

chapter 1

Introduction

This book deals with the problem of the design and management of *complex* systems. Dommasch and Laudeman comment, “It is well to remember that any fool can design a complex system to do a simple job, but it is something else again to design a simple system to perform a complex task.”²

In order to develop the simplest solution to a complex problem, the problem must be well understood. Thus, a central concern in the development of complex systems is understanding. This involves the ability to understand the customer’s needs that are to be addressed by the system; to understand the context in which the system will function; to understand the solution to the problem as it is being developed; and finally to help the customer understand his/her problem and its proposed solution. As systems continue to become more complex, the problem of understanding becomes more acute. The author’s background is spacecraft systems. Like many complex systems, early spacecraft were relatively simple. However, especially after the advent of the computer, spacecraft have become increasingly complex.

I. Is a Structured Approach Needed?

While acknowledging that many companies do employ a structured approach to the development of their systems, many do not. Therefore, the following are offered as suggestions as to why a structured approach to complex system development is increasingly necessary.

Increasing Complexity — First, as just mentioned, systems are becoming increasingly complex. It used to be that the success of a new system development activity often depended upon the competence of the system engineer who knew everything about it. However, the days of those kinds of efforts being successful are numbered. Systems are becoming so complex that it is increasingly difficult to find any one person who is able to “keep it all in his head.”

² Dommasch, Daniel O. and Charles W. Laudeman, *Principles Underlying Systems Engineering*, New York: Pitman Publishing, 1962, p. 393.

Personnel Longevity — A second reason has to do with the longevity of people. The explosion of complexity, in large part, has taken place within the lifetime of the last generation. As those people who grew up with the industries now producing complex systems retire or otherwise leave the industries, who is left to “pick up the ball”? New people, who did not have the opportunity to learn the systems when they were relatively simple. These new people are left with the task of “picking up the ball,” not at the size of a snowflake, but after a system has snowballed to often avalanche-creating proportions. How are these people going to be able to do this effectively?

Workforce Discontinuities — Third, long careers in the same organization are becoming increasingly rare. Companies continue to merge, divest, reorganize, acquire, consolidate, relocate, globalize, centralize, and so on. Likewise, employees are becoming increasingly mobile, moving from one company to another every several years. This causes an instability and discontinuity in the workforce and therefore also on the development programs of which they are a part.

Intense Competition — Fourth, intense competition and limited financial resources are putting enormous pressure on development programs to reduce cost and cycle time. In order to remain competitive, these programs must be executed in an increasingly efficient manner.

This is certainly not an exhaustive list, but these things do identify a need to develop effective ways of dealing with the dynamics of a changing environment. Clearly there are many elements to an effective strategy of dealing with these and other pertinent issues. A central element of an effective strategy must include effective structuring of complex system development activities.

II. Technical and Managerial — Integration is Essential

The design and management of complex systems involves the execution of technical activities (e.g., requirements development, design, analysis, integration, verification, trade analyses, etc.) together with managerial activities (e.g., configuration management, risk management, etc.). Because of the necessary connection between them, these two sets of activities must be integrated. But how should this be done? The approach discussed herein involves clearly defining what the system development process is in terms of the technical activities, and then using it to develop the logical connection between the managerial and technical activities.

III. Motivation

Why study the topic of complex system design and management? The author concurs wholeheartedly with Wymore, who comments:

Every author has several motivations for writing, and authors of technical books always have, as one motivation, the personal need to understand; that is, they write because they want to learn, or to understand a phenomenon, or to think through a set of ideas.”³

Most anyone who has been involved in the development of a complex system has been struck by the amount of chaos that can develop so quickly. Requirements are not communicated to the right people. Sufficient resources are not available when they are needed. The customer does not know exactly what he wants or needs, thereby causing an instability in the top-level requirements. Heritage hardware and software bid in the proposal do not support the interfaces of the new system or perform to the levels required. Back-of-the-envelope analyses and engineering judgment do not agree with current, more detailed analyses, thus invalidating previous assumptions. The list could go on and on. It is this problem of controlling the chaos in complex system development that inspired the author’s fascination.

IV. Objectives

A central objective of this book is to develop a generalized approach to system development, such that the chaos which inevitably ensues can be controlled. More specifically, key objectives include:

- Developing an overarching generalized process that integrates all technical and managerial activities, that is applicable to all development phases from conceptual to detail design, and that is applicable to all levels of the design from top-level system to the lowest levels of the system hierarchy;
- Defining the System Development Framework (SDF) in sufficient detail so as to enable its implementation in the real world;
- Deriving a set of principles that serve to guide the person or teams involved in complex system development;
- Laying the foundation necessary to accurately model the System Development process;
- Clarifying where in the process iteration occurs and why; and
- Clarifying how functional decomposition is performed and its necessary relationship to implementation. This involves understanding the relationship between the “what” provided in specifications and the “how” developed in the implementation.

³ Wymore, Wayne A., *A Mathematical Theory of Systems Engineering — The Elements*, New York: John Wiley and Sons, 1967, p. v.

V. Key Questions

When setting up a program, the management team faces several important decisions that can have a significant impact on the performance of the contract. Some of these include:

- How should information flow and who is responsible for which interfaces?
- What are the technical activities and how should they be ordered?
- How should the managerial aspects of system development be structured?
- How should the technical and managerial activities be coupled?

These questions are addressed in this book.

VI. “System” Defined in the Literature

Since the main subject of this book has to do with the development of complex systems, it is necessary to discuss how the term is used herein. Hall says, “Systems Engineering is probably not amenable to a clear, sharp, one sentence definition.”⁴ There are, however, several definitions found in the current literature, and Hall offers his own: “A system is a set of objects with relationships between the objects and between their attributes.”⁵ Dommasch and Laudeman define “system” as follows:

A complete system is any complex of equipment, human beings, and interrelating logic designed to perform a given task, regardless of how complex the task may be. Logically, very large or complicated systems are broken into subsystems, to be fitted together like blocks to form the entire or total system.⁶

In his 1991 work, Rechtin defines a system as “a collection of things working together to produce something greater. . . . A system has the further property that it is unbounded — each system is inherently a part of a still larger system.”⁷ He continues:

1. A system is a complex set of dissimilar elements or parts so connected or related as to form an organic whole.

⁴ Hall, Arthur D., *A Methodology For Systems Engineering*, Princeton, NJ: D. Van Nostrand, 1962, p. 4.

⁵ *Ibid.*, p. 60.

⁶ Dommasch and Laudeman (1962), p. 182.

⁷ Rechtin, Eberhardt, *Systems Architecting: Creating and Building Complex Systems*, Englewood Cliffs: Prentice-Hall, 1991, p. 1.

2. The whole is greater in some sense than the sum of the parts, that is, the system has properties beyond those of the parts. Indeed, the purpose of building systems is to gain those properties.⁸

Rechtin with Maier suggest, “A system is a collection of different things which together produce results unachievable by themselves alone. The value added by systems is in the relationships of their elements.”⁹ Chase¹⁰, Wymore (1967)¹¹, Ellis and Ludwig¹², and Minds¹³ define a system as essentially “anything that performs work on an input in order to generate an output.”

VII. Working Definition of “System”

There is overlap between the physical system being developed and the processes used to effect the development. In the context of product development, Krishnan et. al. suggest that the product development process involves the

... transformation of input information about customer needs and market opportunities into output information which corresponds to manufacturable designs.
... Individual development activities are themselves viewed as information processors, receiving input information from their preceding activities, and transforming this input information into a form suitable for subsequent activities.¹⁴

This is an excellent summary statement as to how each element of the development process is viewed, as well as how each element of the system itself is viewed in this book. Any complex system composes several sub-systems, which, in turn, compose more sub-subsystems. Each of these “systems” receives input, does work on that input, and then generates an output. This is an important point because it suggests that any process applicable to the development of a system must, by this definition, be applicable to the development of its subsystems. This principle lays the foundation for a modular approach to the process definition.

⁸ *Ibid.*, p. 28.

⁹ Rechtin, Eberhardt and Mark W. Maier, *The Art of Systems Architecting*, Boca Raton, FL: CRC Press, 1997, p. 21.

¹⁰ Chase, Wilton P., *Management of System Engineering*, New York: John Wiley & Sons, 1974, p. 54.

¹¹ Wymore (1967), p. 21.

¹² Ellis, David O., Fred J. Ludwig, *Systems Philosophy*, Englewood Cliffs: Prentice-Hall, 1962, p. 3.

¹³ Minds, Kevin S., “System Engineering The People System,” NCOSE 1995, P066.

¹⁴ Krishnan, Viswanathan, Steven D. Eppinger, Daniel E. Whitney, “A Model-Based Framework to Overlap Product Development Activities,” *Management Science*, Vol. 43, No. 4, pp. 437-451, April 1997.

For this book, the term “system” will be defined as “any entity within prescribed boundaries that performs work on an input in order to generate an output.” Note further that a system’s external interfaces consist of those entities or realities that impinge upon the prescribed boundaries. The system exists within a definable physical space and within a specific timeframe.