Regular Paper

Case Studies of Systems Engineering and Management in Systems Acquisition

George Friedman[1] and Andrew P. Sage*, [2]

[1]*Department of Industrial and Systems Engineering, University of Southern California, Los Angeles, CA*

[2]*Department of Systems Engineering and Operations Research, George Mason University, Fairfax VA, 22030-4444*

Received 6 April 2003; Accepted 22 September 2003, after one or more revisions
DOI 10.1002/sys.10057

ABSTRACT

This paper discusses the role of case studies in systems engineering and systems management, especially case studies that involve systems acquisition. We first provide a brief overview of case studies, including some of the analysis techniques useful for the conduct of case studies. Next, we discuss a two-dimensional framework for systems engineering and management case studies. The framework is in the form of a 9 row by 3 column matrix. We present a number of vignettes of case studies, at least one for each of the 27 cellular entries in this matrix. The hope is that this will be a stimulus and precursor of additional systems engineering and management case study efforts, both in terms of appropriate frameworks for these and in the actual conduct of case study research. © 2003 Wiley Periodicals, Inc. Syst Eng 7: 84–97, 2004

Key words: case studies; case framework; systems engineering taxonomy

1. INTRODUCTION

Systems engineering concepts—including methods, processes and systems management to enable trustworthy implementation of processes and methods to ensure quality products and services—are very important for success today. The material to follow was brought together with two major considerations:

*Author to whom all correspondence should be addressed (e-mail: asage@gmu.edu).

Systems Engineering, Vol. 7, No. 1, 2004

1. In the teaching of systems engineering, case studies are potentially very instructive in that they relate aspects of the real world to the student through exposure to realities in the world of professional practice. Frequently, the bad examples are even more valuable than the good ones, since they emphasize the penalties for not following the proper concepts and processes of systems engineering.
2. In teaching systems engineering, there has previously been little distinction between the duties and responsibilities of public sector, government, and private sector, industry, activities. There is also a third sector, comprised of nonprofit organizations, which may often need to be distinguished from for-profit private sector organizations.

Understanding of all three roles is needed. Many, if not most, of the textbooks available, with the exception of strictly government publications, stress the private sector aspects of systems engineering. However, in future curricula, the distinction should be made clearer and the government aspects deserve much more emphasis, as does the need for and role of joint government-contractor effort. Of course, the private sector industrial aspects should not be ignored; government and nonprofit sector professionals should be expected to become smart buyers of private sector contractor activities and appreciate their many problems and challenges.

It is usually important to define a concept, even those that seem to be familiar. For our purposes (Yin, 2003a), a case study is empirical inquiry that:

- Investigates a contemporary phenomenon within its real-life context, especially when
- Boundaries between phenomenon and context are not clearly evident, and in which
- Multiple sources of evidence are generally used.

Many other authors infer similar definitions. Case studies have has a long history in many disciplines, notably, sociology, psychology, medicine, history and political science, and business. The history of case studies shows periods of intense use and also periods of considerable disuse.

2. AN OVERVIEW OF CASE STUDY RESEARCH

Case studies have had a long history, and formal ones have been conducted for approximately a century. Informal case studies have, of course, existed for a much longer period of time. In the social sciences, case studies became less popular a half century ago as the social science disciplines became more "scientific" and quantitative. Case studies were seen as suffering from fundamental problems of external and internal validity, and sometimes even from construct validity and reliability (Yin, 2003a):

- **Internal Validity**—Were the findings actually justified by the research, or were there problems of researcher bias? Has the case study researcher demonstrated a causal relationship between factors by showing that other plausible factors could not equally well or perhaps better explain the observed relationships?
- **External Validity**—Could the research findings be generalized?
- **Construct Validity**—Do the measures used in the case study make the concepts involved operational? In other words have we used multiple evidence sources, have we sufficiently established chains of evidence and have those providing evidence to the case study been allowed to review the case report before finalizing it.
- **Reliability**—To what extent would other researchers who are studying the same case in exactly the same way arrive at equivalent conclusions?

Each of these is particularly important in case study research. For example, relative to researcher bias, it is particularly important to avoid even the appearance of researcher bias. At a minimum, this suggests strongly that the case study research team members should not all be stakeholders of the programs being reviewed.

A recent revival of interest in case study research has occurred, especially in evaluation research in many areas such as enterprise management. This has led to the recognition that case study research can fill important needs. In particular, it is often insufficient to know that X can cause Y. We also need to know the **how** and the **why** X causes Y, and for **what** specific X and Y. Case study research can potentially answer these interrogatives and thereby help us become more contextually aware. Modern case study research is able to deal with issues of internal and external validity if the case study is well defined, developed, and deployed, with appropriate formulation, analysis and assessment, and interpretation across these three phases of the case study effort. These are the three basic phases and steps in a two-dimensional framework for systems engineering (Sage, 1992, 1995; Sage and Rouse, 1999), and it is fully appropriate and, we believe, desirable that a systems engineering process be used to identify a suitable framework for case study research in systems engineering.

A prototypical characteristic of case studies is that they support a holistic understanding and interpretation of the systems of action, or interrelated activities engaged in by the participants or actors in the case situation subject to study. Selecting cases must be accomplished in such a way as to maximize what can be learned in the period of time available for the study. Case studies often are selective, focusing on one or two issues that are believed most fundamental to understanding the issue under consideration. Case studies should be multiperspective in nature, where the case study researcher considers not only perspectives of the actors, but also of the interaction between them. What is "most fundamental and important" in a given case depends greatly upon the viewpoint or perspective of the case study participants and observers.

