1

SYSTEMS, ENGINEERING, AND SYSTEMS ENGINEERING

Principle 1.1

SE Alpha–Omega Principle

SE *begins* and *ends* with the Users of a system, product, or service.

Have you ever purchased a commercial hardware and/or software product; contracted for development of a system, product, or service; or used a website and discovered that it:

- May have complied with its specification requirements but was not what you wanted, needed, or expected?
- Was difficult to use, unforgiving in accepting User inputs and errors, and did not reflect your thought patterns of usage?
- Consisted of an overwhelming number of *non-essential* features that were so distracting it was difficult to navigate?
- Buried commonly used features under several layers of linkable structures requiring numerous mouse clicks to reach and invoke?
- Has software updates that are incompatible with standard operating systems. The System Developer's customer service response was to post a question in an online "community forum." Then, wait (potentially forever) for some other "community User" to offer a solution as to how they solved the System Developer's problem?

Then, in frustration, you and millions of other Users question whether the System Developer and its designers ever bothered to communicate with and listen to the Users or marketplace to understand and comprehend:

- The jobs or missions the User is expected to perform to deliver the system's outcomes to their customers
- How the User expects to deploy, operate, maintain, sustain, retire, or dispose of the systems, products, services, or by-products required to perform those jobs or missions

Welcome to Systems Engineering (SE)—or more appropriately the lack of SE. If you talk with Users such as the ones in the examples above, you will often hear comments such as:

- Company XYZ needs to do a better job "Engineering" their systems, products, or services!
- System ABC needs some "SE!"

From an SE perspective, what emerges from a distillation of User comments are questions such as: *What is SE*? Answering this question requires understanding (1) *what is a system* and (2) *what is Engineering*. Then, *what is the interrelationship between Engineering and SE*?

System Engineering Analysis, Design, and Development: Concepts, Principles, and Practices, Second Edition. Charles S. Wasson.

Companion Website: www.wiley.com/go/systemengineeringanalysis2e

Opinions vary significantly for definitions of these terms and their context of usage. Industry, government, academia, professional, and standards organizations have worked for years to reach consensus definitions of the terms. To achieve a consensus—global in some cases—the wording of the definitions becomes so diluted and abstract that it has limited utility to the User communities the organizations serve. In some cases, the abstractness distorts User perceptions of what the terms truly encompass. For example, the definition of a *system* is a classic example.

The problem is exacerbated by a general lack of true Systems Engineering & Development (SE&D) courses that focus on *problem-solving and solution development* methods and Engineering. Unfortunately, many of the so-called SE courses focus on: (1) System Acquisition & Management - how to manage the acquisition of systems and (2) equation-based courses – "Engineering the box," not the system. This results in a major deficiency in Engineering knowledge and skills required to actually transform a User's abstract, operational need into the Engineering of a physical system, product, or service that meets those needs. *Should there be any surprise as to why User frustrations with systems, products, or services highlighted above occur*?

Given this backdrop, Chapter 1 establishes the foundational definitions for understanding what it means to perform SE addressed in Chapters 2–34.

1.1 DEFINITIONS OF KEY TERMS

- **Capability—**An explicit, inherent feature *initiated* or *activated* by an external stimulus, cue, or excitation to perform an action (function) at a specified level of performance until terminated by external commands, timed completion, or resource depletion.
- **Engineering—**"[T]he profession in which knowledge of the mathematical and natural sciences gained by study, experience, and practice is applied with judgment to develop ways to utilize economically the materials and forces of nature for the benefit of mankind" (Prados, 2007, p. 108).
- **Entity**—A generic term used to refer to an operational, logical, behavioral, physical, or role-based object within a system. Physical entities, for example, include PERSONNEL; EQUIPMENT items such as SUBSYSTEMS, ASSEMBLIES, SUBASSEMBLIES, or PARTS comprised of HARDWARE and/or SOFTWARE; PROCEDURAL DATA such as User's guides and manuals; MISSION RESOURCES such as *consumables* (water, fuel, food, and so on) and *expendables* (filters, packaging, and so on); and SYSTEM RESPONSES—performance-based outcomes—such as products, by-products, or services or FACILITIES.
- **Environment**—A general, context-dependent term representing the NATURAL, HUMAN SYSTEMS, or INDUCED Environments that in which a SYSTEM or ENTITY of Interest must operate and survive.
- **Ilities—**Specialty Engineering disciplines such as Reliability, Maintainability, and Availability (RMA); Sustainability; Safety; Security; Logistics; and Disposal.
- **System—**An integrated set of interoperable elements or entities, each with specified and bounded capabilities, configured in various combinations that enable specific behaviors to emerge for Command and Control (C2) by Users to achieve performance-based mission outcomes in a prescribed operating environment with a probability of success.
- **System Engineering (SE)**—The multidisciplinary application of analytical, mathematical, and scientific principles to formulating, selecting, developing, and maturing a solution that has acceptable risk, satisfies User operational need(s), and minimizes development and life cycle costs while balancing Stakeholder interests.

1.2 APPROACH TO THIS CHAPTER

Our approach to this chapter focuses on defining SE. Since the term SE is comprised of *System* and *Engineering*, we begin with establishing definitions for both of these terms as a precursor for defining SE.

Most definitions of a *system* are often too abstract with limited utility to the User. This text defines a *system* in terms of its attributes and success criteria—what a system is, why it exists, its compositional structure, what it accomplishes, under what conditions, and User expectations for success. Although systems occur in a number of forms such as Enterprise, social, political, and equipment, we focus on Enterprise and Engineered Systems.

One of the challenges in discussing systems is the need to differentiate systems, products, or services. We address those differences and relationships and provide examples. When systems are developed, they may be (1) new innovations (*unprecedented* systems) based on new or emerging technologies or (2) improvement on existing systems or technologies (*precedented* systems). We address the contexts of *unprecedented* versus *precedented* systems.

Based on establishment of what a *system* is, we introduce a commonly accepted definition of *Engineering* and then derive the definition of SE used in this text. Since people often are confused by the usage of *System* versus *SE*, we delineate the context of usage for these terms.

An introduction to SE is incomplete without some form of background description of its history. Rather than repeating a set of dates and facts that have been documented in other texts, a more important point is understanding what has driven the evolution of SE. To address this point, we introduce an excellent source of SE history for those who want more detailed information. We elaborate this topic in more detail in Chapter 2.

Finally, we close Chapter 1 with a discussion of a key attribute of Systems Engineers—Systems Thinking.

Before we begin, a brief word concerning individuals and teams crafting statements—definitions, specification requirements, and objectives—to achieve consensus agreement is as follows:

When people or organizations develop definitions, attempts to simultaneously create *content* and *grammar* usually produce a result that only has a degree of acceptability. People typically spend a disproportionate amount of time on *grammar* and spend very little time on substantive *content*. We see this in development of plans and specifications, for example. Grammar is important, since it is the root of any language and communications. However, grammar is simply a mechanism to convey: (1) *content* and (2) *context*. "Word-smithed" grammar has *little or no value if it lacks substantive content or context*.

You will be surprised how animated and energized people become during grammar "word-smithing" exercises. After protracted discussions, others simply walk away from the chaos. For highly diverse terms such as a *system*, a good definition may begin with simply a bulleted list of descriptors concerning what a term *is* or *is not*. If you or your team attempt to create a definition, perform one step at a time. Obtain consensus on the key elements of *substantive content*. Then, structure the statement in a logical sequence and translate the substantive content into a grammar statement.

