INTRODUCTION TO SYSTEMS ENGINEERING

Systems engineering [1] methodologies and practices began to emerge from experience gained in the U.S. Department of Defense acquisition programs of the 1950s. These programs often involved complex and challenging user requirements that tended to be incomplete and poorly defined. Additionally, most programs entailed high technical risk because they involved large numbers of different technical disciplines and the use of emerging technology. Following a number of program failures, the discipline of systems engineering emerged to help avoid, or at least mitigate, some of the technical risks associated with the complex equipment acquisition programs. Systems engineering provides the framework, within which complex systems can be adequately defined, analyzed, specified, manufactured, operated and supported. Systems engineering processes and methodologies have continued to develop since the 1950s, and are widely applied to many of today’s challenging acquisition projects.

The focus of systems engineering is on the system as a whole, and the maintenance of a strong interdisciplinary approach. Project management, quality assurance, integrated logistics support, and traditional design disciplines such as hardware and software engineering are but a few of the many disciplines that are part of a coordinated systems engineering effort.

Examples

Throughout the following chapters we use a number of examples wherever possible to illustrate and reinforce the systems engineering theory being introduced. To avoid duplication and assist further with an understanding of the whole systems engineering process, throughout the chapters we also use a single worked example, based on the acquisition of an aircraft system. We do not intend to replicate the design process for a modern aircraft and supporting elements. Rather, an aircraft system has been chosen as a convenient example that can be readily recognized by readers from a wide variety of disciplines and specialties. The illustrations are designed so readers are not forced to become domain experts in a particular field just to understand the illustration. With the aircraft example, the majority of readers can immediately understand the system context, the need, the functional and performance requirements, the interface issues, technical performance measures, the functional-to-physical translation, broad trade-off analyses, as well as the physical configuration items involved in the final design. It should be noted, however, that we do not at all suggest that the aggregation of examples throughout the
text represents an adequate design for an aircraft system; the available space prohibits the inclusion of sufficient detail, which would also obscure the general lessons that are to be illustrated by the example.

**Example 1.1: Introduction to Aircraft Example**

A large aircraft operator (ACME Air) has identified a need for a medium-sized aircraft to replace the aging platform that it currently operates over domestic routes and some short international routes. The company wants to use a systems engineering approach to ensure that the aircraft system produced is ideally suited to the role and to ensure the overall commercial viability of the project.

This brief introduction provides the seed of an idea that grows throughout the remaining chapters as we consider the relevant systems engineering activities required to see the system through definition and design, construction and/or production, operation, maintenance, support, phase-out and disposal.

### 1.1 WHAT IS A SYSTEM?

A system is a complex set of many often-diverse parts subject to a common plan or serving a common purpose. In the broadest sense, a system is something that provides a solution to a complex problem. With this in mind, a system combines a number of resources together in an organized manner so as to perform a collection of specified functions to specified levels of performance.

A system can be defined in two broad ways—in functional terms and in physical terms. A functional description of a system articulates what the system will do, how well it will do it, under what conditions it will perform and what other systems will be involved with its operation. A physical description relates to the system components and explains what the components are, how they look, and how the components are to be manufactured and integrated. The two descriptions live independently as valid descriptions of a system, but an understanding of the relationship between functional and physical description leads to a deeper appreciation of the system.

A system must meet some defined need, goals and objectives, which must be clearly stated by the user, and represent the start point of the design process as well as the ultimate test of the system’s fitness for purpose once it has been introduced into service.

A system is much more than an aggregation of hardware or software and must be described in terms of such resources as personnel, materials, facilities, data, hardware and software. Analysis of these resources yields the majority of the system’s functional and performance requirements. The system is fully defined by the combination of these resources operating in a live environment, which defines the context within which the system must
operate and often dictates functional or performance requirements to be exhibited by the system.

**Example 1.2: System Resources for our Aircraft Example**

Resources for our aircraft system example could include, but not be limited to:

- **Personnel.** Air crew are required to operate the system and ground crew are required to maintain and support the fleet of aircraft.

- **Materials.** Materials are required to operate the system, including fuel, lubricants and other consumables such as tires and spare parts.

- **Facilities.** The aircraft needs a network of maintenance facilities for routine maintenance and repairs throughout its life. Other facilities such as terminals are also necessary to operate the aircraft.

- **Data.** Data is required to maintain and operate the aircraft. Data could include maintenance information such as specifications and drawings, and operational information such as user manuals and instructions.

- **Hardware.** The most tangible part of the system is the hardware itself. The aircraft will be produced, distributed and sold to operators who will then use the aircraft in a number of different ways such as domestic and international operations for passengers or freight.

- **Software.** Software is rapidly becoming a critical item within many systems. The aircraft is likely to use hardware and software to control a range of functions from engine management, through navigation and environmental control systems, to the communications and flight-control systems.

Throughout this book we use a number of terms to refer to the different parties involved in system development.

