CHAPTER 2.2

ELEMENTS AND TECHNIQUES OF SYSTEMS ENGINEERING AND MANAGEMENT

Andrew P. Sage, William B. Rouse

SYSTEMS ENGINEERING AS A MANAGEMENT TECHNOLOGY

Systems engineering is a management technology. Technology is organization, application, and delivery of scientific and other forms of knowledge for the betterment of a client group. This is a functional definition of technology as a fundamentally human activity. A technology inherently involves a purposeful human extension of one or more natural processes. For example, the stored program digital computer is a technology in that it enhances the ability of a human to perform computations and, in more advanced forms, to process information.

Management involves the interaction of the organization with the environment. A purpose of management is to enable organizations to better cope with their environments so as to achieve purposeful goals and objectives. Consequently, a management technology involves the interaction of *technology, organizations* that are collections of *humans* concerned with both the evolvement and use of technologies, and the *environment.* Figure 2.2.1 illustrates these conceptual interactions. Information and associated knowledge represents the “glue” that enables the interactions shown in this figure. Information and knowledge are very important quantities that are assumed to be present in the management technology that is systems engineering. This strongly couples notions of systems engineering with those of technical direction or systems management of technological development, rather than exclusively with one or more of the methods of systems engineering, important as they may be for the ultimate success of a systems engineering effort.

Figure 2.2.2 illustrates the view that systems engineering knowledge comprises:

1. *Knowledge principles*—which generally represent formal problem solving approaches to knowledge, generally employed in new situations and/or unstructured environments.
2. *Knowledge practices*—which represent the accumulated wisdom and experiences that have led to the development of standard operating policies for well-structured problems.
3. *Knowledge perspectives*—which represent the view that is held relative to future directions and realities in the knowledge areas under consideration.

Clearly, one form of knowledge leads to another. Knowledge perspectives may create the incentive for research that leads to the discovery of new knowledge principles. As knowledge principles emerge and are

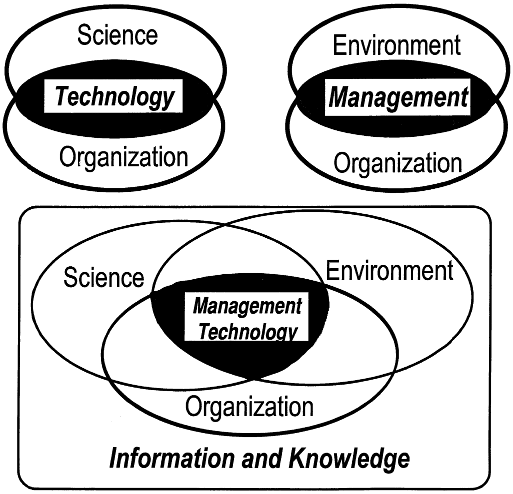


FIGURE 2.2.1 Systems engineering as a management technology.

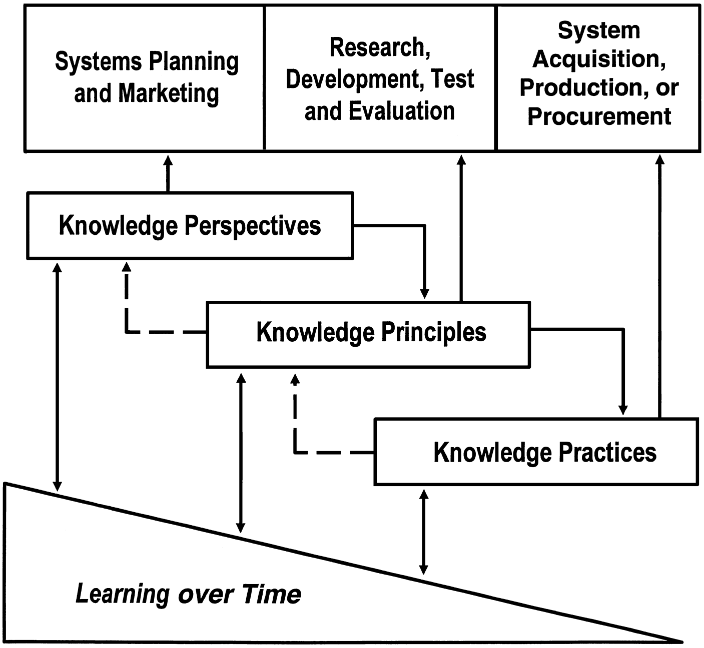


FIGURE 2.2.2 Knowledge types and support for systems engineering efforts.

refined, they generally become embedded in the form of knowledge practices. Knowledge practices are generally the major influences of the systems that can be acquired or fielded. These knowledge types interact as suggested in Fig. 2.2.2, which illustrates how these three types of knowledge support one another. In a nonexclusive way, they each support one of the principal life cycles associated with systems engineering. Figure 2.2.2 also illustrates a number of feedback loops that are associated with learning to enable continual improvement in performance over time. This supports the view that it is a mistake to consider these life cycles in isolation from one another.