Let's begin our discussion with *What Is a System*?

1.3 WHAT IS A SYSTEM?

Merriam-Webster (2014) states that the term *system* originates from "late Latin *systemat-, systema,* from Greek *systēmat-, systēma,* from *synistanai* to combine, from *syn-* + *histanai* to cause to stand." Its first known use was in 1603.

There are as many definitions of a *system* as there are opinions. Industry, government, academia, and professional organizations over many decades have worked on defining what a system is in their context. If you analyze many of these definitions, most of the definitions have become so diluted due to "wordsmithing" to achieve a consensus of the user community, the remaining substantive content is almost nil. That is reality, not a critique! It is a very challenging task given a diverse set of views and levels of experience weighted toward those who are willing to participate.

The *definition* that emerges from these exercises accomplishes a different objective—obtain a consensus definition of what a User community believes a system is versus *what a system actually is* and *what its Users expect it to accomplish*. Additionally, the definitions are often *abstract* and *intermix* different types of information and levels of detail that may impress uninformed customers but are *technically incorrect*. Consider the following example.

Making Statements That Are Partially True but Technically Incorrect

Example 1.1

Definitions over the years loosely infer that a system is a collection of people, hardware, software, procedures, and facilities—for example, entities. Systems do encompass those entities. However, general definitions such as this crafted to achieve a consensus do not express what a system is, why it exists, who it serves, its operating conditions, required outcomes and performance, criteria for success, etc.

The intent here is not to critique established definitions. If they work for you and your organization, fine. However, let's establish a definition that expresses *what a system actually is*. This is a crucial step in establishing a foundation for understanding Chapters 2–34. Therefore, we establish the following definition of a *system*:

- **System**—An integrated set of interoperable elements or entities, each with specified and bounded capabilities, configured in various combinations that enable specific behaviors to emerge for Command & Control, C2 by Users to achieve performance-based mission outcomes in a prescribed operating environment with a probability of success.

The "system" definition above captures a number of key discussion points that define a *system*. A *system* is composed of two or more integrated entities that enable accomplishment of a higher-level purpose—emergence—that cannot be achieved by each of the entities on an individual basis. However, a *purpose* without some *measure of success*—an outcome and level of performance—has limited value to the User or its stakeholders. With the establishment of this theme as a backdrop, let's explore each of the definition's phrases individually to better understand what they encompass and communicate.

1.3.1 System Definition: "An Integrated Set of Interoperable Elements or Entities …"

Systems occur in a variety of forms that include Enterprise and Engineered Systems—equipment hardware and software, social systems, political systems, and environmental systems. This text focuses on two types of systems: Enterprise and Engineered. Let's define each of these terms:

- **Enterprise Systems**—Formal and informal industry, academic, governmental, professional, and nonprofit organizations such as corporations, divisions, functional organizations – departments such as accounting and engineering; projects, and others.
- **Engineered Systems**—Physical systems or products developed for internal use, commercial sale to the marketplace, or for contract development that require one or more Engineering disciplines and skillsets to apply mathematical and scientific principles to design and develop solutions

Since Engineered Systems are an integral part of our home and work lives, let's begin with those.

1.3.1.1 Engineered Systems Engineered Systems, in general, consist of equipment comprised of hardware and/or software, fluids (lubricants, coolants, etc.), gases, and other entities:

- Hardware entities, for example, include hierarchical levels such as PRODUCTS comprised of → SUBSYSTEMS comprised of → ASSEMBLIES comprised of → SUBASSEMBLIES comprised of → PARTS (Chapter 8).
- Software entities include hierarchical terms such as Computer Software Configuration Items (CSCIs) comprised of → Computer Software Components (CSCs) comprised of → Computer Software Units (CSUs) (Chapter 16).

1.3.1.2 Enterprise Systems Enterprise Systems are HIGHER-ORDER SYSTEMS (Chapter 9)—government, corporations, and small businesses—that:

- Employ Engineered Systems—manufacturing systems, vehicles, computers, buildings, and roads—to:
    - Produce and distribute consumer products and contract deliverable systems
    - Provide services such as retail sales; land, sea, air, or space transportation; utilities such as electrical power, natural gas, telephone, water, sanitation, and refuse; and medical, healthcare, financial, educational, and other services

As we shall see in Chapters 8 and 9, analytically:

- **Enterprise Systems** consist of hierarchical levels of abstraction (divisions, departments, branches, etc.) comprised of System Elements (Figure 8.13)—personnel, equipment (hardware and software), procedures, resources, behavioral outcomes, and facilities—that are integrated to perform Enterprise missions.
- **Engineered Systems** consist of hierarchical levels of abstraction (Figure 8.4)—SEGMENTS, PRODUCTS, SUBSYSTEMS, ASSEMBLIES, SUBASSEMBLIES, and PARTS.

Observe the terms Enterprise System *elements* and Engineered System *entities*. Application of these terms will become more important in follow-on chapters. The terms imply that these are discrete objects, which they are. However, remember the earlier point in the *system* definition … *comprised two or more entities in combination that enable accomplishment of a higher-level purpose that cannot be achieved by each of the entities on an individual basis*. The operative term *combination* means that the system elements or entities must be *integrated*—connected.

Integrating elements and entities is a *necessary* condition to leverage or exploit the combination of capabilities. However, suppose the entities are *incompatible*? Hypothetically, you could fill—*integrate*—diesel fuel or kerosene into a gasoline-based automobile's fuel tank. But that does not mean that the engine will perform. Due to the *incompatibility*, fuel station pump nozzles and vehicle fuel tank ports are purposely designed to preclude inadvertent mixing.

Being *compatible* may be a *necessary condition* for some system entities such as rigid mechanical interfaces or System Elements such as procedural consistency between equipment—hardware and software—and User or Operator Manuals. Being compatible, however, does not mean that they can communicate in a language that is intelligible and comprehensible. That leads us to the need for some systems to be … *interoperable*. Consider electronic financial transactions in which debit or credit cards, card readers, and computers must be not only *compatible* in terms of electronic protocols but also formatting in an intelligible language that enables each to understand and interpret what is being communicated—*interoperability*. For those types of systems, compatibility and interoperability are both *necessary* and *sufficient* conditions for success.

In summary, the foundation of a system begins with *an integrated set of interoperable* (*Enterprise System*) *elements* (personnel, equipment, procedures, resources, behavioral outcomes, and facilities) *or* (*Engineered System*) *entities* (PRODUCTS, SUBSYSTEMS, ASSEMBLIES, SUBASSEMBLIES, etc.). In either case, we refer to the *system* being analyzed or investigated as a System of Interest (SOI).

1.3.2 System Definition: "… Each with Specified and Bounded Capabilities …"

If a system requires *System Elements* or *entities* that are *compatible* and *interoperable*, how do we ensure that they are? This requires multi-discipline Engineering - SE - to *specify* and *bound* these operational, behavioral, and physical capabilities—*attributes*, *properties*, and *characteristics*

(Chapter 3)—via specification requirements as a starting point.

Observe usage of the term *capability*. Traditionally, the term *function*—as in *form, fit, and function*—has been used by Engineers to characterize *what* a system is expected to accomplish. However, there is a gross disparity between the true definition of a *function* and what the User expects the system to *accomplish*. Here's the difference.

A Word of Caution 1.1

Form, Fit, and Function: An Implied Catch Phrase for Failure!