The **customer** organization is generally represented by an acquisition element (typically headed by a **project manager**) that procures the new system on behalf of the **users** (**operators**) within the organization who own the perceived need. In addition to the acquisition staff and users, there are many others within the customer organization who have a stake in the successful implementation of the project. These **stakeholders** include representatives from the management, financial, operations, supply, maintenance and facilities areas, to name just a few.

The **contractor** is the entity responsible for designing and developing the system to meet the customer requirements. The relationship between the customer and the contractor varies with each project but, for each project, is defined by the terms and conditions of the **contract.** In most cases the contractor is not able to perform all of the work required and devolves
packages of work to a number of subcontractors. The terms and conditions relating to this work are described in the relevant subcontract.

Systems are hierarchical by nature. In this text, we limit our discussions to a three-layer hierarchy to describe the focus of various systems engineering processes. We describe a top-level entity known as the system that comprises a number of subsystems that, in turn, consist of a number of components. We could go further, but it serves little purpose in this text.

The application of the terms to specific situations and examples depends very much on the context of the situation and where within the overall picture the players involved in the situation sit. For example, if we talk about the highest-level considerations of the aircraft example, we would talk about the aircraft system consisting of, among others, the engine subsystem and the avionics subsystem. The engine subsystem may consist of components such as fuel tanks, pumps and lines, turbines, compressors and gear boxes, and hydraulic pumps. If, however, we were talking from the viewpoint of an engine manufacturer, we would talk of the engine as the system, comprising fuel, power plant and hydraulic subsystems and so on.

1.2 SYSTEM LIFE CYCLE

The life cycle of a system commences with the statement of a need and ends with disposal of the system. In between there are a number of system phases and activities, each of which builds on the results of the preceding phase or activity. There is no universally accepted agreement on how many phases exist in a system life cycle or what those phases are called. For the discussion in this book, we use the system phases defined by Blanchard and Fabrycky [2] and MIL-STD-499B [3] and illustrated in Figure 1.1.

We have selected this model for a number of reasons. It emphasizes that a system begins with a perceived need and finishes upon disposal—the so-called cradle-to-grave approach. There is a clear delineation between the acquisition and in-service (utilization) phases of a system, allowing the application of systems engineering during utilization to be investigated and documented. In addition, it shows sufficient detail in the early stages of the acquisition where the application of systems engineering methodologies and practices has the potential to make the most significant contribution.

![Figure 1.1. System life cycle.](image-url)
As shown in Figure 1.1, the system life cycle can be divided into two very broad phases: the *Acquisition Phase* and the *Utilization Phase*. The life cycle of a system begins with a perceived need by a group of users, which provides the input for the Acquisition Phase that continues until the system is brought into service. The system stays in service until the Utilization Phase concludes with disposal of the system. Quite often the conclusion of one system life cycle marks the beginning of another system’s life cycle.

The significance of focusing on the system life cycle is that decisions made early in the Acquisition Phase are informed of the proposed and intended activities in the Utilization Phase. For example, the design of an aircraft airframe must take into account the maintenance and operation of that airframe during the Utilization Phase. It would be pointless to design the best airframe in the world if it did not have the necessary access points to allow maintenance personnel to service it or operators to operate it in the intended environment. Other examples of Utilization Phase requirements that impact on equipment design or selection during the Acquisition Phase include *reliability* and *availability*. Reliability normally refers to the ability of the equipment to operate without failure for a given period of time. Availability is a measure of the degree to which a system is in an operable condition when required at some random point in time. Again, a superior design is pointless unless the system can meet specified minimum levels of reliability and availability.

Economic factors provide arguably the most compelling evidence to support the focus on life cycle as opposed to product. In short, a life cycle focus can save money in the long term. Experience has shown that a large proportion of total life cycle cost for a given system stems from decisions made early in the Acquisition Phase of the project. Some 60% of errors in system development originate in the requirements analysis process [4]. To that end, the maximum opportunity to reduce the total life cycle cost of a system is presented in the Acquisition Phase. Poor requirements cannot be rectified by good design, so that it invariably follows that rigorous development of requirements is essential for the acquisition to be successful.

The life cycle illustrated in Figure 1.1 shows the phases and activities in sequence and is not intended to represent any particular process model such as the *waterfall, evolutionary development, incremental development, reusable components, technology application, reverse engineering, spiral, vee or evolutionary acquisition* models [5]. These models represent different approaches to implementing the activities of the system life cycle in Figure 1.1.