It is on the basis of the appropriate use of the three knowledge types—principles, practices, and perspectives—depicted in Fig. 2.2.2, that we are able to accomplish the technological system planning and development and the management system planning and development that lead to new and innovative products, technologies, and services. All three types of knowledge are needed. The environment associated with this knowledge needs to be managed, and this is generally what is intended by use of the term *knowledge management*. Also, the learning that results from these efforts is very much needed, both on an individual and an organizational basis.

LIFE CYCLES

Three Essential Life Cycles. There are three major and quite purposefully different life cycles for technology evolution through systems engineering: system planning and marketing; research, development, test and evaluation (RDT&E); and system acquisition, production, or procurement. These are needed for evolution of trustworthy products and services, and each involves use of one of the three types of knowledge: knowledge perspectives, knowledge principles, and knowledge practices.

Systems engineers are concerned with the appropriate definition, development, and deployment of systems. These comprise a set of phases for a systems engineering life cycle. There are many ways to describe the life-cycle phases of systems engineering life-cycle processes. Each of these basic life-cycle models, and those that are outgrowths of them, comprise these three phases of definition, development, and deployment. For pragmatic reasons, a typical life cycle will almost always contain more than three phases. Generally, they take on a "waterfall" like pattern, although there are a number of modifications of the basic waterfall, or "grand design" life cycle, to allow for incremental and evolutionary development of systems. Figure 2.2.3 represents such a life cycle. It shows the three basic phases of efforts associated with a systems engineering life cycle. It also shows an embedding of the notions of three basic steps within these phases.

Figure 2.2.4 suggests the essential three systems-engineering life cycles. Each is responsive to a particular question:

- Planning and Marketing: What is in demand?
- RDT&E: What is possible?
- Acquisition: What can be developed?

It is only in the response space that is common to all three questions, as suggested in Fig. 2.2.5, that it will truly be feasible to build a trustworthy productive product or system. The life-cycle processes shown in Figs. 2.2.3 and 2.2.4 will often need to be repeated in an evolutionary manner to accommodate successive builds of a technology product.

DEVELOPING TRUSTWORTHY SYSTEMS

Management of systems engineering processes, which we call systems management (Sage, 1995), is necessary for success. Systems engineering may fail at the level of systems management. Often, the purpose, function, and structure of a new system are not identified sufficiently before the system is defined, developed, and

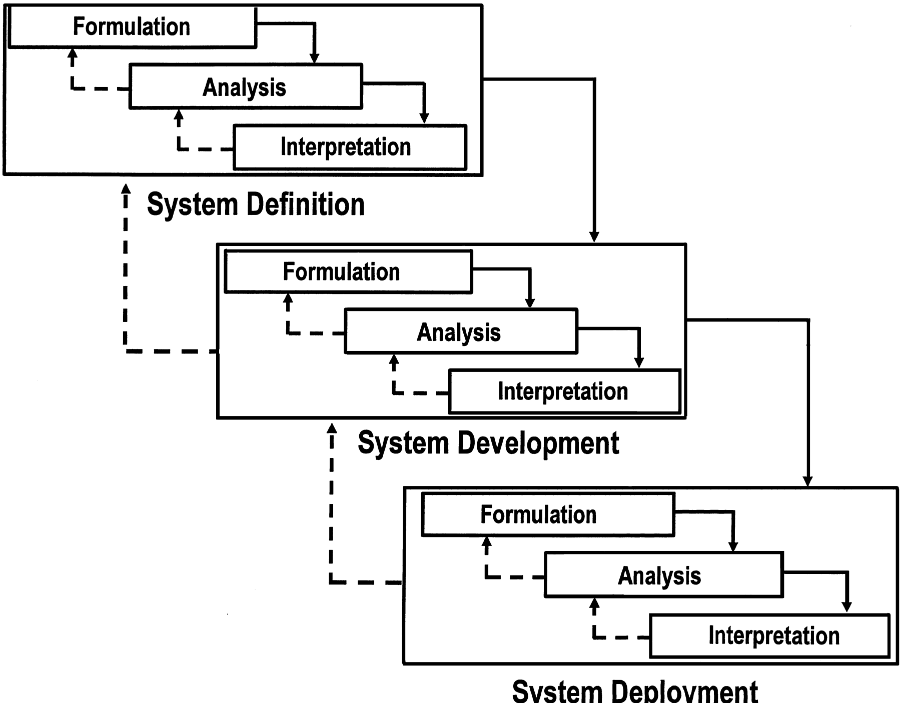


FIGURE 2.2.3 A systems engineering life cycle comprising of three phases and three steps per phase.