The phrase *form, fit, and function*, which is deeply ingrained as a paradigm in everyday Engineering, is a well-intended concept that is subject to misinterpretation. By virtue of the sequence of terms, people sometimes interpret the phrase as the sequence of steps required to perform Engineering:

- Step 1—Design the physical system—*form*.
- Step 2—Figure out how to get the pieces to *fit* together.
- Step 3—Decide what the system must do—*function*.

Evidence of this paradigm is illustrated in Figure 2.3. PURGE the *form, fit, and function* paradigm from your mind-set! The phrase simply identifies three key attributes of a system, product, or service that must be considered, nothing more!

Simply stated, a *function* represents an *action* to be performed such as Perform Navigation. A function is a *unitless* term that does not express a level of performance to be achieved. In general, it is easy to "identify functions" via *functional analysis*—sounds impressive to uninformed customers. The challenge is specifying and bounding the level of performance a function must achieve. Although *functions* and *functional analysis* are certainly valid within their own context, from a current SE perspective, the concept of functional analysis as a primary driving SE activity is outdated. The reality is functional analysis is still valid but only as a supporting SE activity. So, *how do we solve this dilemma*?

The solution resides in the term *capability*. A capability is defined as follows:

- **Capability**—An explicit, inherent feature *activated* by an external stimulus, cue, or excitation to perform an action (function) at a specified level of performance until terminated by external commands, timed completion, or resource depletion.

From an Engineering perspective, think of a capability using a vector analogy. A *capability* (vector) is characterized by a *function* (direction) and a *level of performance* (magnitude).

In summary, this text replaces *functions* and *functional analysis* with more appropriate terms *capability* and *capability analysis*.

1.3.3 System Definition: "… Configured in Various Combinations That Enable Specific Behaviors to Emerge …"

Configuration of various combinations of System Element and entity capabilities to produce system responses for a given set of system inputs—stimuli, cues, and excitations—represents a system architecture. However, system responses vary based on the User's operational needs at different times during a mission. Consider the following example of an aircraft.

Example 1.2

Aircraft Configurations and Behaviors

For an aircraft to perform a mission, it must be capable of loading passengers and cargo; taxiing; performing phases of flight (taking off, climbing, cruising, holding, and landing); and unloading passengers to accommodate various Use Cases and Scenarios (Chapter 5). Each of these activities requires unique sets of capabilities—*architectural configurations*—provided by the (Enterprise System) Elements and (Engineered System) entities to accomplish the performance-based mission outcomes and objectives.

Observe the phrase "… *enable specific behaviors to emerge* … ." Emergent behavior is a key attribute of systems that enables them to accomplish a higher-level purpose that cannot be achieved by the individual elements or entities. In general, *emergent behavior* means that the system exhibits behaviors that are not readily apparent from analysis of its individual elements or entities. Consider the following example.

Example 1.3

Emergent Behavior

As humans, we have the capability to walk, run, etc. However, there is a need to travel more efficiently in a shorter period of time. To achieve this higher-level purpose, humans created bicycles expressly for enabling a human to travel great distances more efficiently. But *how would you know that (1) a set of physical components could be assembled into a Bicycle System as a prime mover capable of rolling and steering (emergent behaviors) and (2) a human could simultaneously C2—balance, pedal, and steer (emergent behaviors)—the Bicycle System*? If we analyzed the human or the bicycle, do they exhibit or reveal the capabilities—emergent behaviors—that enable them as an integrated system to accomplish the higher-level mission (travel more efficiently in a shorter period of time)? Similar emergent behavior examples include jet engines or aircraft that can counter the effects of gravity and fly.

Chapter 3 provides additional discussion on *emergent* behavior.

1.3.4 System Definition: "… For Command & Control (C2) by Users to Achieve Performance-Based Mission Outcomes …"

Observe that the thrust of this phrase is an expectation to accomplish something—an outcome with a level of performance. More specifically, accomplish performance-based mission outcomes. Those who work in non-Aerospace and Defense (A&D) sectors often associate the term *mission* as unique to military systems. That is factually incorrect! Enterprises, projects, and individuals—medical doctors, educators, and so on—all perform missions.

A *mission* represents an Enterprise or Engineered System outcome and supporting performance-based objectives to be achieved. Consider the following mission examples.

Example 1.4

- **Medical Mission**—Improve the health conditions of …, find a cure for …, administer intravenous drugs to a patient in accordance with a doctor's orders, and so on.
- **Transportation Mission**—Safely transport passengers via air, train, or bus from one city to another, deliver parcel packages, and so on.
- **Services Mission**—Provide cable and Internet services to customer's businesses or homes, respond to fire and medical emergencies, and so on.
- **Educational Mission**—Offer an accredited (EE, ME, SwE, ChemE, IE, etc.) Engineering degree program.

The concept of missions, however, is not limited to Enterprise Systems. Interestingly, Enterprises for decades have developed *vision* and *mission statements*. Yet, often fail to recognize that the Engineered Systems they produce for the marketplace are designed to perform *missions* to support their customers'—Users and End Users—Enterprise System missions. When a system, product, or service ceases to perform a *mission*, it has no value to its Users in terms of outcomes to be accomplished—End User satisfaction and shareholder value and revenue generation—and will likely be retired or disposed.

Chapters 4 and 5 address *missions* and *mission analysis* in more detail.

1.3.5 System Definition: "… In a Prescribed Operating Environment …"

Humans and equipment often have performance limitations in terms of what types of operating environments they can operate to accomplish a mission. This requires knowledge and understanding of (1) *where* missions will be conducted—land, sea, air, space, or combinations of these—and (2) under *what* types of conditions. Once the external operating environment is understood, it must be *specified* and *bounded* in terms of performance requirements such as temperature, humidity, shock and vibration, and salt/fog.

1.3.6 System Definition: "… With a Probability of Success"

Finally, to support a User's missions, the system must be available *on demand* to reliably conduct missions and deliver performance-based outcomes with a *probability of success*. If a system, product, or service is unable to fulfill the minimum requirements for mission success prior to its mission, then mission failure may be the consequence and other alternative systems must be considered.

1.3.7 Other Definitions of a System

As a final note, national and international standards and professional organizations as well as different authors present various definitions of a *system*. If you analyze these, you will find a diversity of viewpoints, all influenced and tempered by their personal knowledge and experiences. Moreover, achievement of a "one size fits all" *convergence* and *consensus* by standards organizations often results in weak wording that many believe it to be *insufficient* and *inadequate*. For additional definitions of a system, refer to the following standards:

- INCOSE (2015). *Systems Engineering Handbook: A Guide for System Life Cycle Process and Activities* (4th ed.).
- IEEE Std 1220$^{\text{TM}}$-2005 (2005)—Institute of Electrical and Electronic Engineers (IEEE)
- ISO/IEC 15288:2015 (2015)—International Organization of Standards (ISO)
- DAU (2011)—Defense Acquisition University (DAU)
- NASA SP 2007-6105 (2007)—US National Aeronautics and Space Administration (NASA)
- FAA SEM (2006)—US Federal Aviation Administration (FAA)

You are encouraged to broaden your knowledge and explore definitions by these organizations. Depending on your personal viewpoints and needs, the definition stated in this text should provide a more definitive characterization.