### 1.2.1 Acquisition Phase

Figure 1.1 shows that the Acquisition Phase comprises the four main activities of Conceptual Design, Preliminary Design, Detailed Design and Development, and Construction and/or Production.
1.2.1.1 Conceptual Design

The initial systems engineering effort is aimed at producing a clearly defined set of user requirements at the system level. This initial effort is referred to as Conceptual Design and represents the efforts to articulate the system design in functional terms. Although clearly defining the functional requirements of the system would seem a logical (and essential) first step, it is often poorly done and is most often the direct cause of problems later in the development process. Customers sometimes prefer to describe their needs in a loose and ambiguous manner to protect themselves from changes and developments in their needs. The Conceptual Design process aims to avoid this ambiguity and establishes what is called a Functional Baseline (which describes the whats and whys of the system). The Functional Baseline represents a system-level functional architecture that meets the customer needs. Conceptual Design can therefore be considered to be another term for functional design.

1.2.1.2 Preliminary Design

With the initial Functional Baseline established, the Preliminary Design process can commence. The aim of Preliminary Design is to convert the Functional Baseline into a preliminary definition of the system configuration or architecture (the hows of the system). The preliminary definition of the system configuration represents the initial attempt at physical design. Preliminary Design is therefore the stage where functional design is translated into physical design. This translation occurs through an iterative process of requirements analysis, synthesis (or design), and evaluation. The result of the Preliminary Design process is a subsystem-level design known as the Allocated Baseline. Note that the Allocated Baseline indicates that the functional requirements (defined in the Functional Baseline) have now been grouped together logically and allocated to subsystem-level components, which combine to form the overall system design. The Allocated Baseline therefore represents a subsystem-level physical architecture that meets the needs of the functional architecture in the Functional Baseline. Traceability between the functional and physical designs is a critical cornerstone of systems engineering that must be established and maintained during Preliminary Design and subsequent stages.

1.2.1.3 Detailed Design and Development

Detailed Design and Development is the next activity of the Acquisition Phase. The Allocated Baseline developed during Preliminary Design is used in the Detailed Design and Development process to commence development of the individual subsystems and components in the system. Prototyping may occur and the system design is confirmed by test and evaluation. The result of the Detailed Design and Development process is the initial establishment of
the *Product Baseline* as the system is now defined by the numerous products (subsystems and components) making up the total system. The definition of the system at this stage should be sufficiently detailed to commence the construction and production activities.

### 1.2.1.4 Construction and/or Production

The final activity within the Acquisition Phase is *Construction and/or Production*. The Product Baseline produced during Detailed Design and Development will have been refined and finalized prior to entering this phase due to the results of the test and evaluation effort. System components will be produced in accordance with the detailed design specifications and the system is ultimately constructed in its final form. Formal test and evaluation activities will be conducted to ensure that the final system configuration meets its intended purpose. Configuration management activities called *configuration audits* will confirm that the system (as produced) agrees with the documentation comprising the Product Baseline prior to full-scale production occurring. At the successful completion of the configuration audits, the Product Baseline for the system is said to be approved or in place.

### 1.2.2 Utilization Phase

On delivery, the system moves into the Utilization Phase (the final process within the system life cycle prior to disposal). The major activities during this phase are operational use and system support. Systems engineering activities may continue during the Utilization Phase to support any modification activity that may be required. Modifications may be necessary to rectify performance shortfalls, to meet changing operational requirements or external environments, or to enhance current performance or reliability. Another common reason for modifications is to enable ongoing support for the system to be maintained.

Following operational use and system support, the system is eventually phased out and retired from service completing the system’s entire life cycle, which may take many years to complete.

#### Example 1.3: The RAAF F-111 Aircraft

An excellent example of an extended life cycle is the F-111 aircraft currently being operated by the Royal Australian Air Force (RAAF). Following a mid-life upgrade in the 1990s involving both avionics and engine modifications, the RAAF expects the F-111 to remain in service until 2010. Since the origins of the F-111 can be traced back to the 1950s, the life cycle of the RAAF F-111s may eventually exceed 60 years.
1.3 WHAT IS SYSTEMS ENGINEERING?

There is a wide range of systems engineering definitions, each of which tends to reflect the particular focus of its source. The following are some of the more accepted and authoritative definitions of systems engineering from recent standards and documents.

“Systems engineering is the management function which controls the total system development effort for the purpose of achieving an optimum balance of all system elements. It is a process which transforms an operational need into a description of system parameters and integrates those parameters to optimize the overall system effectiveness.” [6]

“An interdisciplinary collaborative approach to derive, evolve, and verify a life cycle balanced system solution which satisfies customer expectations and meets public acceptability.” [7]

“An interdisciplinary approach encompassing the entire technical effort to evolve and verify an integrated and life cycle balanced set of system, people, product, and process solutions that satisfy customer needs. Systems engineering encompasses: the technical efforts related to the development, manufacturing, verification, deployment, operations, support, disposal of, and user training for, system products and processes; the definition and management of the system configuration; the translation of the system definition into work breakdown structures; and development of information for management decision making.” [8]

“Systems engineering is the selective application of scientific and engineering efforts to: transform an operational need into a description of the system configuration which best satisfies the operational need according to the measures of effectiveness; integrate related technical parameters and ensure compatibility of all physical, functional, and technical program interfaces in a manner which optimizes the total system definition and design; and integrate the efforts of all engineering disciplines and specialties into the total engineering effort.” [9]

“Systems engineering is an interdisciplinary, comprehensive approach to solving complex system problems and satisfying stakeholder requirements.” [10]

Although each of these definitions has a slightly different focus, a number of common themes are evident and are described in the following sections.