Yin (2003a) has summarized the major strengths and limitations inherent in case study designs. First, case studies are useful for addressing questions regarding **how** and **why** phenomena behave the way they behave. These case studies more often lead to hypotheses about behavior rather than being useful for validating general claims about behavior. Second, case study research often reveals a rich detail of information that highlights the critical contingencies that exist among the variables in the case study. Finally, the case study method is especially useful for exploration of topics when there is not a strong theory to which one can appeal. Even when there is a strong normative theory, case studies can often lead to valuable insights, and potentially even to revision of the normative theories.

Case studies can involve either single or multiple case designs. Single case designs are generally used to confirm or challenge a theory, or to represent a unique or extreme case of behavior. Single case designs require very careful investigation in order to avoid misrepresentation and to allow maximum access by the researcher to relevant evidence. Multiple case studies follow a replication logic in which each individual case study consists of a "whole" study in which relevant information is gathered from various sources and conclusions are drawn based on this information.

Yin (2003a) has identified six sources of evidence in case studies.

1. **Documentation**, which could be letters, memoranda, agendas, administrative documents, newspaper articles, or any information believed relevant to the investigation. Documents are communications between actors in the study and often serve to corroborate or refute evidence from other sources. Documents are used to make inferences about events. They can result in false clues, especially in the hands of inexperienced case study researchers, and this is one criticism of case study research, leading to the observation that a case study researcher should be a "critical observer" to avoid being misled.
2. **Archival Records** can be organizational documents, official lists of names, survey data, and other such records. A case study researcher must be careful in evaluating the accuracy of records before using them. Even if the records are quantitative, they might still not be accurate.
3. **Interviews** represent a most important source of case study information. There are several forms of interviews: open ended, focused, and structured or survey.
4. **Direct Observations** are obtained by such activities as field visits to a case study site, and other efforts to obtain data that range from the formal to the casual. It is thereby possible to cover events and contexts in real time. The disadvantages are that it may be a time consuming and costly activity, discriminatory results may be obtained unless the observations are broadly based, and direct observation may cause events to proceed differently than the otherwise would.
5. **Participant Observations** have many of the same strengths and weaknesses as direct observations. They may lead to insightful observation of interpersonal behavior but are subject to bias if the case study researcher manipulates events in some ways.
6. **Physical Artifact Observations** may provide much insight into cultural features and technical operations. The weaknesses of this source of evidence are selectivity and availability of observations.

The current rebirth of interest in case study research highlights the central role of theory and the importance of having a framework within which to address case study research. We address this latter subject in our next section of this paper. Theory should guide design and analysis of case studies, and the results of a case study should assist in the development of theory. Therefore, identification of boundaries for a case study should be strongly influenced by the theoretical perspective taken by the case study researcher to the purpose and subject matter for the study.

Yin (2003a) recommends three general analytical strategies for establishing a case study:

1. Relying on theoretical propositions. The first and generally more preferred strategy is to follow the theoretical propositions that led to the case study

in the first place, at least initially. These propositions necessarily shape data collection and allow us to focus attention on certain data and to ignore other data. They ideally help us to organize a complete, or entire, case study and to identify alternative potential explanations that should be examined. The "how" and "why" are especially relevant here.

2. Identify thoughts about contending explanations. This allows us to examine and accept or reject alternative premises concerning case behavior.
3. Developing a case description. In this often-used approach, the researcher writes a narrative that describes the case and its history.

There are at least five specific techniques for analyzing case studies: pattern matching, explanation building, logic models, time-series analysis, and cross-case synthesis. These approaches are not mutually exclusive and use of multiple approaches will generally be very useful, whenever it is possible to do so.

1. **Pattern Matching.** Pattern matching contrasts and compares an empirically observed pattern with one that is predicted. This might be used, for example, to compare our case study to one found in the literature. Here a researcher develops a prediction about patterns expected to be seen in collected data. The associated analysis attempts to determine the pattern that best corresponds to observed data. The major difficulty with this approach, assuming, of course, that it is possible to collect the relevant data, is to decide upon how close is the "fit" of the data to a predicted pattern. Much researcher interpretation and judgment may be needed.
2. **Explanation Building.** Here, the researcher analyses data by building up an assumed explanation about the case. The purpose of this is to analyze the case study data by establishing an explanation about the case. This is usually accomplished by linking events and issues causally. This approach goes beyond pattern matching in that specification of the nature of the links, usually causal, between elements that make up the pattern is a hoped for result. The researcher then tests the evidence for the relationships.
3. **Toulmin Logic Models.** Toulmin argumentation (Toulmin, Rieke, and Janik, 1984) is based on the notion that the statements that make up an explanation fall into six basic types: data or grounds, inference warrants, backing, modal qualifier, rebuttals or reservations, and resulting claims or conclusions. Toulmin logic, the structure of which is suggested in Figure 1, and associated analysis involves breaking case study material down into these categories and then linking them together such as to make structure, function, and purpose of cases and their explanation more explicit. Janssen and Sage (2000) discuss the use of Toulmin logic in a knowledge management and conflict resolution support system that may well have case studies as suitable inputs.
4. **Time Series Analysis** is used in an attempt to match observable patterns in data to an underlying model proposed for explanatory purposes. There are three major situations in time series analysis. The first and simplest situation is when there is a single endogenous or dependent variable that may or may not be affected by unusual events. The second situation extends the first by inclusion of input or causal variables that may have a role in the prediction or modeling of the assumed single dependent variable. The third situation provides for simultaneous modeling of multiple dependent time-series. Very interesting situations occur when the data is such as to suggest chaotic behavior patterns and the potential need to involve complex adaptive system properties as case explanations.
5. **Cross-Case Analysis and Synthesis** is particularly useful when case studies are themselves comprised of more than a single case.