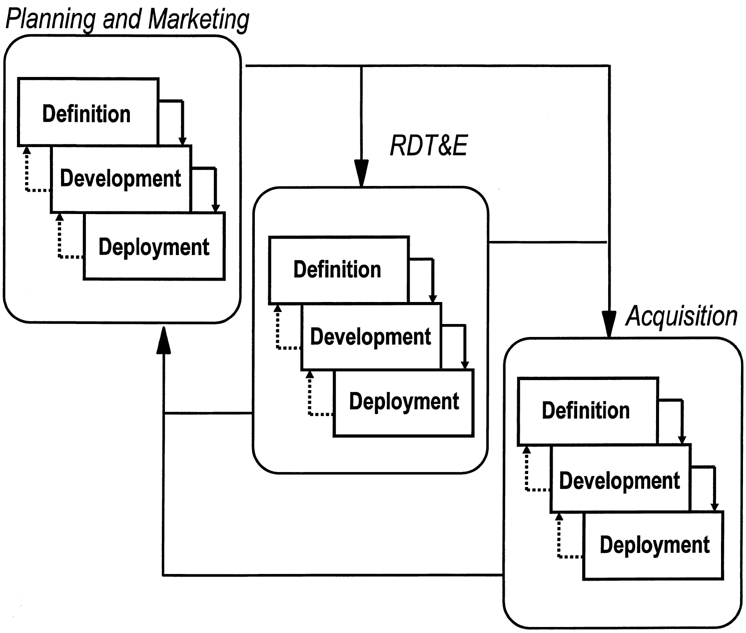


FIGURE 2.2.4 Major systems engineering life cycles and three phases within each.

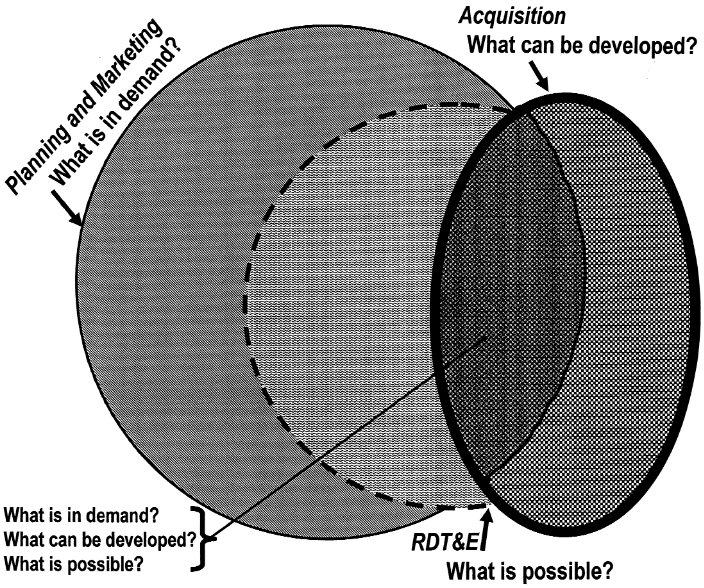


FIGURE 2.2.5 Illustration of the need for coordination and inteqration across life cycles.

deployed. These failures, generally, are the result of costly mistakes that could have been avoided by more diligent use of systems engineering and systems management.

System management and integration issues are of major importance in determining the effectiveness, efficiency, and overall functionality of system designs. To achieve a high measure of functionality, it must be possible for a system to be efficiently and effectively produced, used, maintained, retrofitted, and modified throughout all phases of a life cycle. This life cycle begins with need conceptualization and identification, through specification of system requirements and architectures, to ultimate system installation, operational implementation or deployment, evaluation, and maintenance throughout a productive lifetime.

There are many difficulties associated with the production of functional, reliable, and trustworthy systems of large scale and scope. Among them are:

1. Inconsistent, incomplete, and otherwise imperfect system requirements and specifications
2. System requirements that do not provide for change as user needs evolve over time
3. Lack of proper concern for satisfying the functional needs of the customer and inappropriate attention to assurance that the overall life cycle of product development and use is conservative of natural resource use
4. Poorly defined management structures for product, or service, development, and deployment

These lead to delivered products and services that are difficult to use, that do not solve the intended problems, that operate in an unreliable fashion, that are not maintainable, that are overly consumptive of natural resources and damaging to the environment, and that, as a result, may become quickly obsolete. Sometimes these failures are so great that products and services are never even fully developed, much less operationally deployed, before they are abruptly canceled.

The major problems associated with the production of trustworthy systems often have more to do with the organization and management of complexity than with direct technological concerns that affect individual subsystems and specific physical science areas. Often more attention should be paid to the definition, development, and use of an appropriate process for production of a product than to the actual product itself. Direct attention to the product or service without appropriate attention to the process leads to the fielding of a low quality and expensive product or service that is unsustainable.

Management and Metrics. Systems engineering efforts are very concerned with technical direction and management of the process of systems definition, development, and deployment, or systems management. Through adopting and applying the management technology of systems engineering, we attempt to be sure that correct systems are designed, and not just that system products are correct according to some potentially ill-conceived notions of what the system should do. Appropriate metrics to enable efficient and effective error prevention and detection at the level of systems management, and at the process and product level will enhance the production of systems engineering products that are "correct" in the broadest possible meaning of this term. To assure that correct systems are produced requires that considerable emphasis be placed on the front-end of each of the systems engineering life cycles.