1.3.1 Requirements Engineering

The complete and accurate definition of system requirements is a primary focus of the early systems engineering effort. The life cycle of a system begins with a simple statement of need, which is translated into a large number of statements of requirement that form the basis for the functional design and subsequently the physical architecture. These transitions must be
managed by a rigorous process that guarantees that all relevant requirements are included (and all irrelevant requirements excluded). The establishment of correct requirements is fundamental to the success of the subsequent design activities.

Once requirements have been collected, the systems engineering process then focuses on the management of these requirements from the system level right down to the lowest constituent component. This requirements engineering (sometimes referred to as requirements management or requirements flowdown) involves elicitation, analysis, definition and validation of system requirements. Requirements engineering (described in more detail in Section 7.1) ensures that a rigorous approach is taken to the collection of a complete set of unambiguous requirements from the stakeholders.

Requirements traceability is also an essential element of effective management of complex projects. Through traceability, design decisions can be traced from any given system-level requirement down to a detailed design decision (forward traceability). Similarly, any individual design decision must be able to be justified by being associated with at least one higher-level requirement (backwards traceability). This traceability is important since the customer must be assured that all requirements can be traced forward and can be accounted for in the design at any stage. Further, any aspect of the design that cannot be traced back to a higher-level requirement is likely to represent unnecessary work for which the customer is most probably paying a premium. Traceability also supports the change process, especially the investigation of change impact.

Support for requirements traceability is a feature of the top-down approach that provides a mechanism by which it can be guaranteed that requirements can be satisfied at any stage. A bottom-up approach cannot provide the same guarantee.

1.3.2 Top-down Approach

Traditional engineering design methods are based on bottom-up approach in which known components are assembled into subsystems from which the system is constructed. The system is then tested for the desired properties and the design is modified in an iterative manner until the system meets the desired criteria. This approach is valid and extremely useful for relatively straightforward problems that are well defined. Unfortunately, complex problems cannot be solved with the bottom-up approach.

Systems engineering begins by addressing the complex system as a whole, which facilitates the initial allocation of requirements as well as the subsequent analysis of the system and its interfaces. Once system-level requirements are understood, the system is then broken down into subsystems and the subsystems further broken down into components until a complete understanding is achieved of the system from top to bottom. This top-down
approach is a very important element of managing the development of complex systems. By viewing the system as a whole initially and then progressively breaking the system into smaller elements, the interaction between the components can be understood more thoroughly, which assists in identifying and designing the necessary interfaces between components (internal interfaces) and between this and other systems (external interfaces). For example, Figure 1.2 illustrates the ANSI/EIA-632 approach to top-down development. [11]

It must be recognized, however, that while design is conducted top down the system is implemented using a bottom-up approach. That is, the aim of system engineering can be considered to be to provide a rigorous, reproducible process by which the complex system can be broken into a series of simple components that can then be designed using the traditional engineering bottom-up approach. Importantly, the other principal facet of systems engineering is to provide a process by which the components and subsystems can be integrated to achieve the desired system properties.

Note that, as discussed in Section 1.1, the terms *system*, *subsystem* and *component* are relative. Each system comprises subsystems that consist of components. Each subsystem, however, can be considered to be a system in its own right, which has subsystems and components, and so on. Consequently, while the building-block concept for top-down development is very useful, it is often a source of confusion among novices due to the relative nature of the associated terms.

![Figure 1.2. ANSI/EIA-632 building block concept for top-down development.](image)
1.3.3 Focus on Life Cycle

Systems engineering is focused on the entire system life cycle and takes this life cycle into consideration during decision-making processes. As described in Section 1.2, a system’s life cycle begins during system definition and design, and passes through construction and/or production, operation, maintenance, support and phase-out. The life cycle concludes only with the disposal of the system.

In the past it has been too common to consider design options only in the light of the issues associated with the Acquisition Phase and to pay little attention to through-life support issues. Project teams typically focus on the Acquisition Phase of the project and on the development of a system that meets the functional user requirements while minimizing cost and schedule. This has often led to larger-than-expected costs in the Utilization Phase to be met from budgets that are insufficient to keep systems in service. A life cycle focus requires a system focus not a product focus. A system focus takes into account all constituent elements of the system including operation, maintenance and support, and retirement or disposal.