The works of Yin (2003a) and Stake (1995) concerning case study research are especially recommended. Yin (2003b) provides several illustrative examples of case studies in the social sciences. Tien (1999) provides a useful discussion of the evaluation of systems and Adel-

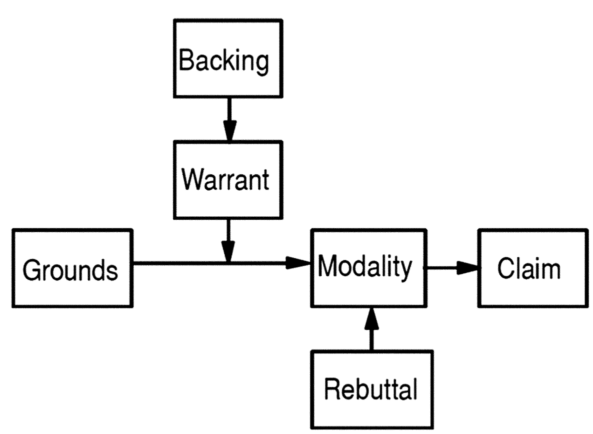


Figure 1. Toulmin logic structure.

man (1992) discusses evaluation in decision support and expert systems..

The Harvard Business School is among the many business schools that emphasize case studies in their educational program (Barnes, Christensen, and Hansen, 1994). Shapiro (1988) suggests that the total case process in an academic study environment, based upon the premise that management is primarily a skill-based activity rather than one based on a collection of tools and techniques, is comprised of four steps:

1. Individual analysis and preparation of and for the case;
2. Optional informal small group discussions;
3. Classroom discussions; and
4. End of class generalization relative to the learning accomplished.

There are two generic types of case studies, the first of which is retrospective analysis, which can be further categorized as historical descriptions, research events studies, and matched comparisons. The second type represents a combination of retrospective assessment with prospective approaches and use of such analysis approaches as aggregate statistics, peer review, and the laws of bibliometrics. Each approach is useful and development of newer case study techniques has not resulted in the obsolescence and retirement of the earlier approaches.

In this section, we have discussed the completeness and rigor of case studies descriptions. However, in applying case studies to systems engineering research and education, there is a symmetrical issue: that of potential overdescription of the elements and environment associated with crucial events of the systems engineering process. It will often be the minimum essential set of measurements and data that most clearly illuminates the workings of newly investigated phenomena. Excessive descriptions and elaborations may tend to cloud the inner workings of the judgmental process needed for effective case study research.

3. A FRAMEWORK FOR SYSTEMS ENGINEERING CASE STUDY RESEARCH

To support the objectives identified in Section 1, we identify a two dimensional framework for systems engineering case study research. Figure 2 represents the

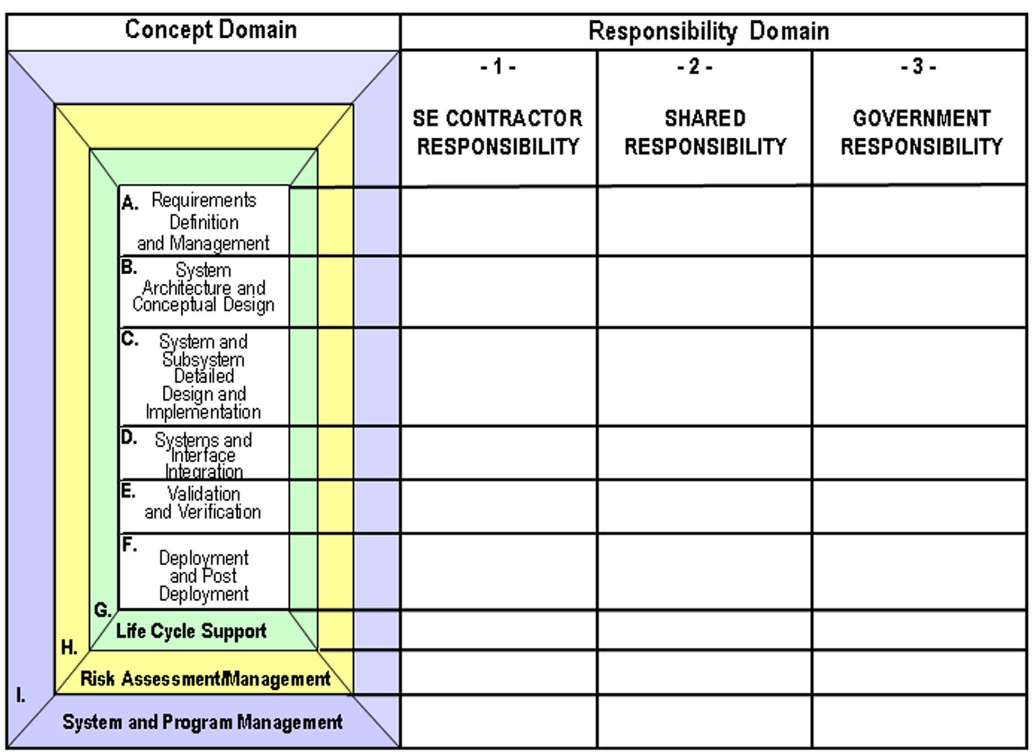


Figure 2. A framework of key systems engineering concepts and responsibilities. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

initially suggested framework. It is constructed as a 9 $\times$ 3 matrix with nine general SE concept areas as rows, A–I and three columns, 1–3, depicting the domain of the principal activity: primarily contractor, shared between contractor and government, and primarily government. This is intended to provide the general structure of a framework for systems engineering concepts and their illustration through related case studies. This two dimensional framework is central to the case study efforts suggested here.