1.4 LEARNING TO RECOGNIZE TYPES OF SYSTEMS

Systems occur in a number of forms. High-level examples include:

Example 1.5

System Examples

- Economic systems
- Communications systems
- Educational systems
- Entertainment systems
- Financial systems
- Government systems
- Environmental systems
- Legislative systems
- Medical systems
- Judicial systems
- Corporate systems
- Revenue systems
- Insurance systems
- Taxation systems
- Religious systems
- Licensing systems
- Social systems
- Military systems
- Psychological systems
- Welfare systems
- Cultural systems
- Public safety systems
- Food distribution systems
- Parks and recreation systems
- Transportation systems
- Environmental systems

Observe that many of the example systems are subsets of others and may be interconnected at various levels to form Systems of Systems (SoS). If we analyze these systems or SoS, we find that they produce combinations of performance-based outcomes such as products, behaviors, by-products, or services. As systems, they exemplify the definition of a system introduced earlier.

1.4.1 Precedented Versus Unprecedented Systems

Enterprise and Engineered Systems, in general, are either *precedented* or *unprecedented*:

- **Precedented Systems**—Systems for which earlier versions exist and provide the basis for upgrades such as technology and performance improvements
- **Unprecedented Systems**—Systems that represent innovations and radical new designs that depart from traditional designs, for example, the introduction of hybrid vehicles

To illustrate these terms, consider the following automobile example.

Example 1.6

Automobile Application: Precedented and Unprecedented Systems

Gasoline-powered automobiles are an example of *precedented* systems. Over many decades, they consisted of a frame, body, doors, engine, inflatable ties, steering, and so on.

Then, as newer automotive technologies evolved over the past 100+ years, manufacturers added new features and capabilities that were *unprecedented*. Examples included heaters, air-conditioning, power steering, electronic ignition, electrical doors and windows, compression bumpers, air bags, entertainment systems, satellite radio and phone data communications, and hybrid engines.

1.4.2 Products as Systems

Our discussions to this point have focused on the generic term *system*. *Where do consumer products and services fit into the context of a system*? A *product* consisting of two or more entities integrated together to provide a performance-based capability, by definition, is an instance of a system. Observe that a product provides a "capability" but does not address outcome. Why? Unless preprogrammed to run autonomously, products as inanimate objects are dependent on humans to apply them to a specific situation and subsequently achieve an outcome. For example:

- A pencil is a *product*—an instance of a system—comprised of a lead, a wooden or composite holder, an attached eraser that provides a capability but no outcome on its own.
- A computer monitor is a *product*—an instance of a system—comprised of an chassis, touch screen display, motherboard, processor, sound board, and interface ports—power, video, audio, and communication ports such as USB:
  - The computer processor transmits commands and data to the monitor to display formatted information to its User.
  - In response to the display data, the User has the option to provide a stimulus via the touch screen display to select an action to be performed—command

and audio volume—that results in an outcome as verification feedback of acceptance and subsequent completion of the action.

1.4.3 Tool Context

Some systems or products are employed as tools by HIGHER ORDER SYSTEMS such as an Enterprise. Let's define what we mean by a tool:

- **Tool**—A physical product or software application employed by a User to enable them to leverage their own capabilities to more *efficiently* and *effectively* perform a task and accomplish a performance-based outcome that exceeds their own strengths—capabilities—and limitations.

Consider the following example:

Example 1.7

Software Application as a Tool

A statistical software application, as a support tool, enables a statistician to efficiently sort and analyze large amounts of data and variances in a short period of time.

Now, is a wooden log, as an entity, a system? No, however, the log is considered a tool that has the capability to deliver a performance-based outcome when applied by a human operator under specific conditions.

1.4.4 Service Systems

The preceding discussion illustrates that the outcomes produced by a system may be (1) physical such as products and by-products or (2) behavioral responses—services. *What is a service*?

- *A service* is an activity provided and performed by an Organizational or Engineered System to produce an outcome that benefits its User.

Consider the following example.

Example 1.8

Consumer Product Services

- Weight scales are a consumer *product*—an instance of a system with multiple parts integrated together as a system—that respond to a User stimulus to provide weight measurement information in pounds or kilograms as a *service* response. Observe that the service delivers an outcome–displayed weight, however, no physical products are-produced.
- A digital alarm clock as a consumer product provides a service by displaying current time and an alarm when activated and set for a specific time.

Now that we have established what a system is brings us to the next question: *what is SE*?

1.5 WHAT IS SE?

Definition of SE requires an understanding of its two constituent terms: *system* and *Engineering*. Since the preceding discussions defined a *system*, the next step is to define *Engineering* to enable us to define SE.

1.5.1 Definition of Engineering

Engineering students often graduate without being introduced to the root term that provides the basis for their formal education. To illustrate this point, consider a conversational example.

Example 1.9

The Engineer's Dilemma

- *What is your profession*?
- I'm an Engineer—SE, ME, EE, SwE, ChemE, Test, and so on.
- *What do Engineers do?*
- We Engineer things.
- *So, what is Engineering?*
- (Silence) I don't know. Our instructors and courses didn't cover that topic!
- *So, even though you have received an Engineering degree, you are unaware of how "Engineering" is defined by your profession?*

The term *engineering* originates from the Latin word *ingenerare*, which means "to create" (Britannica, 2014). Its first known use is traceable to 1720 (Merriam-Webster, 2014). Let's introduce a couple of example definitions of Engineering:

- **Engineering**—"The profession in which knowledge of the mathematical and natural sciences gained by study, experience, and practice is applied with judgment to develop ways to utilize economically the materials and forces of nature for the benefit of mankind" (Prados, 2007, p. 108).
- **Engineering**—"The application of science and mathematics by which the properties of matter and the sources of energy in nature are made useful to people" (Merriam-Webster, 2014).

The Prados (2007) definition of Engineering above originates from earlier definitions by the Accreditation Board for Engineering and Technology (ABET), which accredits Engineering programs in the United States. ABET evolved the definition from its founding in 1932 until 1964. It continued to appear in ABET publications from 1964 through 2002 (Cryer, 2014).

Two key points emerge from the introduction of these definitions:

- First, you need to understand the definition and scope of your profession.
- Secondly, on inspection, these definitions might appear to be a mundane, academic discussion. The reality is that these definitions characterize the traditional view of Engineering. That is, Engineering the "Box (Equipment Hardware & Software)" Paradigm or "Box" Engineering that contributes to systems, products, or services failures attributed to "human error" (Chapter 24) or are considered by the User to be failures due to a lack of *usability*. This is a critical staging point in differentiating the scope of the SE – "Engineering the (User-Equipment) System, which includes the (Equipment) Box," versus traditional "Box" Engineering. In that context, SE exemplifies the cliché "Learning to think outside the (Engineering) box" to develop systems, products, and services that Users actually need, can use, and lead to a reduction in human errors that contribute to system failures. As a result, this impacts Enterprise System reputation, profitability, customer satisfaction, marketplace perceptions, and subsequently shareholder value.

Now that we have established definitions for a *system* and *Engineering*, let's proceed with defining SE.

1.5.2 Definition of System Engineering (SE)

Principle 1.2

Content–Grammar Principle

Substantive content must always precede *grammar* to achieve successful results. Avoid negotiating content for the sake of achieving grammatical elegance and eloquence unless it precludes misinterpretation.

There are a number of ways to define SE, each dependent on an individual's, project's, or Enterprise's business domain, perspectives, and experiences. SE means different things to different people. You will discover that even your own views of SE will evolve over time. So, if you have a diversity of perspectives and definitions, *what should you do*? What is important is that you, project teams, or your enterprise should:

- Establish a consensus definition for SE.
- Document or reference the SE definition in enterprise command media to serve as a guide for all.