Example 1.4: Life-cycle Focus

As a simple analogy to demonstrate the concept (and problems) of a product focus as opposed to a life-cycle focus, consider a motor vehicle manufacturer that has introduced a new model. The new family sedan is a roomy, good-looking, high-performance vehicle that is available at three-quarters of the price of similar models. Strong interest seems assured with buyers attracted by the vehicle’s features as well as the cost-effectiveness of the purchase.

On further investigation, however, it transpires that the low purchase price has been achieved by a design that incorporates poor-quality components that require iterative, regular maintenance resulting in running costs that are five times higher than competing models.

Clearly, a decision to purchase the new vehicle must focus on more than just the purchase price of the vehicle and must take into account the system life-cycle aspects of operation, maintenance and support.

1.3.4 System Optimization and Balance

As we discuss in Chapter 3, it does not necessarily follow that the combination of optimized subsystems leads to an optimized system. Additionally, the system architecture must represent a balance between the large number of requirements that, as well as the technical considerations, cover a wide range of factors such as environmental, economic human factors, moral, ethical, social, cultural, psychological, and so on. A further advantage, therefore, of the top-down approach in systems engineering is that system optimization and balance can be achieved as a byproduct of the design process, something that cannot be guaranteed in a bottom-up design method.
A balance must also be struck across the life cycle. Metrics such as cost-effectiveness must be measured across all phases, not just acquisition.

1.3.5 Integration of Disciplines and Specialties

Systems engineering aims to manage and integrate the efforts of a multitude of technical disciplines and specialties to ensure that all user requirements are adequately addressed. Rarely is it possible for a complex system to be designed by a single discipline. Consider our aircraft example. While aeronautical engineers may be considered to have a major role, the design, development and production of a modern aircraft system requires a wide variety of other engineering disciplines including electrical, electronics, communications, radar, metallurgical, and corrosion engineers. Of course, in system terms, other engineering disciplines are required for testing and for logistics and maintenance support as well as the design and building of facilities such as runways, hangars, refueling facilities, embarkation and disembarkation facilities, and so on. Other non-engineering disciplines are involved in marketing, finance, accounting, legal, environmental, and so on. In short, there could be hundreds, even thousands, of engineers and members of other disciplines involved in the delivery of an aircraft system.

The aim of the systems engineering function is to break up the task into components that can be developed by these disparate disciplines and specialties and then provide the management to integrate their efforts to produce a system that meets the users’ requirements. In modern system developments, this function is all the more important because of the complexity of large projects and their contracting mechanisms, and the geographic dispersion of contractor and subcontractor personnel across the country and around the world.

1.3.6 Management

While systems engineering clearly has a technical role and provides essential methodologies for systems development, it is not limited simply to technical issues and is not simply another engineering process to be adopted. As we discuss throughout this text, systems engineering has both a management and a technical role. Project management is responsible for ensuring that the system is delivered on-time, within-budget and meets the customers expectations. The trade-offs and compromises implicit in those functions are informed by the products of systems engineering. Systems engineering and project management are therefore inextricably linked. These issues are discussed in more detail in Chapter 8.
1.4 SYSTEMS ENGINEERING RELEVANCE

Systems engineering principles and processes are applicable (albeit to varying degrees) to a wide range of projects. For example, ANSI/EIA-632 [12] states that the standard itself is intended to be applicable to:

“the engineering or the reengineering of
a) commercial or non-commercial systems, or part thereof;
b) any system, small or large, simple or complex, software-intensive or not, preceded or unprecedented;
c) systems containing products made up of hardware, software, firmware, personnel, facilities, data, materials, services, techniques, or processes (or combinations thereof);
d) a new system or a legacy system, or portions thereof.”

Because it is difficult to imagine a project that does not fit into the above description, it is critical to understand the merits of systems engineering and apply them in a tailored manner, cognizant of the relative size, complexity and risks associated with each undertaking. At one end of the spectrum are large complex projects making use of leading-edge developmental technology. These projects typically involve large sums of money, long time scales and significant risks. At the other end of the spectrum are small projects making use of extant techniques and existing technology. These projects typically involve short periods, low costs and involve a minimum of risk. Clearly different levels of systems engineering are applied to each of these types of projects.

The most obvious application of systems engineering principles and methodologies is in projects that are large and complicated. However, smaller and less complex projects can also benefit from the application of systems engineering principles.

Example 1.5: Systems Engineering Relevance to Large Projects

A classic example cited by the Software Engineering Institute in their Systems Engineering Capability Maturity Model (SE-CMM), Version 1.0 [13] demonstrates the need for good systems engineering. The Tacoma Narrows bridge was a long suspension bridge with a flexible roadway that collapsed in 1940 due to strong winds that had set up an aerodynamic oscillation.