ACCURATE DEFINITION OF A SYSTEM

There must be considerable emphasis on the accurate definition of a system, what it should do, and how people should interact with it before one is produced and implemented. In turn, this requires emphasis on conformance to system requirements and specifications, and the development of standards to ensure compatibility, integratibility, and sustainability of system products and services. Such areas as documentation and communication are important. Thus, we see the need for the technical direction and management technology efforts that comprise systems engineering, and the strong role for process and systems management related concerns.

Ingredients associated with the development of trustworthy systems include the following:

- Systems engineering processes, including process development life cycles and process configuration management
- Process risk management, operational level quality assurance and evaluation, and product and process development standards and associated maturity models
- Metrics for quality assurance, to ensure standardization, and for process and product evaluation
- Metrics for cost estimation, and product cost and operational effectiveness evaluation
- Strategic quality assurance and management, or total quality management
- Organizational cultures, leadership, and process maturity
- Reengineering at the levels of systems management, organizational processes and product lines, and products

These ingredients and issues often strongly interact. One of the first efforts in systems management is to identify an appropriate process life cycle that is sustainable and that will lead to production of a trustworthy and sustainable system. This life cycle for the engineering of a system involves a sequence of phases. These phases include identification of client requirements, translation of these requirements into hardware and software requirements specifications, development of system architectures and conceptual designs of the system, detailed design and production of the system, operational implementation and evaluation, and maintenance of the delivered system over time. The precise life cycle that is followed will depend on the client needs. It will also depend on such overall environmental factors as the presence of existing system components, or subsystems, into which a new system must be integrated, natural resource considerations, and the presence of existing software modules that may be retrofitted and reused as part of the new system. These needs generally result in continued evolution and emergence of solutions that are adaptive to organizational needs that are similarly evolving and emergent.

SYSTEM REQUIREMENTS

A necessary first step in the engineering of any system is to determine just what the user wants, the purpose that is to be fulfilled by the resulting system in fulfilling these needs, and the necessary accompanying nonfunctional requirements, such as reliability and quality of the system. The resulting *definition* of user functional needs and associated nonfunctional requirements represent the first and most important phase in the effort to engineer a system. These requirements must be clear, concise, consistent, unambiguous, and represent user goals and purposes for the system to be engineered. A process to identify user needs and requirements must include:

1. Identification of purposes for the system, and associated constraints, functions, and operational environment; and translation of these into a set of functional requirements.
2. Assessment of the allowable risks and the desired quality of the system, and translation of these into a set of nonfunctional requirements
3. Transformation of these functional and nonfunctional requirements into the concise and formal language of a technical specification.

Successful systems engineering is very dependent on the ability to identify user needs in the form of (functional and nonfunctional) requirements and to reflect them in specifications for the delivered system. User needs and requirements that are complete, correct, consistent, and error free, play a major role in ensuring that the delivered system meets the intended purpose.

While there is widespread and general acceptance of the necessity to provide adequate requirements definition, there is often controversy as to the ultimate need, purpose, and cost of performing the requirements and system definition associated activities that are necessary to assure that the engineered system meets user needs. The controversy arises both because detailed system developers often do not visualize that the benefits that may accrue to the final product through implementation of what may be viewed as an expensive total requirements process is comparable to the time and effort involved. System users generally do not understand the complexity associated with transforming their needs to requirements and specifications documents that detailed system developers can use.

Production of well-understood and well-developed requirements information, and the identification and management of risk in the requirements engineering, is a major concern to assure production of systems that meet user needs by engineering a system that is delivered on time and within budget. Many studies suggest that between 50 and 80 percent of the errors found in deployed systems can be traced to errors in identification and interpretation of requirements.

Difficulties related to requirements definition generally revolve around the necessity to elicit or develop requirements information from a variety of sources, including users and/or existing systems. There is usually a lack of definitive information from the user and a lack of agreement on mechanisms to transform these concepts to technical specification documents for use by the detail system developer. These difficulties lie at the interface between the humans involved in the systems engineering process, especially system users and detailed system developers. Once user requirements have been established, these must be transformed to the exact language of technical system specifications; another source of potential problems. There are also potential technical difficulties that relate to hardware, performance, capacity, interfaces, and other issues.

Most often, user originated system level requirements are stated in natural language, and this brings about a high possibility of incorporation of ambiguities, conflicts, inconsistencies, and lack of completeness in the resulting requirements. These problems must be addressed prior to transformation from the informal language of system users' requirements to the formal or semiformal languages of detailed system design. Otherwise, these deficiencies may be incorporated in the final engineered product rather than being resolved at the time of requirements development. If this occurs, these requirements deficiencies must be corrected during later phases of systems development, or following system deployment provided the system is considered to be acceptable at all by the user such as to merit initial deployment.

Strategies for Determining Information Requirements

Davis (1982) has identified four strategies for determining information requirements. Taken together, these yield approaches that may be designed to ameliorate the effect of three human limitations: limited information processing ability, bias in the selection and use of information, and limited knowledge of what is actually needed.