For those who prefer a brief, high-level definition that encompasses the key holistic aspects of SE – "Engineering the System" - consider the following definition:

- **System Engineering**—The multi-disciplined application of analytical, mathematical, and scientific principles for formulating, selecting, developing, and maturing an *optimal* solution from a set of *viable* candidates that has *acceptable* risk, *satisfies* User operational need(s), and *minimizes* development and life cycle costs while *balancing* Stakeholder interests

To better understand the key elements of the SE definition, let's address each of the phrases separately.

1.5.2.1 SE Definition: "The Multi-disciplined Application of …" System, product, and service development typically require multiple Engineering disciplines of expertise to translate a User's *operational need* and *vision* into a deliverable system, product, or service that produces performance-based outcomes required by the User. Accomplishment of that translation process requires *multi-disciplined* integration of hardware, software, test, materials, human factors, reliability, maintainability, and logistics Engineering.

1.5.2.2 SE Definition: "… Analytical, Mathematical, and Scientific Principles …"

Author's Note 1.1

Constructive Assessment

The following discussion is intended to be a constructive assessment concerning the state of traditional Engineering and its views of SE today versus what twenty-first-century Engineering and SE demand. The time has come to shift the educational paradigm!

Although not explicitly stated, the ABET (Prados, 2007, p. 108) definition of Engineering infers that the work scope of Engineering focuses on the innovation and development of devices, mechanisms, and structures to produce one or more performance-based outcomes for the benefit of mankind. In fact, we analytically represent the boundaries a system, product, or service as a "box" such as Figures 3.1 and 3.2. Psychologically, the simple act of establishing these boundaries automatically fosters an "Engineering within the walls of the box and connections between the boxes" paradigm. As a result, discipline-based Engineering courses and instruction focus on:

- Developing systems, devices, and mechanisms that utilize materials—technology—to harness, adapt, transfer, store, convert, transform, and condition the “forces of nature” such as energy, forces, information, and operating environment conditions to produce performance-based outcome(s) that benefits mankind

Unfortunately, this physics-based *scientific* and *mathematical* paradigm fosters *misperceptions* that SE is limited to Engineering and designing:

- Mechanical structures, enclosures, and mechanisms that can withstand, survive, divert, convert, transfer, transform, and store the physical “forces of nature”
- Electrical devices, components, and mechanisms that (1) respond to electrical, electronic, optical, and acoustical stimuli, excitations, and cues to produce specific outputs and characteristics; (2) store and retrieve energy and information; (3) select and locate components on printed circuit layouts; (4) perform self-tests and diagnostics; and (5) interconnect wiring and cables to compatible and interoperable components
- Software to (1) perform algorithmic decisions and computations to C2 systems and products and provide Situational Assessments and (2) mathematically model and simulate component and physics characteristics and other phenomena

The scope of SE does in fact encompass these multi-discipline Engineering activities as illustrated on the right side of Figure 1.1. In general, Engineers graduating from accredited institutions are well educated and competent in performing these activities. However, SE encompasses more than simply this traditional Engineering view of SE as illustrated by the left side of Figure 1.1 concerning Analytical Problem-Solving and Solution Development. This requires more than simply “plugging and chugging” equations to harness the “forces of nature.” As a result, most Engineers are unprepared to enter industry and government to perform these activities. We will address this point in more detail in Chapter 2.

Finally, the term *mankind* at the end of the ABET (Prados, 2007, p. 108) definition … as a beneficiary of Engineering work. The question is: *who determines what would “benefit” mankind that motivates the need to involve and initiate Engineering actions*? In general, we can say *mankind* represents the marketplace. But, *who determines what the needs are for the marketplace, in general, or one of its segments*? The answer has two contexts: consumer product development and contract-based system development detailed in Chapter 5 (Figure 5.1) or services support:

- **Consumer Product Development**—Commercial industry expecting to make a profit as a Return on Investment (ROI) develops systems, products, and services for marketplace consumers. As a result, they have to *understand* and *anticipate* what potential consumers of a system, product, or service Want, Need, Can Afford, and are Willing to Pay (Principle 21.1).
- **Contract-Based System Development and Support Services**—Industry and government analyze their own needs and either develop systems and services internally or acquire/outsource them from external contractors or vendors.

Principle 1.3

Intellectual Control Principle

One of the key roles of an SE is to maintain “intellectual control of the problem solution” (McCumber and Sloan, 2002).

Returning to the question, *who determines what the marketplace needs are*? In either of the consumer or contract development cases above, the answer is someone with a technical background preferably in Engineering who has the *interpersonal* skillset to collaborate with the Users—consumer or contract—to:

1. Understand, analyze, identify, and document their operational needs and expected performance-based outcomes.
2. Specify and bound the Problem, Opportunity, or Issue Space (Figure 4.3) that needs to be solved or mitigated.
3. Specify and bound the Solution Space(s) (Figure 4.6 and 4.7) that represents what the User Wants, Needs, Can Afford, and Is Willing to Pay (Principle 21.1) to acquire or develop.
4. Collaborate with multiple Engineering disciplines to translate the Solution Space(s) performance-based outcomes and characteristics into an architectural-driven set of multi-level specification requirements.
5. Select an overall system, product, or service that is *optimal* across all User Solution Space scenarios and conditions.
6. Plan, implement, and orchestrate the technical strategy for a project as a Project or Lead Systems Engineer (LSE) or as a development team SE.
7. Maintain intellectual control (McCumber and Sloan, 2002) of the *evolving* and *maturing* System Design Solution to ensure that it is *consistent* and *traceable* to User Solution Space(s) *source* or *originating* requirements.

These points illustrate *why* the traditional Engineering view of SE as illustrated by the right side of Figure 1.1 is short scoped. SE encompasses more than simply the design of physical systems, devices, and components.

Based on the preceding discussion, the scope of SE encompasses three areas of concentration (Figure 1.1):

- **Analytical Problem-Solving and Solution Development—**
  Example activities include *collaboration* with *external* and *internal* Users to identify, specify, and bound their operational needs and capabilities; oversight of multi-level design development and integration; assessment of System Integration and Test results for compliance to specification requirements; and conduct and review of Analysis of Alternatives (AoA).
- **Multi-discipline Engineering**—Example activities include *collaboration* with Engineers concerning the development and interpretation of requirements, design integrity, analyses and trade-offs, prototype development, and Modeling and Simulation (M&S).
- **Technical PM**—Example activities include planning, tailoring, orchestrating, and implementing the technical project including baseline and risk management, conducting technical reviews, Specialty Engineering Integration, performing Verification and Validation (V&V) oversight, and preserving the technical integrity of the project.

As a project development "system," these activities are not just discrete activities. They must be integrated at two levels: (1) the Enterprise System developing the (2) Engineered System. Remember that Figure 1.1 illustrates a project's Enterprise System performing multi-discipline SE. As with any type of system, its interfaces must be *compatible* and *interoperable* to orchestrate the interactions and bi-directional communications across each interface to achieve success.

Therefore, SE not only requires the *application of mathematical and scientific principles* addressed in the Prados (2007) Engineering definition but also encompasses *analytical principles*—both *inside* or *outside* the system, product, or service and within the Enterprise System among its system developers.

1.5.2.3 SE Definition: "… For Formulating, Selecting, Developing, and Maturing an Optimal Solution from a Set of Viable Candidates …" Engineers and teams often exhibit a propensity to take a *quantum leap* from requirements (Figure 2.3) to a single Point Design Solution without due consideration of:

1. How the User expects to deploy, operate, maintain, sustain, retire, and dispose of a system or product.
2. An evaluation of a viable set of candidate solutions and selection.
3. User life cycle costs and risks.

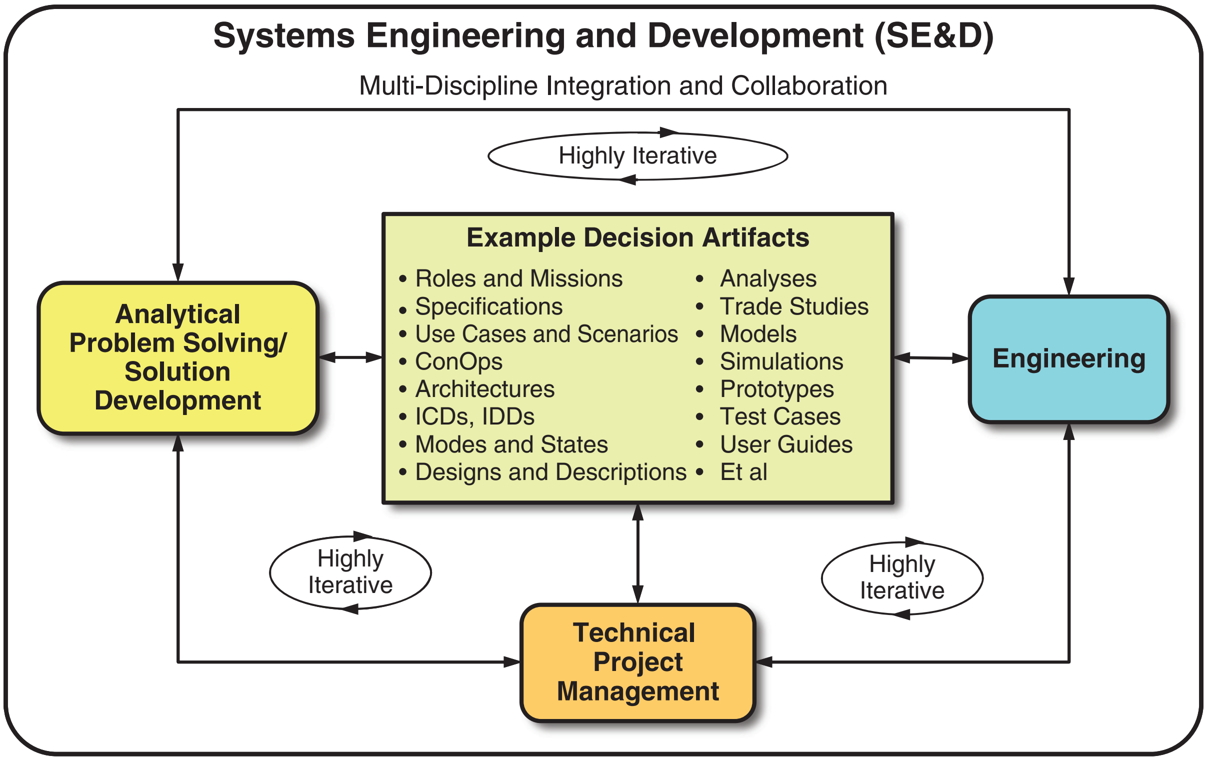

Figure 1.1 The Scope of SE and Its Relationship to Traditional Engineering

Therefore, a key objective of SE is to ensure that each design is formulated, evaluated, and selected from a set of *viable alternatives* using Multivariate Analysis or AoA. The selection may not be *ideal*; however, for a given set of constraints and operating environment conditions, it may be the best—*optimal*—that can be achieved.

1.5.2.4 SE Definition: "… That Has Acceptable Risk …" If you ask Engineers what *level of risk* a system, product, or service should have, without hesitation, a common answer is *low risk*. The reality is customer budgets, schedules, technical requirements, and technologies may impose constraints in some cases that result in low-, medium-, or high-risk situations—whatever is *acceptable* to the User. This assumes that the System Developer has collaborated with the User to enable them to make an informed risk decision concerning the options and consequences. Therefore, under certain circumstances, low, medium, or high risk may be *acceptable* to the User. Ideally, SE tries to *mitigate* and *reduce* the risk via methods such as rapid prototyping, proof of concept and proof of technology demonstrations, and M&S methods.

1.5.2.5 SE Definition: "… Satisfies User Operational Need(s) …" If you ask Engineers and Analysts where their requirements originate, the response is "from our contracts organization." From an Enterprise protocol perspective, that is true. However, *how do you know that the User requirements passed along by your contracts organization accurately and completely characterize the User's operational need*? Suppose you develop a system, product, or service that *complies* with those requirements and the User determines after delivery that it did not meet their operational needs. Who is to blame—legally and professionally? This brings us to a key principle of SE:

When a consumer or User—System Acquirer—purchases a system, product, or service, there is an expectation that it will achieve performance-based outcomes that characterize their operational needs. Those criteria typically characterize what is required to enable them to perform their missions. Therefore, SE technically *begins* and *ends* with the User and their End Users. Within this timeframe, the focal point of SE decision making centers on the User as a User's advocate. principle 1.1 illustrates the symbolism.

1.5.2.6 SE Definition: "… Minimizes Development and Life Cycle Costs …" Many years ago, Enterprises and Engineers often had the view that their objective was to develop a system or product within a project's contract or task order triple constraints—cost, schedule, and technical. That was and is true, especially on Firm-Fixed Price (FFP) contracts. However, review the response again. It offers no indication of concern for the customer or User and their costs to deploy, operate, maintain, and sustain a system or product *after* it is delivered. In fact, the attitude was "we get paid to develop the system. Operations, Maintenance, and Sustainment (OM&S) costs are the User's problem." Those firms are either out of business today, consumed by their competition, or forced to change to survive.

Today, with demands on budgets—do more for less cost—Users are challenged to deal with the realities of System OM&S costs and the Total Cost of Ownership (TCO) of a system as an asset. Therefore, a key objective of SE during System Development is to *minimize* both development and User life cycle costs.

1.5.2.7 SE Definition: "… While Balancing Stakeholder Interests" Enterprises and Engineered Systems have a variety of stakeholders to satisfy beginning with the acquisition of a system or product and continuing through its disposal. The same is true for System Developer—shareholders and suppliers. Therefore, another objective of SE is to achieve a *balance* not only in their own Enterprise interests but also to be a User's advocate. How is this achieved? Figure 1.2 illustrates how multi-discipline SE "bridges the gap" between a User's operational needs and the Engineering disciplines required to engineer a system, product, or service.

Additionally, as we shall see in Chapter 3, *stakeholders* include competitors and adversaries that have vested interests in the *success* or *failure* of a system, product, or service.

1.6 *SYSTEM* VERSUS *SYSTEMS* ENGINEERING

People sometimes get into debates about references to *System* versus *SE*. Which is correct? The answer depends on the *context* of the usage from a *project* SE or *enterprise* SE perspective.

1.6.1 Project SE

For example, a customer—System Acquirer or User—issues a contract to develop "a system." From a PM perspective, a project has (1) a Project Manager (PM) and (2) a work scope that is time-bounded with a beginning and ending for the development and delivery of the system, product, or service. The project's organizational element accountable for SE is labeled Project XYZ System Engineering (singular).

For large, complex systems that require development of multiple systems, typically project is assigned to a PM. The collective set of projects are organized underneath a Program managed by a Program Manager. At that level, the program's organizational label for SE is Program ABC Systems Engineering (plural).