It is believed that the problems associated with the road could have been avoided or managed more effectively had systems engineering principles and processes been applied rigorously during the project. While adhering to good systems engineering practices would not necessarily ensure that designers are more likely to know the parameters of a particular problem, rigorous developmental approaches are more likely to draw out the relevant risk areas of a system (such as environment-related risks in this case).
The failures of system developments are often attributed to misunderstandings, ambiguities, misinterpretations, errors and omissions in what the contractor is attempting to deliver. The result is a system that fails to solve the customer’s problem leading to a breakdown in the relationship between contractor and customer. Systems engineering specifically targets these problems and therefore is relevant to both parties in any system development.

Within a customer’s project office, systems engineering is particularly relevant to the technical personnel who are responsible for the application of systems engineering principles as part of the overall project management effort. In most customer organizations, systems engineering personnel report to the project manager. The systems engineer is in an excellent position to apply the tools of the systems engineering to assist the project manager in each of the nine project management knowledge areas (discussed in more detail in Chapter 8). Systems engineering is therefore an essential element of the project manager’s ability to acquire a quality system within budget, time and scope constraints.

Contractors should also be focused on developing processes and best practices to support the delivery of superior products and services to their customers. Systems engineering offers tools and philosophies that support the consistent development of quality products and services.

1.5 SYSTEMS ENGINEERING BENEFITS

There are a number of potential benefits from the successful implementation of systems engineering processes and methodologies.

The first and most visible benefit is the scope for saving money during all phases of the system life cycle—life-cycle cost (LCC) savings. While some may argue that the additional requirements imposed by systems engineering can increase costs, these increases are generally felt in the very early design phases. If applied appropriately, systems engineering can ensure that the savings achieved far outweigh the cost of implementing appropriate procedures and methodologies. Experience indicates that an early emphasis on systems engineering can result in significant cost savings later in the construction and/or production, operational use and system support, and disposal phases of the life cycle [14].

Systems engineering should also assist in reducing the overall schedule associated with bringing the system into service. Systems engineering ensures that the user requirements are accurately reflected in the design of the system helping to minimize costly and time-consuming changes to requirements later in the life cycle. If changes are required, they can be incorporated early in the design and in a controlled manner. The rigorous consideration and evaluation of feasible design alternatives during the design phases of the project promote greater design maturity earlier.
System failures, cost overruns and schedule problems are often the direct result of poor requirements-management practices. The systems engineering discipline aims to put in place a rigorous process of requirements management to produce well-defined requirements, adequate levels of traceability between the different levels of technical design documentation back to the original user requirements, and requirements which are both verifiable and consistent. This requirements-management process must achieve these results without pre-supposing a particular technical solution or placing unnecessary technical constraints on the solution.

Figure 1.3 illustrates the impact of systems engineering on the system life cycle. For more-complex system developments, the profile may be more complex than Figure 1.3 especially when the development spans many years and early production systems are used to refine the functionality and performance of later versions. Note that systems engineering has its greatest impact through the rigorous application of processes and methodologies during the early stages of the project where the ease of change and cost of modification is the lowest. In fact, the curve in Figure 1.3 could be relabeled as the ease with which changes can be made throughout the system life cycle. Because systems engineering has the largest impact during the early stages of the process, in this book we concentrate primarily on the conceptual, preliminary and early detailed design processes.

Systems engineering leads to a reduction in the technical risks associated with the product development. Risks are identified early and monitored throughout the process using a system of technical performance measures, and design reviews and audits. Design decisions can be traced back to the original user requirements and conflicting user requirements can be identified and clarified early, significantly reducing the risk of failure later in the project.
Finally, the disciplined approach to requirements engineering leads to a product that meets the original intended purpose more completely. This improved functional performance makes for a quality system where quality is measured by the ability of the system to meet the documented requirements.

1.6 ANALYSIS, SYNTHESIS AND EVALUATION

All extant systems engineering standards and practices extol processes that are built around an iterative application of analysis, synthesis and evaluation. The iterative nature of the application is critical to the systems engineering processes. Initially the process is applied at the systems level. It is then reapplied at the next level of detail and so on until the entire development process is complete. During the earlier stages the customer is heavily involved; in the latter stages, the contractor is mainly responsible for the continuing effort, which is monitored by the customer.

Prior to detailing the individual activities within the systems engineering processes, it is worth considering the basic foundations of the analysis-synthesis-evaluation loop illustrated in Figure 1.4. This concept is neither new nor complex; it is simply a good, sound approach to problem solving. While applicable in any domain, the loop is particularly fundamental to systems engineering.

1.6.1 Analysis

Analysis commences with a statement of perceived need or a set of customer requirements. During Conceptual Design, analysis investigates these needs and requirements, and identifies the essential functions that the system must perform in order to meet the needs. Requirements analysis at the system-level aims to answers the what, how well and the why questions relative to the system design. Analysis activities continue throughout the subsequent stages of the life cycle to help in defining lower-level requirements often called derived requirements associated with physical aspects of system design.