1. The first strategy is to simply ask people for their requirements. The usefulness of this approach will depend on the extent to which the interviewers can define and structure issues and compensate for biases in issue formulation. There are a variety of methods that can be used to assist in this.
2. The second strategy is to elicit information requirements from existing systems that are similar in nature and purpose to the one in question. Examination of existing plans and reports represent one approach of identifying information requirements from an existing, or conceptualized, system.
3. The third strategy consists of synthesizing information requirements from characteristics of the utilizing system. This permits one to obtain a model or structure for the problem to be defined, from which information requirements can be determined. This strategy would be appropriate when the system in question is in a state of change and thus cannot be compared to an existing system.
4. The fourth strategy consists of discovering needed information requirements by experimentation, generally through constructing a prototype. Additional information can be requested as the system is employed in an operational, or simulated setting, and problem areas are encountered. The initial set of requirements for the system provides a base point for the experimentation. This represents an expensive approach, but is often the only alternative when there does not exist the experience base to use one of the other approaches. This approach is therefore equivalent to use of a prototype, either an evolutionary prototype or throwaway prototype.

Each of these four strategies has advantages and disadvantages, and it is desirable to be able to select the best mix of strategies. One's choice will depend on the amount of risk or uncertainty in information requirements that results from each strategy. Here, uncertainty is used in a very general sense to indicate information imperfection.

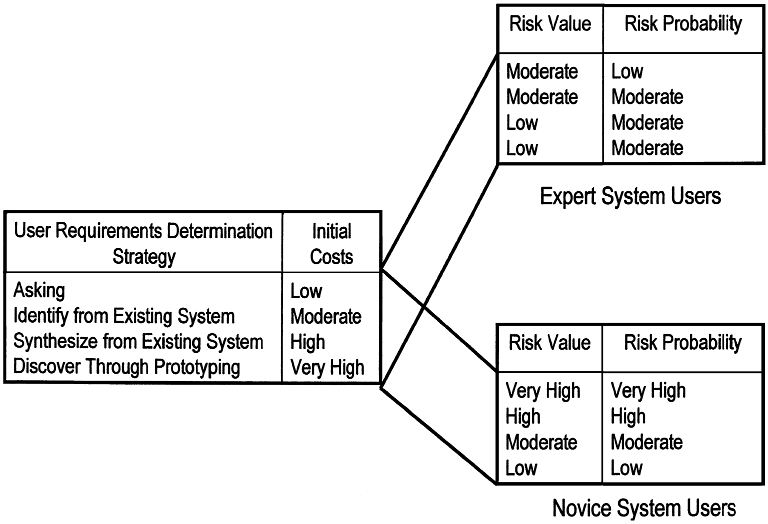
Information Uncertainties

Five steps are useful in identifying information uncertainties and then selecting appropriate strategies.

1. Identify characteristics of the utilizing system, technology system, users, and system development personnel as they affect information uncertainty.
2. Evaluate the effect of these characteristics on three types of information requirements determination uncertainties:
   - ***a.*** Availability of a set of requirements
   - ***b.*** Ability of users to specify requirements
   - ***c.*** Ability of systems engineers to elicit and specify requirements
3. Evaluate the combined effect of the requirements determination process uncertainties on overall requirements volatility.
4. Select a primary requirements determination strategy.
5. Select a set of specific steps and methods to implement the primary requirements determination strategy.

These steps may be used to identify an appropriate mix of requirements identification strategies. The uncertainty associated with requirements determination, that is to say the amount of information imperfection that exists in the environment for the particular task, influences the selection from among the four basic strategies as indicated in Fig. 2.2.6. This illustrates the effects of experiential familiarity on the part of the system users that are used by a generally experienced requirements engineering team to identify requirements. The factors that influence this information imperfection include:

1. Stability of the environment
2. Stability of organizational management and system users
3. Previous experience of system users with efforts associated with systems engineering and management
4. The extent to which there exists a present system that is appropriate
5. The extent to which a change in requirements will change the usage of present system resources and thereby degrade the functionality of legacy systems and result in this needing to be considered as part of the overall effort



User Requirements Determination Strategy	Initial Costs
Asking	Low
Identify from Existing System	Moderate
Synthesize from Existing System	High
Discover Through Prototyping	Very High

Risk Value	Risk Probability
Moderate	Low
Moderate	Moderate
Low	Moderate
Low	Moderate

Risk Value	Risk Probability
Very High	Very High
High	High
Moderate	Moderate
Low	Low

FIGURE 2.2.6 Effects of user familiarity on selecting requirements-determination strategy.

Used appropriately, this effort should enable selection of an appropriate requirements determination approach that can cope with the three essential contingency dependent variables owing to risk elements in requirements determination. To fully enable this it is desirable to also consider the effects of experiential familiarity of the requirements engineering team with both requirements determination in general and for the specific task at hand.

Requirements determination must lead to technological specifications for detailed design and fielding of the system. A purpose of the system definition phase of the life cycle is to determine possibilities of insufficient and/or inappropriate information; that is, information that is sufficiently imperfect such as to make the risk of an unacceptable design too high to be tolerated. The requirements elicitation team should be able to determine the nature of the missing or otherwise imperfect information, and suggest steps to remedy this deficiency. These are truly major efforts in systems and software engineering efforts today and are the subject of many contemporary writings on requirements (Andriole, 1996; Robertson and Robertson, 1999; Sommerville and Sawyer, 1997; Thayer and Dorfman, 1997; Young, 2001).