Six of the nine concept domain areas in Figure 1 represent phases in the systems engineering lifecycle:

A. Requirements Definition and Management
B. Systems Architecting and Conceptual Design
C. Detailed System and Subsystem Design and Implementation
D. Systems and Interface Integratio
E. Validation and Verification
F. System Deployment and Post Deployment

Three of the nine concept areas represent necessary process and systems management support:

G. Life Cycle Support
H. Risk Management
I. System and Program Management

While other concepts could be identified, we believe that these nine are the most relevant to systems engineering in that they cover the essential life cycle processes in systems acquisition, and the systems management support of these. Most other concept areas that were identified appear to be subsets of one of these.

As noted, the three columns of this two-dimensional concept framework represents the responsibility domain and perspectives of government, and contractor, and shared responsibilities of government and contractor. This could be expanded much further. For example, each of these groups will be concerned with being responsive to abstractions or contextual awareness interrogatives of why, how, what, who, when, and where. We might also disaggregate the contractor responsibilities into nonprofit contractor responsibilities and for-profit contractor responsibilities. Our belief, at this point, is that this additional complexity is not warranted as the notion of "contractor" and "government" needs natural modification when there are multiple contractors and multiple government agencies involved. This leads naturally into consideration of "system family" (Sage, 2004) considerations needed to engineer a system of systems, a family of systems, a federation of systems, or a coalition of systems. These expressions are in common contemporary use to describe a system (family, federation, coalition) that is comprised of a set of systems each of which are fully formed and functional, and which possesses such properties as managerial independence, geographic and/or temporal distribution, emergence, evolution, emergence, and adaptation.

Thus, while the three column structure of "Contractor Responsibilities," "Shared Responsibilities," and "Government Responsibilities" is very appropriate; the specific nature of these will vary greatly depending upon whether the system being engineered is a monolithic (stand alone) system or a System of Systems (or Family or Federation of Systems). In a great many contemporary efforts that involve the engineering of a system, we are really dealing today with engineering a System (or Family or Federation or Coalition) of Systems. When the government (or a private developer, for that matter) is acquiring a stand-alone system, it may often contract directly with a contractor that offers "maximum value added" to the government. But when the government is acquiring a System of Systems, there are at least three rather different approaches that might be chosen (Carlock and Fenton, 2001):

a. Option (a) is that of letting a prime SOS engineering contract to a single systems engineering (and integration) firm. This prime contractor is then responsible for the total systems engineering effort to bring about the SOS. There are a number of pitfalls here. For example, the government may feel, after contract award, that the prime contractor is selecting dismal quality firms as subcontractors for various systems that comprise the SOS.
b. Option (b) is that where the government assigns separate contracts for the engineering of each of the component systems in the SOS and then the government acts as the integrator and SOS program manager.
c. Option (c) is that where the government assigns separate contracts for the engineering of each of the component systems in the SOS and then also contracts with another firm that acts as an independent SOS integrator and SOS program manager.

Thus, delineation of responsibilities in the three columns may be expected to vary considerably between acquisition of a stand alone system and acquisition of an SOS and the program policy and management options (a, b, c) chosen. What is contractor responsibility, shared responsibility, and government responsibility depends on which of these three options is chosen. What this suggests is the great importance of considering the three basic responsibility domains in the systems engineering concept framework suggested here; and developing familiarity with the roles of the two major players, government, and contractor, as well as shared roles, in the contemporary engineering of systems.

4. CASE STUDIES ILLUSTRATING SYSTEMS ENGINEERING CONCEPTS

It would be highly desirable to write and present here one or two case studies, with each illustrating the wealth of information and knowledge that would fill all of the cells in the matrix of Figure 2. While ultimately this is possible, we present here, for conceptual purposes, simpler studies or vignettes illustrating cell-wise facets of good and bad systems engineering practice. We can obtain a synthesis of lessons learned from each of these cell wise representations and do present this here. From experiences gained in doing this and in understanding these, one is in a better position to attempt larger more holistic efforts involving complete cases.

What follows is a list comprising at least one systems engineering concept within each of the 27 elements of the case study concept framework matrix. These concepts are written in the spirit of being requirements for good systems engineering; thus they are written as "shall statements." Each of the concepts is associated with at least one illustration or vignette of a situation, in this instance drawn from a multitude of different cases, which would in an actual application be characteristic of a specific case history. Essentially, then, the trajectory of these composite situations across the rows and columns of the matrix would form the skeleton of a full case history. These are intended for augmentation and revision as experience with this effort is gained. Some of these situations are decades old, and the current focus on evolutionary acquisition and system families should provide even more current and broad scope illustrations. Perhaps this list might ultimately become a set of useful Heuristics for a Body of Knowledge of Systems Engineering Concepts and Practices.

4.1. Requirements Management (A)

4.1.1. Contractor (A1)

SE concept: Requirements shall flow down in a coherent and traceable manner from the top level to all lower levels of the system being engineered.

Case study: The flow-down of requirements was often slowed by the lack of early lifecycle systems modeling capabilities on the part of the contractor, so there were many paragraphs with TBDs ("to be determined" elements instead of quantitative specifications). As the TBDs stubbornly refused to be quantified, program management "incentivized" their determination by coupling pay raises with how many TBDs were eliminated. The response by the engineers was to "temporarily" eliminate these TBDs by removing the paragraphs in which they were imbedded, thereby creating "silent specs" with the hope to fill them in when the analysis was complete. Time passed, and the engineers were transferred to other programs without warning their successors of the silent specs. Some of these specs should have been passed on to a key subcontractor but were not. The result of this was that the systems-level test failed, the prime contractor lost schedule and over $100M in profit. The program was eventually cancelled.

4.1.2. Shared (A2)

SE concept: Customer and contractor shall share with one another their knowledge of the state of technical maturity relative to the new, unprecedented systems being engineered.

Case study: A government customer desired to procure two new navigational systems: one stellar inertial and the other pure inertial. The most capable contractor in the pure inertial field decided to expand the portion of the market sector under their purview and substantially underbid the proposal for the stellar inertial system. This contractor won; so the stellar inertial contractor counter attacked and won the pure inertial contract. Both contractors, unfamiliar with the new (to them) technologies to be used in the product to be engineered, did poorly both technically and financially. They suffered—as did the customer—from the procurement policy that they could not be negative about the technical aspects of contractors who had submitted reputable proposals in the past.

4.1.3. Government (A3)

SE concept: The government shall integrate the needs of its user organizations with the management activities of its developmental organizations. Often, the user and development organizations are separated by gulfs of culture and language. The development organization sponsors new technology and contracts out programs; the user organization—the ultimate client—operates the new systems, trains the personnel, fights battles, and should be intimately involved in writing the requirements for new programs.

Case study: An advanced state of the art navigational system was specified for an attack aircraft. Several years after the program start, the operational arm of the government added more requirements in the crew station interface area. They claimed that they had insufficient voice in the construction of the original specification. Results: substantial financial and schedule impact.

4.2. Systems Architecture (B)

4.2.1. Contractor (B1)

SE concept: The systems baseline architecture of complex programs shall be established early in every program and shall involve all dimensions of technical issues, as well as such enterprise architecture issues as customer needs and satisfaction, political pressures and continuity of funding. A properly executed systems architecture activity provides benefits of effectiveness far in excess of its costs. In dealing with system families, it is to be anticipated that the systems architecture will evolve and emerge from the baseline architecture as the system is engineered.

Case study: If the allocation of functions to equipment and crew is performed without deep understanding of human/machine interactions and their respective strengths and limitations, the overall systems performance will suffer. In a particularly stressful case of a combat aircraft required to perform aggressive maneuvers while attempting to acquire and classify a low contrast target, the task of aircraft controls was delegated to the pilot, and the task of classification was delegated to an automatic pattern recognition program. The effectiveness of the system would have been substantially improved if the maneuvers were under automatic control (supervised by the pilot) and the task of pattern recognition were given to the far superior cognitive capability of the crew member.

4.2.2. Shared (B2)

SE concept: The systems architecture should be established early for the reasons stated in 4.2.1, and the best judgment of both government and contractor shall be employed across all the key issues, including the choice of employing newly developed or legacy systems.

Case study result: The existence of commercial off-the-shelf (COTS) hardware and software appears to offer a very attractive alternative to the expensive and lengthy developments required to produce specially designed hardware and software. However, unless the total system architecture implications are well understood, COTS can be extremely disappointing regarding its incomplete characterization, testing and maintenance, and logistic aspects. Frequently, many upgrade generations of software occur during a single government development cycle, and the producer of the software—who view the government as a small customer compared to their commercial markets—is unwilling to continue supporting the old versions. In many cases, the government pendulum of employing excessive specifications swinging to relying entirely on commercial best practices has injured the government and industry's ability to manage the integrity of new designs.

Additional case study result: The effectiveness of guided missiles could be greatly improved by integrating the functions of guidance and fusing; that is, the information regarding the terminal kinematics gathered by the guidance system can be transferred to the fusing system, substantially improving the probability of kill. However, these two systems are generally kept separate by traditional procurement policies and divisions.

4.2.3. Government (B3)

SE concept: A total systems architecture shall be employed by the government for the reasons stated in 4.2.1 in order to provide a sound basis of effectiveness across the broadest spectrum of contractors and operations.

Case study: Joint developments in aircraft and missiles have been undertaken across the services in order to attain envisioned economies. In most cases, these have been met with several difficulties in extended schedules, higher costs, compromises in operational effectiveness, and awkward integration within diverse logistical support systems. In the early stages of program planning and systems architecture, all these potential problems were lost in the glare of the hoped-for savings. In some cases, the joint programs were cancelled.

4.3. System/Subsystem Design (C)

4.3.1. Contractor (C1)

SE concept: System design shall proceed in a logical and orderly manner through a process of functional decomposition and design traceability that originates with the system functional architecture and ultimately results in design specifications for the system to be engineered.

Case study result: If the functional decomposition is excessively rigorous, potential technical synergies may be missed. For example, if the decomposition is strictly treelike, then the merging of radar and inertial navigation would never have the opportunity to develop into synthetic array radars and inertial components for flight control and for navigation would not have the opportunity for integration.

4.3.2. Shared (C2)

SE concept: Government customers and contractors shall have a contractually feasible sharing of systems design responsibility.

Case study result: On a defense suppression missile, the government customer engineered the overall systems architecture and performed sole source selection of the subsystems. The top level architecture appeared logical, but second-order interface problems developed during systems integration and test. Had the total systems responsibility been given to a single prime contractor, these problems would have been very unlikely. Result: eventual program cancellation with criticism of the contractor.

Additional case study result: A contractor proposed a technology which was proprietary. Despite the fact that this technology substantially improved system effectiveness, the government gave this contractor a negative evaluation because the intellectual property (IP) protection would limit competitive reprocurement later in the program. The result was that the program went ahead with an inferior technology and diminished effectiveness—injuring the government—while the contractor was discouraged from performing internal research and development (IR&D), thereby injuring both contractor and government capabilities.

4.3.3. Government (C3)

SE concept: The customer shall share high level measures of effectiveness with the contractor, thereby ensuring that the proposals selected for funding are those which are most responsive to all stakeholders, especially the operational organizations.

Case study result: At a meeting of the contractors' chief engineers and the government engineers, the concept of such overarching measures of effectiveness was suggested by a contractor. The government responded that their engineers wanted to reserve the right to alter their selection criteria after they had reviewed all the proposal submissions. The result: the government lost an opportunity to communicate its fundamental needs to the contractors who really are interested in serving the customers better, and when the final decisions are made, the contractor feels that the government acted arbitrarily and perhaps even capriciously. In short, there was a diminishing trust at both government and contractor ends.

4.4. Systems Integration and Interfaces (D)

4.4.1. Contractor (D1)

SE concept: The contractor shall assure that systems integration and interfaces at each of the subsystems and components levels supports total system functionality throughout its life cycle.

Case study result: In general, component and subsystem testing have proved to be insufficient in predicting problems that are likely to occur in the many stages of systems integration. It is necessary to insure that systems integration issues are dealt with throughout the lifecycle, especially in architecting and preliminary system design

4.4.2. Shared (D2)

SE concept: The contractor and government shall assure that all systems are integrated within themselves as well as interfaced with other existing operational equipment and systems.

Case study result: Joint programs across two or more of the services have suffered from differences in operational use and doctrine, as well as from very different support and logistics systems. In the planning and proposal stages, neither the government nor the contractor often, or at all, considers the multitudes of issues associated with satisfying diverse customers with the same—or small variations of the same—system.

4.4.3. Government (D3)

SE concept: The government shall assure that all its operational systems—in development, in operation, or in planning—are compatible and mutually supportive in a broad "system of systems" and "federation of systems" context.

Case study result: Most DOD agencies have existing, evolving, and planned intelligence, command and control systems for war fighting and antiterrorism. The difficulty of obtaining reliable intelligence is made far more difficult by the interagency rivalries and delays in getting the information to the fighting forces on time.

4.5. Verification and Validation (E)

4.5.1. Contractor (E1)

SE concept: Every requirement shall have a test and every test shall have a requirement which requires validation and verification. The criteria for determining test success and failure shall be established early in the program, as shall verification and validation measures.

Case study result: Near the end of the development phase of a complex and unprecedented electronic countermeasures program, an exhaustive test was conducted. The contractor was elated by the results, feeling that the grade earned for performance was at least an A or A–. The government gave the contractor a D+, only then sharing with the contractor the specific test criteria that had been used to evaluate performance.

Another case study result: Although test planning should occur almost simultaneously with requirements development, often the test planning is delayed by years and test people are frequently not even involved with the earlier life cycle phases of architecting and preliminary systems design.

4.5.2. Shared (E2)

SE concept: Government facilities are often the most expensive and their use in the verification and validation (V&V) process shall be shared with the contractors. Test criteria shall be shared early.

Case study result: The coordination of scarce government testing facilities is crucial to the success of all programs. Often, program slippages occur which directly impact the scheduling of these resources negatively affecting schedules and budgets.

4.5.3. Government (E3)

SE concept: The government shall be the final word on the confidence levels derived from its testing during development, operational test and evaluation, and actual deployment and operational use. These require judgments at the highest levels, because not all operational conditions can be adequately tested and actual operations are frequently far different from what the planners initially imagined and intended.

Case study result: Most anti-terrorism and third world military technology which are presently fielded were developed to solve military goals of the cold war, where the Soviet Union was the dominant adversary. Today, the focus has shifted to combat environments such as the Persian Gulf and antiterrorism—with substantially different goals, scenarios and testing challenges.

4.6. Deployment and Post-Deployment (F)

4.6.1. Contractor (F1)

SE concept: As systems undergo operational test and evaluation, the contractor shall maintain the appropriate engineering and testing organizational capabilities to support gathering, analyzing, and recommending possible changes in the system design or support through reengineering.

Case study result: Once the system is delivered to the customer, contractors sometimes are tempted to transfer their most capable engineering and test personnel to new programs and essentially forget the needs of properly supporting the system throughout its deployment lifecycle and planning for reengineering.

4.6.2. Shared (F2)

SE concept: Both government and contractor shall work cooperatively to conduct an effective operational test and evaluation (Opeval) and to be open to feeding information back to the original management organizations to consider design changes or subsequent modifications through reengineering.

Case study result: Coordination is often difficult and the program review process is not as intensely focused as those during the development phase. The Opeval environment is sometimes not a realistic representation of actual test conditions. In many cases, the real Opeval does not occur until actual combat situations emerge.

4.6.3. Government (F3)

SE concept: The government shall assure that a properly funded Opeval occurs, one which draws upon both contractor and government resources, and that all data gathered in the tests be evaluated for potential recommendation for system modification, redesign, or reengineering. Preplanned product improvement policies shall be encouraged.

Case study result: In many cases, as the deployment becomes fully operational and deficiencies are discovered, the choice of whether to perform design modifications or to start a new development is not as illuminated as other debates in the procurement world. For decades, there has been a rhythm in development that every generation will have a next generation and this precept has often blinded both the development and user communities to the possibilities of extending the life of existing and proven systems in a "pre-planned product improvement" manner.

4.7. Life Cycle Support (G)

4.7.1. Contractor (G1)

SE concept: All design activities shall be performed from the viewpoint of the entire life cycle rather than the early prototype development phase.

Case study result: Many preliminary design teams travel from program to program writing proposals and—if the company is a winner—sometimes working on those programs for as much as a year, until they depart for the next program. In this flurry of activity, the greatest emphasis is given to the "dramatic" vector of key performance parameters, while the many other factors required for an effective life cycle are ignored or postponed.

Another case study result: The manufacturing world has often complained that producibility has not received proper attention from the design community. Yet there are many instances where a design that optimizes certain aspects of producibility—such as minimum time to assemble—is very poorly designed for maintainability. Furthermore, it is rare that enough attention is ever paid to training and the writing of readable and useful user manuals.

4.7.2. Shared (G2)

SE concept: A balanced blend of all methods, measurements, technologies, and processes shall be employed in support of an effective life cycle.

Case study result: Over the years, feeling that their operational readiness and capability have suffered because of the overemphasis on performance, operational commanders have exerted considerable pressures to increase the priority of reliability, availability, and maintainability (RAM) and other aspects of the integrated logistics support (ILS) family. Additional management initiatives have entered the scene, adding to the complexity of program development with often conflicting requirements, such as the "maximization of reliability, availability, and maintainability."

4.7.3. Government (G3)

SE concept: Funding support, throughout a program's life cycle, shall be maintained in balance, and procurement of early development programs shall recognize the importance of total life cycle cost control through such approaches as earned value management.

Case study result: The record shows many programs which suffer disruption in funding, thereby adding substantially to the total cost.

Additional case study result: Despite the preaching of the importance of total life cycle cost, it is very rare that a development contract is awarded based on the total life cycle cost rather than the cost of the specific development contract.

4.8. Risk Management (H)

4.8.1. Contractor (H1)

SE concept: At every level of detail, risk shall be identified, prioritized, and mitigated, since the early management of risk requires orders of magnitude fewer resources and causes orders of magnitude less program disruption than allowing the potential risk become an actual crisis. Risk is associated with all three major programmatic dimensions: technical, schedule, and cost.

Case study result: Most programs in the past have had no formal risk management processes; in some cases risk was partially handled via a "program manager's reserve." When the reserve was positive, the manager could afford to be reasonable. As it diminished and became negative, the instance of retorts such as "I only want can-do team players working for me, not the-sky-is-falling worry-warts" and "you're paid to fix problems, not bring them to me; if your job had no problems then I could have filled it with someone with half your salary" become more and more frequent. As risk management becomes more formalized, the resulting qualitative matrices of risk likelihood vs. risk consequence are developed to help prioritize. While better than nothing, they are not as valuable as quantitative assessments of probability and impact, employing decision and risk management theory. Even in the highest tech companies, there is a fear that these quantitative methods are onerous and unduly mathematical—despite the fact that they feel that they are up to the most advanced engineering challenges the world has ever seen.

4.8.2. Shared (H2)

SE concept: It is to the government's benefit as well as the contractor's to identify and mitigate risk early. The government shall insure that risk management is part of the contract systems engineering management plan (SEMP), is properly funded, and is represented in every program review.

Case study result: The government has recently increased the emphasis on risk management, but the approach is still largely qualitative. Some government organizations view risk management as being exclusively in the domain of program management rather than shared between program management and systems engineering. Typically, the analysis required to perform rational decisions regarding risk mitigation comes from engineering organizations, while the decisions to initiate (or terminate) risk mitigation actions come from program management.

4.8.3. Government (H3)

SE concept: Risk management at all program levels shall be an essential and inherent part of all systems management program planning and life cycle activities.

Case study result: A method for risk mitigation—especially for those who are risk averse—is to employ insurance, since the utility functions of a risk neutral insurance company and a risk averse client are both benefited. However, the U.S. Government, with its multi-$trillions of resources should be far more risk neutral than any insurance company; yet it covers the cost of insuring many of its contracted activities. An example is that of space launches in which several billions of dollars were spent on insurance during the past decade and the projection is that at least that amount will be spent during the next decade. Over half of this amount went to the profit and overhead structure of the insurance companies—who are this business to earn the profits they feel they deserve, considering the risks they take. These funds could have been saved if the government had undertaken appropriate self-insurance measures.

4.9. System and Program Management (I)

4.9.1. Contractor (I1)

SE concept: Each and every systems engineering program—even small ones—shall have a Systems Engineering Management Plan (SEMP) which allows the systems engineering process to be tailored to that program. Programs differ by many orders of magnitude in size, complexity, unprecedented technology, and difficulty of interface; the SEMP permits the right amount of effort for each circumstance.

Case study result: A systems engineer wrote a SEMP—because the customer directed it as part of the request for proposals—but as soon as it was written and approved, it was filed and never read. When corporate reviewers showed up to provide program oversight, they suffered paper cuts when opening documents with their fresh, unread pages.

Additional SE concept: Develop a professional cadre of capable systems engineers, including professional development and promotion structures and plans.

Additional case study result: In many companies, systems engineering is performed by an unorganized, disparate, uneven, disgruntled group of beleaguered semiprofessionals drifting like mercenaries from one battlefield to the next, never knowing when the next war will come, but fearing that—no matter how hard they work in the short term—they will be unprepared for it. It would not be too great a conjecture to estimate that a large percentage of the vast majority of unpaid overtime contributed by industry to the national defense posture is from this set of people.

4.9.2. Shared (I2)

SE concept: The role of systems engineering in program development and management shall be recognized and supported.

Case study result: A major program started with a well-funded systems engineering program. A few years into the program, the requirements changed and these changes necessitated major structural revisions and therefore a massive hit on the budget. All activities were reviewed for possible budgetary reductions. The systems engineering budget was hit the hardest because neither the contractor nor the government engineers could clearly define the program risks associated with reducing the SE activity. One of the activities cut was the recording of the traceability of the design rationale from the top requirements to the subsystem specs. Years later, a new program office asked for this rationale and the corporate representative had to convene a special team to reconstruct the entire process.

4.9.3. Government (I3)

SE concept: The government shall establish security levels for programs to protect crucial technologies and special operational capabilities.

Case study result: The government establishes security levels in order to streamline procurement management processes and to minimize the number of people involved with oversight. In some cases, although secrets were kept inside the compartments, competence was kept out. In many cases, early in the development, the so called “ilities people” were also kept out, thereby making life cycle support difficult if not impossible. For those programs which were sufficiently successful to enter limited production, the belated entrance of the –"ilities" folk had a doubly negative impact:

1. The neglected disciplines finally had to be dealt with—much more expensively than if they had been considered properly from the beginning.
2. The experts who were rejected earlier delighted in making the adjustments even more painful than they otherwise might have been, as a “proper and fitting revenge” for being left out in the first place.