A basic tool that can be used during analysis is the functional flow block diagram (FFBD), which shows the logical sequences and relationships of the functions of the system at the system level. The system-level functions

![Figure 1.4. The analysis-synthesis-evaluation iteration.](image-url)
detailed in the FFBD are further developed at subsystem level and below using more detailed FFBDs. Examples of FFBDs are presented in Chapters 2 and 3. As the design progresses through the different design phases and progressively becomes more detailed, the system-level functions will be individually investigated and have an FFBD produced and so on until a very detailed set of functions and their interrelationships have been identified.

Once all the functions associated with the system have been identified, the requirements associated with each function can be defined. These requirements could include performance parameters such as speed, altitude, accuracy; interoperability requirements detailing other systems with which the system under development must operate; and interface requirements to describe the necessary outputs expected from the system and the inputs to the system. Depending on the particular design phase, these requirements and functions may also be grouped in accordance with some sort of logical criteria and then allocated to a particular physical component of the system. That is, the component becomes responsible for the satisfaction of those requirements by performing the functions assigned to it. The allocation of requirements forms a description of the system elements and architecture and therefore assists in the process of synthesis or design (answering the how questions).

1.6.2 Synthesis

The analysis activity resolves what is required as well as how well, and why. Synthesis, or design, now determines how. Synthesis is possibly the most widely recognized role of a professional engineer. Synthesis is the process where creativity and technology are combined to produce a design that best meets the stated system requirements. The term synthesis is more appropriate than design in the systems engineering context as it hints at the evolutionary nature of design and development.

In the early stages of the systems engineering processes, synthesis is limited to defining completely the functional design of the system and then considering all possible technical approaches using the results from the requirements-analysis effort. From this consideration, the best approach is selected and the process moves to the next level of detail. Later in the systems engineering processes, the selected design concept is synthesized further until, ultimately, the complete system design is finalized.

Although synthesis is the creative part of the systems engineering effort, a number of tools may be used to ensure that all alternatives are considered and the most suitable alternative is ultimately selected. Some potential aids to assist the design engineer during the synthesis process are presented in Chapter 7.
1.6.3 Evaluation

System cost and risk are directly associated with requirements and design. Evaluation is the process of investigating the trade-offs between requirements and design, considering the design alternatives, and making the necessary decisions. The process of evaluation continues throughout all stages of the systems engineering effort ultimately determining the system’s satisfaction of original requirements. Trade-off analysis is one of the tools available to the system designer in performing evaluation of competing requirements or designs—a detailed treatment of trade-off analysis is provided in Chapter 7.

The outcome of the evaluation is a selection or confirmation of the desired approach to design. Discrepancies are also identified if applicable and may result in further analysis and synthesis as the analysis-synthesis-evaluation loop is closed.

1.7 A SYSTEMS ENGINEERING FRAMEWORK

Discussions on systems engineering become complicated by the broad mandate of the system, the complexity and interrelationship of the many systems engineering constituents, and the relationships with other disciplines throughout the entire system life cycle.

The ability to understand a complex subject such as systems engineering is greatly enhanced by a solid framework within which concepts can be considered. An example of such a framework is the Project Management Body of Knowledge (PMBOK) [15] that provides a clear framework within which to consider the many facets of project management. Without an equivalent framework, the broad scope of systems engineering soon becomes confusing given the complexity of its components and their many interrelationships. There are a number of excellent systems engineering standards available today that contribute to the elements of a suitable framework, but each standard contains complexity, terminology and detail that requires substantial interpretation. The entry level of many students, young engineers and project managers therefore does not allow the use of such standards as effective frameworks within which to examine systems engineering.

A systems engineering framework [16] (illustrated in Figure 1.5) has been synthesized by the authors through a thorough survey of existing systems engineering publications and standards, and through experience in teaching systems engineering at a range of levels. The main aim of the framework is to provide a simple construct within which the systems engineering discipline can be understood and implemented.
Chapter 1  Introduction to Systems Engineering

19

SYSTEMS ENGINEERING PROCESSES
Conceptual design tasks
Preliminary design tasks
Detailed design tasks
Construction production tasks
Utilisation related tasks

SYSTEMS ENGINEERING MANAGEMENT

RELATED DISCIPLINES

TOOLS

Figure 1.5. A framework for the consideration of systems engineering.

The framework illustrates the relationship of the three main elements of systems engineering processes, management and tools and places them in context with related disciplines. In common use, the terms systems engineering management and systems engineering processes are sometimes used interchangeably. Here we make a distinction between the two. We present the engineering processes as being the hows of systems engineering, the application of which forms the foundation of the systems engineering effort. Over the top of these processes sits the systems engineering management function, which is responsible for directing the systems engineering effort, monitoring and reporting that effort to the appropriate areas, and reviewing and auditing the effort at critical stages in the entire process. These two elements are supported by a range of tools and all elements interface with related disciplines such as traditional engineering disciplines, project management, integrated logistics support, quality assurance.