SYSTEMS ARCHITECTING, DESIGN, AND INTEGRATION

Stakeholder Viewpoints

From a customer point of view, a system is everything that is required to meet the need of the customer to achieve some particular purpose. A system designer may view the system as the product or service described by the design specifications. A logistician may view a system as the system maintenance and logistics efforts. A systems architect may view the system as the structures and interfaces needed to represent the system in a conceptual fashion. Systems engineering is concerned with the total view of the system and encompasses each of these perspectives, and others. Systems engineering must necessarily relate to the enterprise or organization for which the system is being built. It is necessarily concerned with the process for building the system. It is concerned with the systems management and technical direction relative to the implementation agents that construct the system. It is concerned with the technologies and metrics associated with constructing the systems as

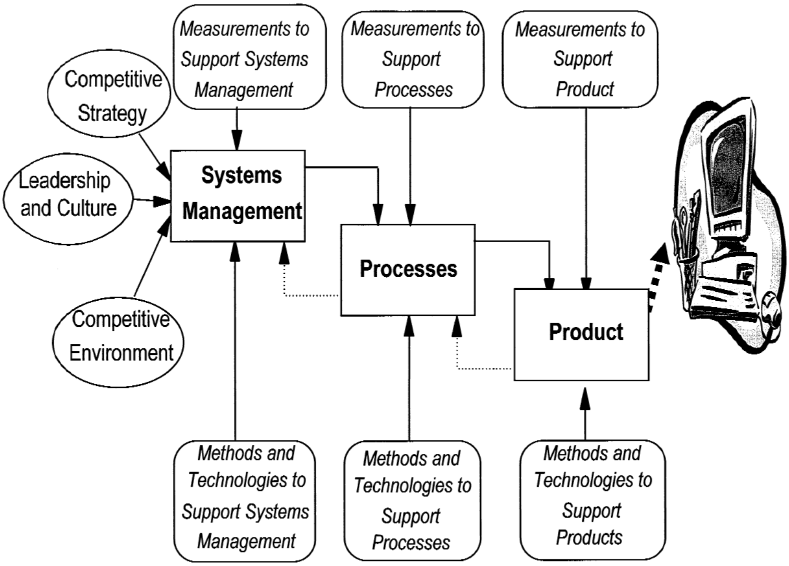


FIGURE 2.2.7 A model for engineering of a system.

well as those associated with the technical direction of this effort. It is concerned with the environment surrounding all of these. Thus, systems engineering is concerned with:

- People
- Processes
- Technologies, in the form of methods and tools
- Metrics
- Systems management
- Environments

All of these considerations are needed in order to engineer a system in the form of a product, service, or process that supports customer needs in an effective manner. Figure 2.2.7 (Sage, 1995) presents a model of system engineering that incorporates these elements. Thus, we see that a multiple perspective view of systems engineering and management is needed. Also needed is a multiple perspective view of systems architecting, design, and integration as an essential ingredient in systems engineering and management.

Integration Issues

Integration is defined in conventional dictionaries as the making of a whole entity by bringing all of the components of that entity together. In other words, integration involves blending, coordinating, or incorporating components into a functioning whole. This definition, while appropriate for systems integration, needs greater specificity in order to illustrate the many facets of integration. We may immediately think of product system integration. This refers to integration of the product or service, which is delivered to the customer. This is the most often used context for the term systems integration. To accomplish external systems integration effectively requires that new system elements be appropriately interfaced with existing, or legacy, systems components. Sometimes these legacy systems are sufficiently “stovepipe” in nature such that the integration with a new system is not possible without reengineering the legacy system. It also requires that it is possible to accomplish integration with later arriving, and potentially unplanned for initially, additional systems elements.

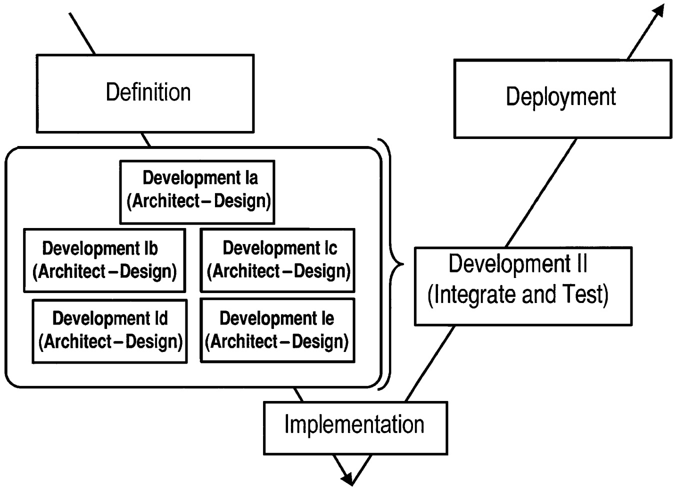


FIGURE 2.2.8 Concurrent engineering "V" model illustrating the need for integration following architecting and design.

This requires much attention to systems level architecting, and design and integration architectures (Sage and Lynch, 1998; Maier and Rechtin, 2001).

This need for system integration brings about a host of systems management and, in many cases, legal and regulatory issues that are much larger in scale and scope than those associated with product development only. In a similar manner, the development of appropriate system level architectures is very important in that efficiency and effectiveness in systems architecting is very influential of the ease with which systems can be integrated and maintained and, therefore, of the extent to which an operational system is viewed as trustworthy in satisfying user functional needs and nonfunctional needs related to quality and sustainability.

Functional, Physical, and Operational Architectures

As we have noted, the life cycle of a system comprises a number of phases that, when completed results in the satisfactory definition, development, and deployment of a system. Three phases are usually not sufficient (without each of them becoming vastly complicated) to represent the plethora of activities needed to properly engineer a system. There are such efforts as the engineering of the functional architecture, followed by the physical architecture, and following this the development of an implementation or operational architecture. This requires detailed design and associated production or manufacturing, operations and support, and other systems deployment efforts that may include disposal or reengineering. These need to be included as phased efforts in any realistic life cycle. Figure 2.2.8 recasts the basic three phase model for engineering a system into a "V" model that more explicitly shows the role of architecting, design and integration in the systems engineering life cycle. Sometimes, the initial efforts in the systems engineering life cycles of Fig. 2.2.8 are called "down-stroke" efforts and those in the latter part of the life cycle are called "up-stroke" efforts.

Not explicitly shown in Fig. 2.2.8 are the drivers for the engineering of a system and how these influence the need for systems integration. It is a rare situation that a new stand-alone system is called for and where this system does not have to be compatible with or interfaced to an existing, or legacy, system. There also exists such drivers for change as:

- A changing technology base because of the emergence of new and innovative technologies and improvements in existing technologies
- Changing organizational needs because of newly emerged competitors, the desire to acquire additional critical core capabilities
- Changing human needs because of the need for better knowledge management and enhanced organizational learning

Thus, we see that systems integration needs and considerations find importance in the:

- Planning effort that results in the very high level definition of a "new" system,
- Classical systems engineering and technical direction efforts that result in management controls for development, and
- Actual implementation engineering efforts that lead to realization of the new system.

It is also very important to determine whether some process or organizational restructuring, or reengineering, is needed to accommodate a new system, or whether a new system should be planned such as to accommodate existing organizational structure and functions. The effort associated with building a system must, if it is to be efficient and effective, consider not only the needs of the system using organization for the new system, but also the capabilities of the teams and organizations that, collectively, are responsible for engineering of the system. In effect, all of these ingredients should be integrated. These create the requirements for systems engineering and management efforts that are enterprise and knowledge management focused. These are major issues in systems and implementation of a system. Each of these views or perspectives needs to be accommodated when determining the architecture of a system to be engineered if the resulting system is to be trustworthy and, ultimately, used to fulfill its intended purpose.

SYSTEMS CONFIGURATION

Following the identification of an appropriate systems engineering process life cycle to use in engineering the system, configuration management plans are identified. This involves defining a specific development process for the set of life-cycle tasks at hand. Metrics of cost analysis or cost estimation are needed to enable this to be done effectively. Today, virtually every new system is software and information technology intensive. Thus, there are major roles for cost and economic estimation for software and information technology based systems. They also include effectiveness analysis or estimation of software productivity indices using various metrics. This couples the notion of development of an information technology or software (or other) product into notions concerning the process needs associated with developing this product. These metrics and indices form a part of a systems management approach for process, and ultimately, product improvement.

Critical aspects of a problem are often a function of how the components interact. Simple aggregation of individual aspects of a problem is intuitively appealing but often wrong. The whole is often not simply the sum of its parts. This does not suggest at all that scientific analysis, in which an issue is disaggregated into a number of component issues and understanding sought of the individual issues, is in any way improper. The formal approach involves three basic efforts:

1. Disaggregation or decomposition of a large issue into smaller, more easily understandable parts
2. Analysis of the resulting large number of individual issues
3. Aggregation of the results to attempt to find a solution to the major issue

APPROACH TO SYSTEMS MATURITY

Problems and issues arise in the course of systems engineering a product or service. Three organizational approaches may be useful in addressing them:

Reactive. Examines a potential issue only after it has developed into a real problem.

Interactive. Examines issues while they are evolving, diagnoses and corrects problems upon detection.

Proactive. Predicts the potential for debilitating issues/problems. Synthesizes a life-cycle process to minimize their likelihood.

While proactive and interactive efforts are associated with greater capability and process maturity, reactive efforts are still generally needed. Of course, another option is the "inactive" approach, favored by an organization that worries little about issues that may well become serious problems if not promptly dealt with.

COMMUNICATIONS AND COMPUTING

The role of communications and computing is ubiquitous in systems engineering and management. Major growth in power of computing and communicating and associated networking is quite fundamental and has changed relationships among people, organizations, and technology. These capabilities allow us to study much more complex issues than was formerly possible. They provide a foundation for dramatic increases in learning and associated increases in both individual and organizational effectiveness. In large part, this is because of the networking capability that enables enhanced coordination and communications among humans in organizations. It is also because of the vastly increased potential availability of knowledge to support individuals and organizations in their efforts.

The need for integration of information technology issues with organizational issues has led to the creation of a field of study, the objectives of which generally include:

- Capturing human information and knowledge needs in the form of system requirements and specifications
- Developing and deploying systems that satisfy these requirements
- Supporting the role of cross-functional teams in work
- Overcoming behavioral and social impediments to the introduction of information technology systems in organizations
- Enhancing human communication and coordination for effective and efficient workflow through knowledge management
- Encouraging human and organizational evolution and adaptation in coping with these challenges.

SPECIFIC APPLICATIONS

There are many specific applications as well as specific issues where systems engineering and management is germane. For example, when a system is to be developed and deployed using concurrent engineering techniques, as opposed to strictly sequential techniques, systems engineering and management is particularly important. Indeed, it may reveal that concurrent engineering is inappropriate to a particular project. Conversely, it may enable successful concurrent engineering of a system that might otherwise fail to meet its expected performance and deadline requirements. See Sage and Rouse (1999). Andrews and Leventhal (1993), Kronlof (1993), and Fiksel (1993).

Another issue on which systems engineering and management has a strong influence is that of physically partitioning a system. The factors bearing on this problem can perhaps best be expressed by these questions:

1. Where ought it to be partitioned from a design standpoint?
2. Where ought it to be partitioned from a development and prototyping standpoint?
3. Where ought it to be partitioned from a fabrication standpoint?
4. Where ought it to be partitioned from a deployment and maintainability standpoint?

The final decision will be a compromise, in which one or more of the factors may have a prevailing influence.

For elaborations and in-depth treatment of topics covered in this chapter, the reader is referred to the "Handbook of Systems Engineering and Management" (Sage and Rouse, 1999).

BIBLIOGRAPHY

Andrews, C. C., and N. S. Leventhal, "Fusion—Integrating IE, CASE, and JAD: A Handbook for Reengineering the Systems Organization," Prentice Hall, 1993.

Andriole, S. J. "Managing Systems Requirements: Methods, Tools and Cases," McGraw-Hill, 1996.

Buede, D. M., "The Engineering Design of Systems," Wiley, 2000.

Davis, G. B., "Strategies for information requirements determination," *IBM Syst. J.*, Vol. 21, No. 1, 1982.

Department of Defense, Directive Number 5000.1–The Defense Acquisition System, Instruction Number 5000.1 and 5000.2–Operation of the Defense Acquisition System, April, 2002. Available at http://www.acq.osd.mil/ar/#5000.

Fiksel, J., "Computer-aided requirements management for environmental excellence," *Proc. Ann. Mtg. Nat. Council Syst. Eng.*, 1993.

Kronlof, K. (ed.), "Method Integration: Concepts and Case Studies," Wiley, 1993.

Maier M. W., and E. Rechtin, "The Art of Systems Architecting," 2nd ed., CRC Press, 2001.

Pascale, R. T., "Surfing the edge of chaos," *Sloan Management Rev.*, Vol. 40, No. 3, Spring 1999.

Pascale, R. T., M. Millemann, and L. Grioja, "Surfing the Edge of Chaos: The Laws of Nature and the New Laws of Business," Crown Business, 2000.

Porter, M., "Strategy and the Internet," *Harvard Business Rev.*, Vol. 79, No. 3, March 2001.

Robertson, S., and J. Robertson, "Mastering the Requirements Process," Addison Wesley, 1999.

Rouse, W. B., "Don't Jump to Solutions: Thirteen Delusions That Undermine Strategic Thinking," Jossey-Bass, 1998.

Rouse, W. B., "Essential Challenges of Strategic Management," Wiley, 2001.

Sage, A. P., "Systems Engineering," Wiley, 1992.

Sage, A. P., "Systems Management for Information Technology and Software Engineering," Wiley, 1995.

Sage, A. P., "Towards systems ecology," *IEEE Computer*, Vol. 31, No. 2, February 1998.

Sage, A. P., and C. L. Lynch, "Systems integration and architecting: An overview of principles, practices, and perspectives," *Syst. Eng.*, Vol. 1, No. 3, 1998.

Sage, A. P., and W. B. Rouse (eds.), "Handbook of Systems Engineering and Management," Wiley, 1999.

Sage, A. P., and C. D. Cuppan, "On the systems engineering and management of systems of systems and federations of systems," *Information, Knowledge, and Systems Management*, Vol. 2, No. 4, 2001.

Sommerville, I., and P. Sawyer, "Requirements Engineering: A Good Practices Guide," Wiley, 1997.

Thayer, R. H., and M. Dorfman (eds.), "Software Requirements Engineering," IEEE Computer Society Press, 1997.

Young, R. H., "Effectiveness Requirements Practices," Addison Wesley, 2001.