5. SUMMARY

In this paper, we have provided a summary background in case study research appropriate to the development of definitive case studies in systems engineering and systems management. We identified a systems engineering framework that appears suitable to orchestrate successful case studies. The framework itself is robust in the sense that issues, such as ethical and legal issues, which affect systems engineering and management across the levels of product, processes, or policy are capable of being identified within the framework. Finally, we provided a number of pragmatic illustrations of case study results, with at least one illustration applicable to each of the cells in the 27-cell two-dimensional framework. Each of these is associated with at least one important systems engineering and management concept. It would be of interest also to demonstrate that the 27 vignettes meet the reliability and validity requirements of case study research described in the first section of the paper such that these represent scientifically justified research findings. If this were a single case study effort, this demonstration would, of course, be vital. Some suggestions for extension of the framework are provided in the course of our discussions. Others might include identification of a framework development process and measures of effectiveness for an instantiated framework, although these might well best be considered in terms of specific case studies. It is hoped that the methodological issues discussed in this paper will be of value for systems engineering and management case study research including, but not at all limited to a set of concurrent case studies now in progress at the Center for Systems Engineering of the Air University and the Air Force Institute of Technology.

REFERENCES

L. Adelman, Evaluating decision support and expert systems, Wiley, Hoboken, NJ, 1992.

L.B. Barnes, C.R. Christensen, and A. Hansen, Teaching and the case method, 3rd edition, Harvard Business School Press, Cambridge, MA, 1994.

P.G. Carlock and R.E. Fenton, Systems of Systems (SoS) enterprise systems engineering for information-intensive organizations, Syst Eng 4(4) (2001), 242–261.

T. Janssen and A.P. Sage, A support system for multiple perspectives knowledge management and conflict resolution, Int J Technol Management 19(3/4/5) (2000), 472–490.

A.P. Sage, Systems engineering, Wiley, New York, 1992.

A.P. Sage, Systems management for information technology and software engineering, Wiley, New York, 1995.

A.P. Sage, “System families,” McGraw Hill Yearbook of Science and Technology, McGraw-Hill, New York, 2004, pp. 344–347.

A.P. Sage and W.B. Rouse (Eds.), Handbook of systems engineering and management, Wiley, New York, 1999.

B.P. Shapiro, An introduction to cases Note 9-584-097, Harvard Business School, Cambridge, MA, 1988.

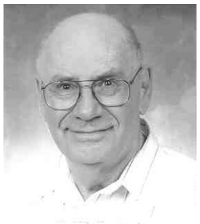
R.E. Stake, The art of case study research, Sage Publications, Thousand Oaks, CA, 1995.

J.M. Tien, “Evaluation of systems,” Handbook of systems engineering and management, A.P. Sage and W. B. Rouse (Editors), Wiley, New York , 1999, pp. 811–824.

S. Toulmin, R. Rieke, and A. Janik, An introduction to reasoning, Collier Macmillan, New York, 1984.

R.K Yin, Case study research: Design and methods, 3rd edition, Sage Publications, Thousand Oaks, CA, 2003a.

R.K. Yin, Applications of case study research, 2nd edition, Sage Publications, Thousand Oaks, CA, 2003.



George Friedman is presently Adjunct Professor of Engineering, Industrial and Systems Engineering Department, School of Engineering, University of Southern California, Los Angeles, and is director of research of the Space Studies Institute in Princeton. Previously, he retired as the Corporate Vice President of Engineering and Technology for the Northrop Corporation, was the Vice President of Publications for the Aerospace and Electronics Systems Society of IEEE, chairman of the Planetary Defense Committee of AIAA, and served as a consultant to the Air Force Scientific Advisory Board and the NATO Industrial Advisory Board. He is a founder of INCOSE, was its third President and serves on the editorial board of *Systems Engineering*. He is an elected Fellow of IEEE, INCOSE, and the Institute for the Advancement of Engineering (IAE). His paper on Constraint Theory was awarded the IEEE Baker Prize for the best paper published by the IEEE from all its societies in 1969. He received the B.S. in Mechanical Engineering from the University of California Berkeley and the M.S. and Ph.D. in Engineering from UCLA.

Andrew P. Sage received the BSEE degree from the Citadel, the SMEE degree from MIT, and the Ph.D. from Purdue, the latter in 1960. He received honorary Doctor of Engineering degrees from the University of Waterloo in 1987 and from Dalhousie University in 1997. He has been a faculty member at several universities, and in 1984 he became First American Bank Professor of Information Technology and Engineering at George Mason University and the first Dean of the School of Information Technology and Engineering. In May 1996, he was elected as Founding Dean Emeritus of the School and also was appointed a University Professor. He is an elected Fellow of the Institute of Electrical and Electronics Engineers, the American Association for the Advancement of Science, and the International Council on Systems Engineering. He is editor of the John Wiley textbook series on *Systems Engineering and Management*, the INCOSE Wiley journal *Systems Engineering*, and is coeditor of *Information, Knowledge, and Systems Management*. He edited the *IEEE Transactions on Systems, Man, and Cybernetics* from January 1972 through December 1998, and also served a two-year period as President of the IEEE SMC Society. In 1994 he received the Donald G. Fink Prize from the IEEE, and a Superior Public Service Award for his service on the CNA Corporation Board of Trustees from the US Secretary of the Navy. In 2000, he received the Simon Ramo Medal from the IEEE in recognition of his contributions to systems engineering and an IEEE Third Millennium Medal. In 2002, he received an Eta Kappa Nu Eminent Membership Award and the INCOSE Pioneer Award. His interests include systems engineering and management efforts in a variety of application areas including systems integration and architecting, reengineering, and industrial ecology and sustainable development.