The systems engineering framework provides an excellent structure within which to examine and explain the complex discipline of systems engineering. Experience in undergraduate and graduate courses as well as commercial short courses [17] has shown that the framework provides an excellent means of communicating the complexities and interrelationships of the systems engineering discipline, particularly to those who do not have a significant amount of project experience. Students can successfully grasp the fundamental concepts of systems engineering within a relatively short timeframe.

1.7.1 Systems Engineering Processes

The basic systems engineering process is the analysis-synthesis-evaluation loop described in Figure 1.4. This loop is applied iteratively throughout the system life cycle, as illustrated in Figure 1.6.
Other systems engineering processes and tasks are divided into the life-cycle stages within which they typically occur. In this book we do not attempt to detail exhaustively all systems engineering processes. Instead, we concentrate on the intent and main aim of each phase, and examine some of the likely techniques that may be used to arrive at that aim. We place particular emphasis on the Acquisition Phase of the life cycle, as it is the phase during which systems engineering has the ability to have the most impact on a system.

In Chapter 2, for example, the tasks completed during Conceptual Design are shown to focus on achieving a clear and complete definition of the system-level requirements. We investigate the processes of articulation of the system needs, goals and objectives; system feasibility analysis; system requirements analysis; system synthesis; and so on. Chapter 3 investigates the processes associated with Preliminary Design and Chapter 4 describes those related to Detailed Design and Development and Construction and/or Production.

1.7.2 Systems Engineering Management

Systems engineering management sits over the top of systems engineering processes and is responsible for directing the systems engineering effort, monitoring and reporting that effort to the appropriate areas, and reviewing and auditing the effort at critical stages in the entire process. In Chapter 5, we address the major systems engineering management elements of technical reviews and audits, system test and evaluation, technical risk management, configuration management, the use of specifications and standards, integration management, and systems engineering management planning.

The pre-eminent position of systems engineering management in the framework illustrates that it is the key to the entire systems engineering effort.
1.7.3 Systems Engineering Tools

Many tools exist to assist systems engineering processes and management. These tools range from techniques and methods through to systems engineering standards. Here we describe the most popular tools and standards under the headings of management tools (Chapter 6) and process tools (Chapter 7). It is not the intended purpose of this book, however, to repeat information contained in standards and documents elsewhere.

Throughout the book we present generic process tools such as requirements breakdown structures (RBS), functional flow block diagrams (FFBD), work breakdown structures (WBS), trade-off analysis, as well as prototyping and simulation as examples of tools that may be applied to the systems engineering process effort. We also describe the systems engineering management tools of standards and capability maturity models. In Chapter 6, the current standards are reviewed and summarized including MIL-STD-499B [18], EIA/IS-632 [19], IEEE 1220 [20] and ANSI/EIA-632 [21]. The SE-CMM [22] is used as an example capability maturity model (CMM) in order to describe how CMMs can be used to assist the overall systems engineering management effort. We also briefly discuss the CMMI.

1.7.4 Related Disciplines

There are many disciplines (both technical and non-technical) related to systems engineering. Examples include project management, logistics management, quality assurance, user-requirements management, software engineering, hardware engineering, and interface engineering (or integration engineering). Related disciplines can be considered to be the “glue” that holds together the other components of the framework.

The relationship between the related disciplines and the other facets of systems engineering depends very much upon the discipline in question. Some (such as project management) oversee the whole systems engineering discipline, while others (such as hardware and software engineering) sit between systems engineering management and the processes, and others (such as quality assurance) sit alongside the systems engineering effort. Chapter 8 discusses these disciplines and their relationship with systems engineering.
1.8 REVISION QUESTIONS

1. Briefly describe a system and list the major resources that make up a system in the context of systems engineering.

2. Briefly describe the system life cycle (as proposed by Blanchard and Fabrycky) and explain briefly the two main phases and the activities that occur within each.

3. Definitions of systems engineering abound, but all agree on the key focuses of the discipline. List and briefly describe the key focuses of systems engineering.

4. Describe some of the benefits of using systems engineering discipline during the development of a technical system.

5. A system can be described functionally and physically. Explain what each description provides and describe the relationship between the two descriptions.

6. Explain why you think systems engineering could be described as applying from "the cradle to the grave".

7. In terms of life-cycle costs, explain why systems engineering is so interested in the Utilization Phase of a system’s life cycle.

8. Assuming the role of both acquirer and supplier of systems, explain why you would apply systems-engineering methods to your respective tasks.

9. Describe diagrammatically the classic analysis-synthesis-evaluation process that is applied iteratively throughout the system life cycle, and explain each of the components.

10. Systems engineering is often perceived to add unnecessary complexity, cost and time to development projects. Playing the systems engineering advocate, comment on why this criticism might be leveled at systems engineering and explain why the perception is short-sighted.

ENDNOTES

[1] Further reading on systems engineering can be obtained from the following major sources:


