PART I

SYSTEMS ENGINEERING – A GENERAL OVERVIEW

1.1

THE ORIGINS, HISTORY, AND UNIQUENESS OF SYSTEMS ENGINEERING

For many decades, each of the industries that relied heavily on engineering, such as electronics, mechanics, and chemistry, had its own unique discipline. The engineers of each discipline evolved and gained experience in their respective specializations. *But, in the early 1970s, the need arose to integrate the various engineering fields and even bridge the gap between engineering, as a whole, and nonengineering systems*.

This phenomenon has its source in two opposing trends: on the one hand, engineering disciplines were becoming more and more specialized; and on the other hand, the need for multidisciplinary skills was on the rise.

Clarification: technological developments led to an increase in specialization and created a need for more and more specialists in subdisciplines of engineering. Today, for instance, an electronics engineer would not be considered an expert in electronics, but rather in one of its more specific subdisciplines, such as communications or control systems. Therefore, in order to create an electronic system, one now needs to integrate these subdisciplines. On the other side of the spectrum, the technological capability of manufacturing complex products for the end user's benefit raised the importance of integration between the overarching engineering disciplines, such as mechanics, electronics, or materials engineering, as well. At the same time, the systemic complexity of developed systems began to increase too.

This need gained much momentum from the development of the software and software engineering fields. Software allows for the creation of complex systems and is,

Managing and Engineering Complex Technological Systems, First Edition.
Avigdor Zonnenshain and Shuki Stauber.

in many cases, the central factor that facilitates the combination of subsystems from various disciplines.

In the words of Eric Honour, who believes that the evolution of systems engineering in the 1980s received a substantial booster shot from the breakthrough in the software field that took place during those years: "In the late 1980s, the industry was faced with a major problem: many software failures were discovered, because software personnel had not received the information they needed, at the quality they required. They began looking at ways of receiving better requirement specifications, and that put systems engineering back into people's minds."

We can assume that one of the main reasons for the emergence of systems engineering is the development of technological abilities that, with the help of software, allowed for the creation of technological systems of ever-increasing complexity. This phenomenon created the need for a technological position holder, charged with the task of integrating the subsystems that form the complex, overarching system.

Technological systems are an integral part of the modern world. They provide us with a variety of services, and play a part in larger and larger systems, some of which contain nontechnological components as well. *This process leads to the creation of supersystems, which can no longer be effectively controlled*, because they are entwined with human systems, among other reasons.

These needs and constraints greatly increased the need for the formulation of an orderly, methodological, systematic approach to the management of complex engineering systems. *The importance of early planning rose greatly*, as did the need for skills that focus on areas other than "pure" engineering. Hence, the technological industry began to understand the increasing need for engineers who followed a predetermined, orderly, controlled, and supervised methodology: *a methodology that would allow them to design holistically and facilitate educated integration processes, while minimizing the ever-present, ever-increasing risk of failure*. Failures were brought on by loss of control over large, complex systems, exposed to a wide variety of constraints, some of which were organic and process-related, rather than pertaining to engineering.

On top of these, came the *financial component*: as aforesaid, the ever-increasing engineering capabilities opened the possibility of developing more and more sophisticated products. As engineering projects grew in scope and complexity, so did their financial costs. Uncertainty levels rose. Complicated projects failed to meet deadlines, went over-budget and even got canceled (a prominent example of this is Israel's Lavi project, which we discuss later in this book).

These trends brought about significant changes in the way large engineering projects (wherein, as aforesaid, the use of systems engineering is especially important) were approached. For instance, in the past, governments used to allocate nearly unlimited resources to the development of complex defense products, such as fighter planes. Today, the costs of these products are so high, that no government can reasonably afford to invest in this area without strict budget limitations and control. Before the late 1970s, most defense projects were managed using cost-plus pricing strategy, while today, most projects operate on a rigid, given budget and are hardly ever allowed to deviate from it. This led to the emergence of yet another fundamental constraint that forced engineers to consider nonengineering factors, especially when planning and developing complex engineering products. It should, however, be noted that the adoption of work methods that rely on a predetermined budget was no magic cure-all, and today's large, complex projects still fail to meet deadlines and stay within their budgets.

Our choice to bring the development of a fighter aircraft as an example of a complex engineering project was not incidental. Some say that aeronautics may be the most complex technological field of all (due to the need to control a large, complex, man-carrying, airborne vehicle). This is what caused systems engineering to evolve in this technological area, first. Moreover, many aeronautics experts contend that they have always been employing the principles of this approach; only back then, they did not refer to it as "systems engineering." As the need for systems engineering became more urgent and began to seep into other industries, it slowly gained recognition as an independent discipline – one worthy of its own, separate training program and career development paths.

1.1.1 ON THE ESSENCE OF SYSTEMS ENGINEERING

Being a discipline in the making, there is, as yet, no consensus on the character and operational frameworks of systems engineering. The interviews in this book show a myriad of perspectives regarding not only the nature of this profession, but also the question of whether it is indeed just that – a profession.

Yossi Ackerman says that a systems engineer is a vague term that defies definition, and he is glad for it. This is because, according to him, this vagueness creates a flexibility that allows the job to be adjusted to suit the circumstances. Ackerman sees systems engineers as "managers, who are engineers by profession, but are able to see the whole technical, technological picture." The more senior the engineer, the more management-oriented his job becomes.

Many experts find that systems engineering is more than a job, it is also a collection of thought and work patterns. Thus, for instance, Mimi Timnat finds that "to a great extent, *systems engineering is more than just a job*. It is an approach to handling and solving problems, and not only work-related ones. It encourages one to look at a problem from different angles, to ask questions and try to gain a better understanding of the problem, before making decisions and formulating solutions." Dr. Cecilia Haskins goes even farther, believing that *a systems engineer does not have to be an engineer at all*, and the word "engineering" may have been wrongfully applied to this term. According to her, the important component is the "systematism" – the ability to see the whole picture and perform the necessary actions methodically. In her perspective, systems engineering is a combination of discipline, worldview, and profession that suggests ways of solving problems.

Niels Malotaux expands on this issue: "*All engineers must be capable of systems thinking.* There is no point in completing part of a system, if it doesn't work together with the other parts. I do not view myself as a systems engineer, although I meet the definition of one. The principles of systems engineers lie at the heart of all engineering disciplines, and all engineers should be able to find an optimal compromise between opposing requirements. 'Systems engineering' is a label applied to all the things engineers have to do, in order to create a good system."

1.1.2 THE DIFFERENT TYPES OF SYSTEMS ENGINEERING

Even the most enthusiastic supporters of systems engineers do not think there is such a thing as a "pure" systems engineer. They believe *systems engineers are engineers who have gained knowledge and experience in one of the classic engineering disciplines* and, being people who possess certain skills and character traits, were able to grow into systems engineers. This professional growth pattern is commonly illustrated using a T model. This model, named for its shape, which resembles the letter "T," places basic training in a concrete engineering field on the vertical axis, and the lateral, multidisciplinary view of the systems engineer, on the horizontal axis.

This perspective suggests that academic programs that train systems engineers have to be for a master's, rather than a bachelor's, degree, and in order to be accepted, applicants would have to possess an engineering degree in one of the fundamental areas of engineering, as well as some hands-on experience working as an engineer is recommended.

Tendencies toward systems engineering are more common among engineers, whose areas of technological expertise frequently involve the tasks of *examining alternatives and facilitating integration*. Aeronautical engineers, whom we have previously mentioned in this context, fit this pattern well. Electronics engineers also have similar traits.

Systems engineering also has some managerial traits, which we will discuss in detail later in this book (see Chapter E), and just as there are different types of managers, there are also different types of systems engineers. *Two super types* prominently figured in the interviews we had performed, their identity derived from the fundamental essence of systems engineering, *as a method that allows its practitioners to analyze a system to its smallest details, and then design it to suit the client's needs*.

This process is commonly illustrated using a "V" shaped graph, where the horizontal axis is time, and the vertical axis is the level of detail. The first step of developing a system is to define the needs of the client. This is represented by the highest point on the V's left branch. Next, begins the process of generally designing the system as a whole, and delving into the details of its subsystems, which should provide concrete answers to the previously defined needs. This "descent" to the lowest point of the V reaches the characterization of the subsystems' most basic components, an activity that focuses on system *analysis*. From here on out, we begin to "ascend" the right branch of the V. This includes the product development and testing processes, all the way to completion. This activity is known as system *synthesis*.

The analysis and synthesis are based on different action patterns that require the use of different skills. Analytic systems engineers are tasked with development, design, and architecture, while synthetic systems engineers focus on implementation and integration.

Prof. Aviv Rosen explains: "Analysis is associated with the world of research, and its products are usually models for understanding various phenomena. Innovations often begin with analysis. Conversely, synthesis is the ability to bring components together and produce an engineering product. This is usually done by the industry. Synthesis is considered to be of a more routine nature, and was therefore perceived as inferior to analysis by the academy for many years. Research was thought of as a more lucrative practice, as it offered the possibility of discovering new things and publishing one's findings in scientific magazines. But times have changed, and the importance of synthesis has slowly increased in many engineering fields. Consequently, the rate of appearance of major technological innovations in these fields has diminished."

Henry Broodney also distinguishes between two types of systems engineering: he defines the first type as systems engineering that deals with planning and the process of designing the system itself. It focuses on what the project's lead systems engineer does, and Broodney refers to it as *Technical Systems Engineering*. The second type of systems engineering includes processes and work methods for managing engineering projects that combine scope, scheduling, and finances. The focus is on the actions of the project manager – on managing the system. This, he calls *Management-Oriented Systems Engineering*.

These two broad classifications are likely to include various systems engineering "subdisciplines," certainly in more complex projects. For instance, technical systems engineering includes systems engineers tasked with developing a certain component that pertains to a field of engineering they are well-versed in (e.g., a systems engineer trained in electronics engineering, in charge of developing an electro-optical component), and systems engineers tasked with integrating components from various disciplines, in the process of creating a comprehensive technological system that meets the client's needs. For example, The Iron Dome project employed systems engineers from all areas, including software, electronics, and mechanics, who worked alongside two systems engineers who coordinated all technological activity, within the framework of what AA referred to as "Lateral Systems Engineering." These two were interdisciplinary systems engineers, while the other systems engineers in the project worked within the boundaries of their specializations.