

## chapter 2

---

# Literature Search and Rationale for this Book

*You can observe a lot just by watchin’.*  
—Yogi Berra

*You can’t get where you want to go if you don’t know where you are.*  
—Anonymous

### *I. Existing and Emerging Standards*

While the act of system engineering is as old as man’s first efforts to engineer systems, as a defined discipline it is relatively new.<sup>15</sup> Although effort is being expended to change things, in many circles, there remains little consensus on nomenclature, metrics, or the system engineering process itself. However, as will be indicated in this chapter by way of a survey of the literature, there is significant agreement regarding which activities must be performed as part of the system engineering process.

Table 2.1 provides a summary survey of two emerging non-government standards (IEEE 1220-1994, EIA/IS-632)<sup>16</sup> as well as three key military standards (Defense Systems Management College [DSMC] Systems Engineering Management Guide, Mil-Std-499A, and Army Field Manual 770-78).

### *II. Individual Works*

The literature provides many views of the system engineering process. Shinnars defines a “closed loop, iterative process.”<sup>17</sup> Reinert and Wertz define

<sup>15</sup> For example, Dommasch and Laudeman assert that the term “system engineering” originated with the Manhattan Project. *Principles Underlying Systems Engineering*, p. iii.

<sup>16</sup> See Rechtin and Maier for a critique/comparison of these two standards, pp. 218-219. Note also the Mil-Std-499B was never released and has subsequently evolved to EIA/IS-632.

<sup>17</sup> Shinnars, Stanley M., *A Guide to Systems Engineering and Management*, Lexington, MA: D. C. Heath and Company, 1976, pp. 14-17.

**Table 2.1** Civilian and Military Systems Engineering Standards

IEEE 1220-1994 <sup>a</sup>	EIA/IS-632 <sup>b</sup> (Was Mil-Std-499B <sup>c</sup> )	DSMC Systems Engineering Management Guide <sup>d</sup>		Army Field Manual 770-78 <sup>f</sup>	Consensus
		Functional Analysis	Mission Requirements Analysis and Functional Analysis		
Requirements Analysis	Requirements Analysis Functional Analysis/ Allocation	Functional Analysis	Mission Requirements Analysis and Functional Analysis	Function Analysis	<b>“What”</b>
Synthesis	Synthesis	Synthesis	Allocation and Synthesis	Synthesis	<b>“How”</b>
Systems Analysis	Systems Analysis and Control	Evaluation and Decision	Optimization: Effective Engineering Analysis	Evaluation and Decision	<b>“How Well”</b>
Functional and Physical Verification	Verification (defined as a feedback)		Production Engineering Analysis		<b>“Verify”</b>
Trade Studies and Assessments		Evaluation and Decision	Optimization: Trade Studies	Evaluation and Decision	<b>“Select”</b>

<sup>a</sup> IEEE 1220-1994, *Trial-Use Standard for Application and Management of the Systems Engineering Process*, Clause 6, 1995.

<sup>b</sup> EIA/IS 632, *Systems Engineering*, pp. 7-12.

<sup>c</sup> MIL-STD-499B.

<sup>d</sup> DSMC, *Systems Engineering Management Guide*, 1990, pp. 5-1 to 8-19, Cf. DSMC 1986 *Systems Engineering Management Guide*.

<sup>e</sup> MIL-STD-499A, *Engineering Management (USAF)*, Chapter 10.

<sup>f</sup> Army Field Manual 770-78, April 1979, [Figure 2-1](#).

a “concept exploration flow.”<sup>18</sup> Coutinho describes his process as a “systems development and test cycle.”<sup>19</sup> Hall delineates a view that captures “the open systems nature of the systems engineering process as it exchanges energy, information, and materials with the environment.”<sup>20</sup> Blanchard and Fabrycky emphasize a “sequential and iterative methodology to reach cost-effective solutions to design alternatives.” They continue, “Systems engineering is directly concerned with the transition from requirements identification to a fully defined system configuration ready for production and ultimate cus-

<sup>18</sup> Reinert, Richard P. and James R. Wertz, *Space Mission Analysis and Design*, Eds. Wiley J. Larson and James R. Wertz, published jointly by Microcosm, Inc., Torrance, CA and Kluwer Academic Publishers, The Netherlands, 1992, p. 20.

