N93-24680

SYSTEMS ENGINEERING FOR VERY LARGE SYSTEMS

158572

p. 6

by Paul E. Lewkowicz

Very large integrated systems have always posed special problems for engineers. Whether they are power generation systems, computer networks or space vehicles, whenever there are multiple interfaces, complex technologies or just demanding customers, the challenges are unique. "Systems engineering" has evolved as a discipline in order to meet these challenges by providing a structured, top-down design and development methodology for the engineer. This paper attempts to define the general class of problems requiring the complete systems engineering treatment and to show how systems engineering can be utilized to improve customer satisfaction and profitability. Specifically, this work will focus on a design methodology for the largest of systems, not necessarily in terms of physical size, but in terms of complexity and interconnectivity.

The literature has generally defined "systems engineering" as in this quote from W.P. Chase in *Management of System Engineering*:

> [Systems Engineering is] the process of selecting and synthesizing the application of . . . knowledge in order to translate system requirements into a system design and . . . to demonstrate that [it] can be effectively employed as a coherent whole to achieve some stated goal or purpose.

This definition points out, in the most general terms, that systems engineering is a process for ensuring that the customer requirements are satisfied. What it also implies is that this satisfaction must be achieved on time and for the agreed-upon price. It is this implicit requirement that is most often unfulfilled in complex engineering projects.

Recent efforts at Hughes Aircraft Company's Space & Communications Group have focused on sharpening the definition of systems engineering and defining standards for improving the implementation of the full systems engineering methodology on large spacecraft programs. Since these programs typically cost in the $100 million range, the pressure to deliver specified performance on time and on budget is enormous. A casual review of programs within the author's experience has shown that the classical approach to systems engineering has been followed throughout, but with varying uniformity and overall success. The question to answer, in the context of even more advanced, more demanding projects, is: "How can it be done better?"

The "classical" method of systems engineering alluded to above consists of requirements definition, technology assessment, solution synthesis and performance verification: four successive steps in the design of the mission solution. Typically, this is an iterative process, since requirements and technology rarely remain static. The customer's mission can be altered by events or even by a better understanding of the technology, risks or costs involved. Synthesized solutions, too, depend on the technology available, as well as the question asked. Often, the proposed technology does not live up to expectations, resulting in a "new" solution and reverification: an embarrassing situation at best, an extremely costly one at worst.

When the verification (or testing) phase of the systems engineering process uncovers a fault, the cause can often be traced to incomplete or improperly stated requirements. An example of this fact is a problem uncovered on one particular series of satellites; an on-orbit failure resulted in the loss of some 16 channels of telemetry data. The failure analysis, performed by the program's

PRECEDING PAGE BLANK NOT FILMED 16 INTENTIONALLY BLANK systems engineering staff, identified the cause as an open circuit in a particular unit. This fault produced an abnormally high telemetry output signal on one channel, which in turn resulted in the degradation of all 16 inputs to the telemetry multiplexer. Had systems engineering levied a requirement to protect against failure-induced overvoltages (via a simple circuit redundancy technique at the unit), only the failed telemetry channel would have been lost, instead of that of 15 other units as well.

The point here is that it is a knowledge of the needs of the whole system that is required, instead of only the needs of the parts. This knowledge exemplifies the principle of "engineering leverage" whereby a few engineers, representing a broad experience base, performing the logical, methodical systems design work, can save money over trial and error or crisis-oriented engineering. It is the concentration of systems knowledge, the "big picture" view, that allows for efficient designs all through the system.

A common question is: "How much systems engineering is required for a given project?" This can usually be interpreted as "How much will this cost?" Clearly a design team with unlimited funds can perform complete requirements analysis, all manner of failure analysis and simulations, and extensive part and unit environmental testing to fully optimize the design of some particular product. But if that product is, say, a ballpoint pen, have they really made it better from the manufacturer's standpoint? Or have they succeeded in making the most expensive writing instrument the world has ever known? The application of systems engineering techniques to a project is a matter of appropriate degree; how much engineering is required to ensure the customer's satisfaction becomes the first question any organization must ask before they can set up a systems engineering program.

This example emphasizes the fact that systems engineering costs are a direct charge to the effort, so the total cost of the engineering must be distributed over the entire production run. Even if the run is large, as in the ballpoint pen case, when the product normally sells for 39 cents, if the engineering costs run into the millions, then the manufacturer could be in serious trouble. For smaller production runs, like a satellite or submarine contract, systems engineering costs can still drive the final sale price, but systems engineering can also reduce the price by preventing errors and rework.