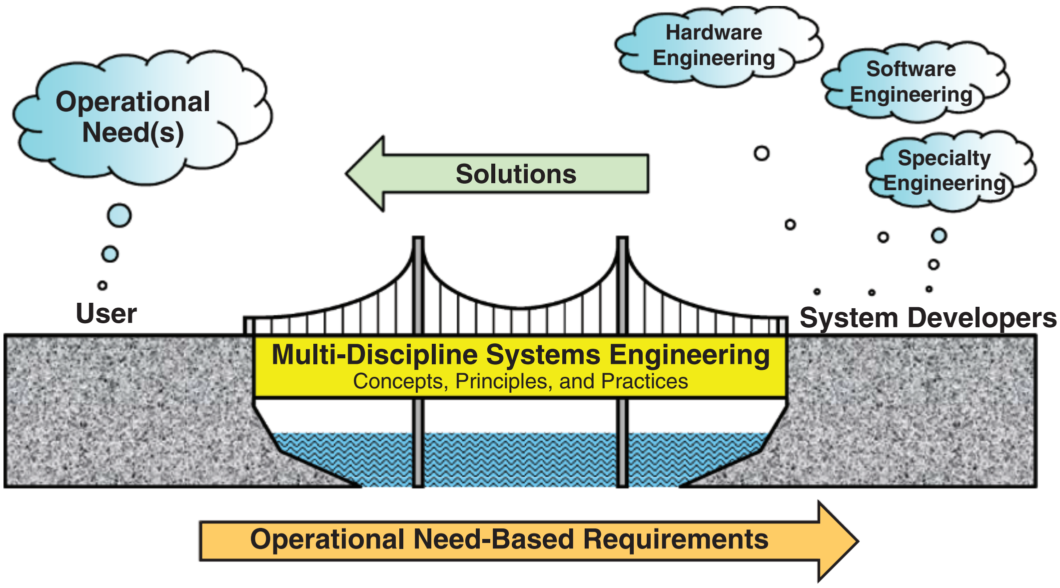

Figure 1.2 Multi-discipline SE "Bridges the Gap" between Users and System Developer Engineering Disciplines

1.6.2 Functional SE

Within most Enterprises, functional departments are established to supply personnel with Engineering discipline specialties such as EE, ME, and SwE to projects. Enterprise organizational charts often include a Systems Engineering Department. Observe the plural form *systems*. Matrix-based organizations such as departments supply Engineers with defined skillsets to perform specific tasks across multiple projects. Therefore, the term Systems Engineering (plural) is often applied to functional organizations.

1.7 SE: HISTORICAL NOTES

The earliest form of system applications began with early man with the innovation of the wooden, stone, and bone tools such as the lever and fulcrum, spear, and wheel. Systems evolved and became increasingly complex such as ground vehicle and ship transportation systems, weapons, and fortresses. The need to "engineer systems" evolved as a response to the demand to counter threats, move large objects, and develop products. Subsequently, the need to mass produce items in the late 1700s lead to the Industrial Revolution. In recent decades, larger complex systems and products drove the need to predictably and reliably develop and produce systems.

Most textbooks attempt to summarize the history of SE with facts and dates. You are encouraged to read those accounts. However, understanding what SE is today requires more than reading and memorizing facts and dates of past history. More importantly, you need to understand *how* and *why* modern-day SE has evolved as a discipline. Key points to note are:

1. During the first half of the 1900s, a new field of Systems Management emerged. Failures were attributed to poor Systems Management. As a result, rigid, inflexible management controls and processes were implemented (Johnson, 2002, pp. 1, 227–231).
2. Failures, however, continued to occur due to increasing complexity of WWII era and beyond military systems. Subsequently, industry and government came to the realization that the failures were due to poor reliability, not just Systems Management. As a result, the focus shifted to development and evolution of Reliability, Maintainability, and Availability (RMA) subsequently improving system performance.
3. During this timeframe, increasing system complexity began to drive the need to formulate a systems methodology. In turn, this led to the emergence of SE processes and methods to meet industry and government needs.

You are encouraged to read Johnson (2002) to better understand these points. Chapter exercises will delve into comparisons of recent SE processes and methods since WWII.

1.8 SYSTEMS THINKING AND SE

SE is often equated to Systems Thinking and Systems Engineers as Systems Thinkers. *What is Systems Thinking*? From the author's perspective, Systems Thinking is the ability to:

- Visualize or conceptualize any type of system—Natural, Engineered, or Enterprise—and all of its constituent levels and components, their interrelationships, and operational interactions with its OPERATING ENVIRONMENT.
- Perform a situational assessment of a system condition and level of urgency to initiate the appropriate, corrective actions in a timely manner.
- Formulate, develop, and synthesize a set of solutions that respond to User operational needs and constraints.
- Perform an AoA to evaluate and select the *optimal* solution that has *acceptable* risk to satisfy the User's operational needs and constraints for the least total life cycle cost.
- Optimize the selected solution to provide the best value—cost-performance–benefit ratio—to the User based on their operational needs, priorities, and acceptable risk.
- Observe system performance or the lack thereof, assimilate the observable facts, model, and analyze the contributory causes and effects.

Observe several key operative terms above that characterize Systems Thinking. These are visualize, conceptualize, assess, formulate, develop, synthesize, evaluate and select, optimize, assimilate, model, and analyze. To illustrate Systems Thinking under pressure, consider Mini-Case Studies 1.1 and 1.2.

Mini-Case Study 1.1

Systems Thinking in Action: The Apollo 13 Accident (Figure 1.3)

On April 11, 1970, NASA launched Apollo 13 on a lunar mission. The mission configuration consisted of a Command Module (CM) and a Service Module (SM) containing the Lunar lander. While in lunar orbit, two astronauts would enter the Lunar Module (LM), separate from the CM, land on the Moon's surface, return to the CM containing the third astronaut circling the Moon, and jettison the LM en route back to Earth.

Two days into the mission, an oxygen tank onboard exploded crippling the SM causing the lunar landing to be aborted. Challenged with being unable to visually assess the damage, the astronaut crew and Engineers on the ground had to make Situational Assessments concerning how to manage (1) redirecting a spacecraft back to Earth that was traveling away from Earth toward the Moon; (2) limited onboard power, water, heat, and breathable air resources; and (3) return of the crew home safely.

The demand for Systems Thinking became paramount. Given the situation, how do you assimilate onboard resources in the integrated CM–SM and attached LM to *synthesize* multiple power, water, heat, air, and other solutions to ensure survival of the astronaut crew? "The LM was designed to support two men on a 2-day expedition to the lunar surface. Mission Control made major revisions in the use rate of water, oxygen, and electrical power to sustain three men for the 4-day return trip to the Earth." (NASA 1970, p. 5–33) Additionally, to conserve power, the astronauts had moved to the LM as a "lifeboat" for return to Earth allowing the CM–SM to be powered down. However, to ensure a safe return, the CM–SM would have to be powered up, an action that was not intended to be performed in-flight, for the astronauts to reenter it prior to entry.

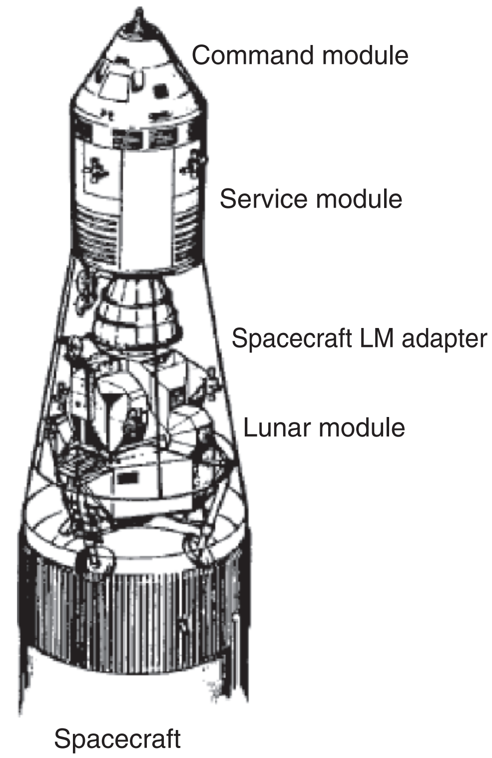

Figure 1.3 Apollo Vehicle (Source: NASA (1970))

Refer to NASA (1970) and Wikipedia (2014) for details of the solutions that illustrate how NASA applied Systems Thinking to *innovate* and *create* real-time solutions that enabled the astronauts to survive and return home safely.

Mini-Case Study 1.2

Systems Thinking in Action: Critical Software System Test

A project was developing a large software intensive system to replace an existing system. To accomplish an orderly transition and replacement, the new system operated in a surrogate "shadow mode" with the existing system to Verify and Validate (V&V) its condition as being *operationally ready*. The initial test was scheduled to occur in the early morning hours when demand for the primary system was low. Due to the system's criticality as a control center, the test was under heavy scrutiny politically, technically, and technologically.

After months of challenging work to meet an unrealistic development schedule, the software "locked up" during Pre-Test Checkout activities. Despite the efforts of many people and heavy pressure from executives and customers demanding corrective action due to careers being on the line, the contributory root cause for the lock-up could not be identified.

Frustrated, the Lead Software Systems Engineer left the control center and walked around the large parking lot several times trying to visualize and assimilate observable facts based on a mental model of the software's architecture and conceptualize a corrective action solution. During one of the laps, a "light bulb" came on in their head when they realized that a key software flag may not have been documented and set in the Pre-Test procedures. The Engineer returned to the control center, set the flag, and the software became fully operational less than 30 minutes before the crucial test.

Is Systems Thinking Unique to SEs and Engineers? Absolutely not! Systems Thinking is a personal attribute unrelated to SE or engineering. Engineers, by virtue of reputation of their interest in "tinkering and understanding how things work," are often characterized by family members and teachers as "Systems Thinkers." The same can be said about auto mechanics, food preparation in the home, PMs, and many other skills. However, there is a difference between being mechanically, electrically, electronically, or software minded—one form of Systems Thinking—and the ability to see things on a much larger, conceptual scale such as Einstein's creation of the Theory of Relativity. Systems Thinkers are present in every field—biology, chemistry, physics, medicine, politics, education, architecture, banking, military, communications, and automotive repair, not just Engineering!

1.9 CHAPTER SUMMARY

This concludes our discussion Systems, Engineering, and SE. Key points include:

1. We defined *system* is in terms of what it is, why it exists, what it is expected to accomplish, and who it benefits.
2. To define SE, we introduced the ABET (Prados, 2007, p. 108) definition for *Engineering* and coupled with the *system* definition.
3. We highlighted the scope of SE as "Engineering the (User-Equipment) System" that encompasses the "Engineering of the (Equipment) Box" by traditional Engineering that often contributes to factors that drive system failures and poor customer satisfaction.
4. We also explored examples of types of systems; distinguished between *precedented* and *unprecedented* systems; and considered the context of systems, products, and tools.
5. Since people often use the terms *SE* and *SE* interchangeably, we delineated the usage based on its project or Enterprise context.
6. Lastly, we explored one of the key attributes of SEs, Systems Thinking.

Given this introductory background, Chapter 2 will address THE EVOLVING STATE OF SE PRACTICE: CHALLENGES AND OPPORTUNITIES.

1.10 CHAPTER EXERCISES

1.10.1 Level 1: Chapter Knowledge Exercises

1. Create your own definition of a system. Based on the "system" definitions provided in this chapter:
   a. Identify your viewpoint of shortcomings in the definitions.
   b. Provide rationale as to why you believe your definition overcomes those shortcomings.

2. From a historical perspective, identify three *precedented* systems that were replaced with *unprecedented* systems.

3. What is a *system*?

4. Is a *product* a system?

5. Is a *service* a system?

6. What are examples of different types of systems?

7. What are the two primary types of systems associated with system, product, or service development?

8. What is an *Engineered System*?

9. What is an *Organizational System*?

10. What is *Engineering*?

11. What is SE?

12. SE consists of three primary aspects. What are they? Describe the interactions among the three.

13. How does the scope of *Engineering* compare with *SE* in terms of "Engineering the System" versus "Engineering the Box."

14. What is the difference between a system, a product, and a tool?

1.10.2 Level 2: Chapter Knowledge Exercises

Refer to www.wiley.com/go/systemengineeringanalysis2e

1.11 REFERENCES

Britannica (2014), *Encyclopedia Britannica*, Chicago, IL: Encyclopedia Britannica, Inc. Retrieved on 1/14/14 from http://www.britannica.com/EBchecked/topic/187549/engineering.

Cryer, Keryl (2014), *Correspondence – ABET Definition of Engineering*, Baltimore, MD: Accreditation Board for Engineering and Technology (ABET).

DAU (2011), *Glossary: Defense Acquisition Acronyms and Terms*, 14th ed. Ft. Belvoir, VA: Defense Acquisition University (DAU) Press. Retrieved on 3/27/13 from: http://www.dau.mil/pubscats/PubsCats/Glossary%2014th%20edition%20July%202011.pdf.

IEEE Std 1220™-2005 (2005), *IEEE Standard for the Application and Management of the Systems Engineering Process*, New York: Institute of Electrical and Electronic Engineers (IEEE).

INCOSE (2015). *Systems Engineering Handbook: A Guide for System Life Cycle Process and Activities* (4th ed.). D. D. Walden, G. J. Roedler, K. J. Forsberg, R. D. Hamelin, and, T. M. Shortell (Eds.). San Diego, CA: International Council on Systems Engineering.

ISO/IEC 15288:2015 (2015), *System Engineering - System Life Cycle Processes*, Geneva: International Organization for Standardization (ISO).

Johnson, Stephen B. (2002), *The Secret of Apollo: Systems Management in the American and European Space Programs*, Baltimore, MD: The Johns Hopkins University Press.

McCumber, William H. and Sloan, Crystal (2002), *Educating Systems Engineers: Encouraging Divergent Thinking*, Rockwood, TN: Eagle Ridge Technologies, Inc. Retrieved 8/31/13 from http://www.ertin.com/papers/mccumber_sloan_2002.pdf.

Merriam-Webster (2014), Merriam-Webster On-Line Dictionary, www.Merriam-Webster.com

NASA (1970), *Report of Apollo 13 Review Board*, Washington, DC: NASA, Accessed on 5/19/14 from http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19700076776.pdf.

Prados John W. (2007), *75th Anniversary Retrospective Book: A Proud Legacy of Quality Assurance in the Preparation of Technical Professionals*, Baltimore, MD: Accreditation Board for Engineering and Technology (ABET).

Wikipedia (2014), *Apollo 13 web page*, San Francisco, CA: Wikimedia Foundation, Inc.