<sup>19</sup> John de S. Coutinho, *Advanced Systems Development Management*, New York: John Wiley & Sons, 1977, pp. 35-51.

<sup>20</sup> Hall, pp. 138-139.

tomers use.”<sup>21</sup> Chase identifies what he calls “the irreducible gross functional steps which must be followed [in any system development activity].”<sup>22</sup> Wymore has developed a list of “archetypal” questions that are reflective of a sound system engineering process: What is the system supposed to do? What is available to build the system? How well must the system perform? What are the cost/performance trades? How can the system be verified?<sup>23</sup> Each of these are summarized in Table 2.2.

As Tables 2.1 and 2.2 indicate, there is general consensus among the various system engineering process standards in terms of the technical activities that must be performed.<sup>24</sup> Each of the general activities identified in each of the sources surveyed may be categorized into one of the following five broad categories: “what,” “how,” “how well,” “verify,” or “select” activity. It is asserted that *any* technical development activity can be categorized under one of these categories. This is the basic organizing concept of the System Development Framework (SDF) derived in this book and is illustrated in Figure 2.1.

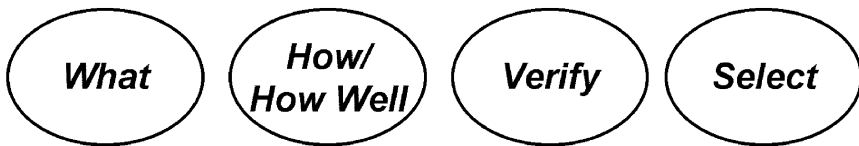


Figure 2.1 System Development Framework (SDF) Organizing Concept.

### III. The Basic Building Block

There is a logical flow to the sequencing of these activities. There must, first of all, be some input, however high or low its fidelity. This input must be analyzed to determine “what” the system must do. This activity is called Requirements Development. The next activity — Synthesis — generates alternatives describing “how” the “what” might be implemented. Synthesis also determines “how well” each alternative performs, as well as verifying that the design meets the objectives. If more than one alternative emerges, the best is chosen through a “selection” process. To reiterate, Figure 2.1 illustrates this top-level flow of activity which is the SDF Organizing Concept. It is used to organize the various engineering activities and subprocesses into an overall integrated process.

<sup>21</sup> Blanchard, Benjamin S. and Wolter J. Fabrycky, *Systems Engineering And Analysis*, Englewood Cliffs, NJ: Prentice-Hall, 1981, pp. 236-240.

<sup>22</sup> Chase, pp. 7-11.

<sup>23</sup> Wymore, Wayne A., *Model-Based Systems Engineering*, Boca Raton, FL: CRC Press, 1993, p. 7.

<sup>24</sup> Cf. White, Michelle M., James A. Lacy, and Edgar A. O’Hair, Refinement of the Requirements Definition (RD) Concept in a System Development: Development of the RD Areas, “Systems Engineering Practices and Tools,” *Proceedings Sixth Annual Symposium INCOSE*, Vol. 1, July 7-11, 1996, Boston, MA, pp. 749-756.

*Table 2.2* Individual Works

Shinners	Reinert and Wertz	Coutinho	Hall	Blanchard and Fabrycky	Chase	Wymore	Consensus
Mission and Requirements Analysis and Functional Analysis	Define Preliminary Requirements	Operational Requirements	Problem Defining	System Functional Analysis	Mission Objectives, System Performance Requirements, Operations and Support Requirements	“What” is the System Supposed to Do?	<b>“What”</b>
Synthesis of the System	ID Concepts, Define Mission Architecture	System Design	Synthesis and Design	Synthesis and Design	System Design and Synthesis	What is Available to Build the System?	<b>“How”</b>
Effectiveness Analysis Simulations	Evaluate Concepts	Systems Analysis	Analysis	Optimization and Analysis	System Analysis	“How Well” must the System Perform?	<b>“How Well”</b>
Test of System	Generate Measures of Effectiveness	Prototype Design, Fab and Test Plan, Acceptance	Test	Prototype, Test and Evaluation	Evaluation of System Design Adequacy	How can the System be Verified?	<b>“Verify”</b>
Tradeoff Analysis	System Level Trades		Decision Making		Select Alternative System Design Approaches, Perform Trade-Off Studies	What are the Cost/ Performance Trade-Offs?	<b>“Select”</b>

