system requirements, evaluation, and review process.

1.6 SUMMARY

This chapter provides an abbreviated introduction to some of the key terms and definitions, principles and concepts, and critical issues in the implementation of *system engineering* and associated requirements in the design and development, production/construction, operation and support, and retirement of systems. Such terms as a *system, system of systems (SOS)*, *system architecture*, *system science*, *system analysis*, *logistics*, *integrated system maintenance and support*, *configuration management, total quality management*, and *system value and life-cycle cost* are introduced. Hopefully, this will stimulate the thought processes needed for the material ahead. The information presented herein, and particularly the concepts illustrated in Figures 1.13, 1.14, and 1.15, are a natural introduction to the system engineering process discussed in Chapter 2.

QUESTIONS AND PROBLEMS

1. Provide, in your own words, a definition of a *system*. Include some examples.

2. Select a system of your choice and describe the system life cycle. Construct a detailed flow diagram *tailored* to your situation.

3. Describe what is meant by a *system of systems (SOS)*. Provide an illustrated example.

4. When referring to the basic *elements* of a system, what is included?

5. Define *system engineering*. What is included? Why is it important? How does system engineering differ from system science and system analysis?

6. What are the differences (or similarities) between system engineering and some of the more traditional disciplines such as civil engineering, electrical engineering, mechanical engineering, and so on?

7. Refer to Figure 1.11 (Example A). Describe the interrelationships between the three illustrated life cycles.

8. Refer to Figure 1.14. What are some of the key system engineering objectives that can be applied?

9. Refer to Figure 1.15. What are some of the key system engineering objectives that can be applied?

10. What is the significance of the feedback process illustrated in Figure 1.16?

11. What are the major system engineering functions in conceptual design? Preliminary design? Detail design and development? System operational use and life-cycle support? Retirement, phase-out, and disposal?

12. Define Agile Engineering, its application, and how it relates to system engineering.

13. What are the basic differences between the Waterfall Model and the Agile Engineering Model?

14. Describe the basic differences between the Waterfall model, the Spiral Model, and the Vee Model. How do they compare with the model proposed by the authors?

15. Refer to Figure 1.21. Briefly highlight the activities that are critical for the successful implementation of the system engineering process. When in the life cycle must these activities be addressed?

16. Refer to Figure 1.22. Describe how these activities might affect/influence system engineering (if at all).

17. Refer to Figure 1.23. Describe how these activities might affect/influence system engineering (if at all).

18. Refer to Figure 1.24. Explain why these elements should be considered (or not considered) as inherent elements of a system.

19. Refer to Section 1.4.2. Although each of these supporting design disciplines is important in the implementation of system engineering requirements, from your perspective which one(s) should receive a greater degree of emphasis than the others? Why? If some degree of prioritization is required, how would you accomplish such?

20. The successful implementation of the system engineering process is dependent on both technological and management issues. Explain why. Provide an example of how one can affect the other.

21. Why is the system specification (Type A) important? Develop an outline for a system specification of your choice.

22. Describe PLM, ERP, and MES. How do these processes interact with systems engineering management?

23. What is the purpose of design reviews?

24. What is *concurrent engineering?* How does it relate to system engineering?

25. What is *configuration management?* Why is it important in system engineering?

26. Why is *logistics* important? How does it relate to system engineering (if at all)?

27. What is *life-cycle cost?* What is included? When is it first considered and applied? Why is it important to consider such cost in the decision-making process?

28. Describe, in your own words, some of today's challenges relative to the implementation of system engineering, when considering the current environment. Consider the application of MBSE concepts and principles as applied in the system engineering process in your discussion.