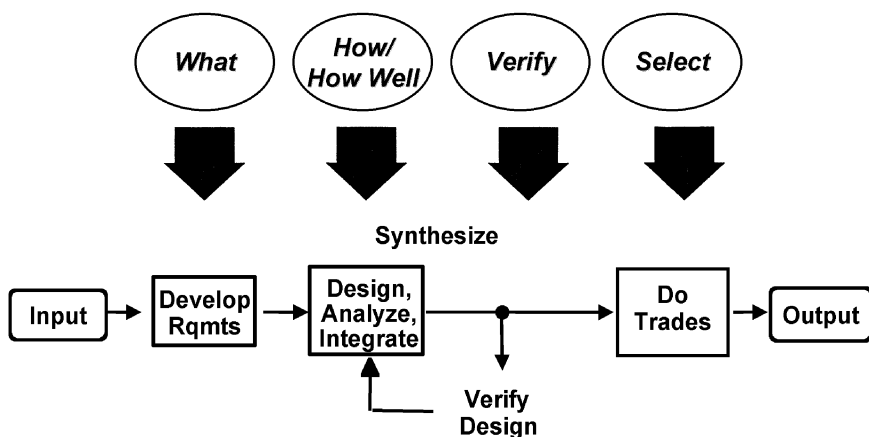


Figure 2.2 The SDF Basic Building Block

Figure 2.2 represents this sequence as the basic building block of the SDF. It represents one “module” of activity. It is applied to each system, subsystem, sub-subsystem, etc. of the program hierarchy. A detailed lower level decomposition of this basic building block is developed in Chapter 5.

## IV. Unique Features of this Book

In the above discussion, the fundamental building block of the SDF has been developed, based upon a consensus derived from the system engineering literature. At this point one may ask, “Why another book on the system engineering process?” There are several unique features in this present work.

### A. Time and Logical Domains

The time and logical domains have been identified and characterized in distinction, thus enabling the SDF’s application to many contexts. There have been and continue to be many efforts focused on defining the elusive generic System Engineering Process. It is suggested that one reason why industry, government, and academic efforts have had limited success in defining a generalized process applicable to many contexts, is that the time and logical domains (discussed in detail in Chapter 3) have not been explicitly identified and characterized in distinction. When the Logical Domain view is combined with the Time Domain view, the resulting process often becomes application specific. When these are characterized in distinction, the overall framework can be preserved. This book develops a generalized process that maintains this distinction and is thus applicable to many contexts.

### B. Tier Connectivity

Many system engineering processes are defined with reference to one tier of activity. They do usually acknowledge multiple tiers (i.e., system subsystem,

sub-subsystem). However, in general, they do not precisely describe how the various tiers are coupled in terms of logical and consistent flow-down and feedback paths. This book clearly defines the coupling between tiers, which has implications regarding information flow, roles and responsibilities, change impact analyses, risk management, etc.

### *C. Modularity*

The SDF defined in this book is truly modular. Through prudent partitioning of the system into focused modules, the SDF is tailored to meet the demands of individual development programs.

### *D. Coupling of Technical and Managerial Activities*

The design and management of complex systems involves the execution of technical activities together with managerial activities. Because of the organic connection between these two sets of activities, they must be integrated in order to maximize the potential for success. This integration requires a clear definition of what the system development process is in terms of the technical activities and how they logically interact. In this book, the “control logic” (see [Chapter 5](#)) provided by the SDF is used to develop the logical connection between the managerial and technical activities.

### *E. Clear Presentation of Functional Decomposition*

In some circles, there is significant confusion regarding system development by functional analysis and decomposition. This book attempts to provide a clear and logical approach to this important activity.

### *F. Explicit Inclusion of the Rework Cycle*

Within the discipline of System Dynamics, the rework cycle was developed some years ago and it has been adapted into the overall SDF. This is essential for an accurate understanding of real-world system development dynamics and accurate modeling of any development activity.

### *G. Explicitly Defined Generalized Outputs*

As one 40-year veteran system engineer put it, “Many system engineering processes have been defined, but few remember the main point — output.” Therefore, an indication of what each activity must produce in terms of generalized output has been provided. That output then becomes the input to subsequent activities.