THE SYSTEMS ENGINEERING METHODOLOGY

The procedure followed in systems engineering consists of four distinct phases, described here in the simplest terms: requirements definition, technology assessment, solution synthesis and performance verification. These sobriquets are intended to be mnemonic; the details of what they really signify are presented below.

Requirements Analysis. The initial step consists of defining the problem to be solved and the constraints on the solution set. This is perhaps the single most critical phase of the systems engineering process in that a misunderstanding of the problem to be solved, either in characterizing it or defining the context of the solution, can result in an erroneous conclusion. As in the satellite telemetry example, the customer can be somewhat less than satisfied when a partial solution is delivered.

In large systems, the problem definition is usually described by the contractual documents. The request for proposal (RFP) or the statement of work typically contains directives as to the overall mission of the system, but these are not always completely specific; some interpretation of what the customer really meant is often required.

Another aspect of requirements analysis often underappreciated is that of constraining the solution. The RFP for a program may state that only a certain rocket booster or parts of a specific grade can be used, but the implications of such statements, and especially the implications of the "unstated" or "implied" requirements, can have serious consequences in the final design. These requirements, sometimes called derived or secondary requirements, determine the limits of the parametric trades that can be made in characterizing the problem's solution.

Technology Assessment. Once the basic requirements, both primary and secondary, are in place and understood by the design team, the technology available to solve the problem can be examined for suitability. This step is intuitively obvious for small systems, but when complexity is high, making the appropriate choice is not always easy. Typical activities in this phase include comparative tradeoffs between different processes and materials, architectures and performance. The technology assessment phase may also consider the design and documentation methods and the management organization to be employed on a specific project. Overall, this phase is concerned with selecting the best tools for performing the system design.

Solution Synthesis. This is usually the most time-consuming step in engineering a system to perform complex tasks and meet stringent requirements simply because of the number of choices available. If the requirements are well understood and the available hardware and software appropriate to the task are known, then trade studies can be carried out (on paper) that result in myriad viable combinations. During this phase, compromises are often required in order to satisfy conflicting requirements. For example, in a communications system design, a large antenna may be desired to provide high gain, but this will reduce its coverage capability by reducing the beamwidth. Out of this sea of alternatives must come a single "best fit" solution, meeting all of the original and derived requirements, especially such items pertaining to cost and producibility. If it can't be built or bought, then it's not the right answer.

Performance Verification. Last, but definitely not least, is the performance verification or testing phase. The task here is to prove, with all the rigor possible, that the suggested solution does in fact meet all of the system requirements in a clearly documented way. A standard approach is to utilize specification trees and a verification matrix to show where each requirement from the original customer's source documents is captured in lower level specifications. Additionally, the verification matrix shows how compliance with the requirement is proven, either by inspection, test, demonstration or analysis. In general, the specification system is designed to show a clear, unambiguous flowdown of all system requirements into individual component designs. The verification phase is the test of this flowdown as well as a measure of system performance.

REQUIREMENTS FOR SUCCESSFUL SYSTEMS ENGINEERING

The foregoing text has all been a precursor to this: exactly what does an organization have to do to apply a full-scale systems engineering approach to their work? And, perhaps more importantly, what does it cost that organization? As expected, in systems engineering, as in life, there are no free lunches. This section details the inputs to the process, or what is required by a systems engineering organization in order to function properly.

Formality. First and foremost, a formal, planned approach to the systems engineering process must be in place. Not only must the "generic" methodology for systems engineering be understood by all involved, the detailed program plans for the specific application of systems engineering must reflect this commitment. The major components in the formal system are review procedures, specification generation and maintenance (or "configuration control") procedures, and planning.

As can be deduced from the discussion of the phases of the systems engineering process, some degree of review and checking is inherent to all operations. The establishment of specification and design review teams to examine the documents (e.g., specifications, trade study reports, etc.) and help polish them into complete and correct inputs to the final design cannot be avoided. Without concrete review milestones, the design will often wander and become unfocused with respect to its objectives, which results in inefficient time and money management.

Since the specifications define the problem to be solved and its constraints, it is clear that they must be reliable and well documented. The configuration control function is to provide a routine for the introduction, validation and documentation of new requirements and the updating of old ones within the system. This is an important step in the review process, as well as the design process, in that all parties (customer and contractor alike) need a stable, well-defined basis of judgment for the validation of the system.

Planning is mentioned last in this case only for emphasis: without complete planning for the entire system design effort, from requirements definition through systems engineering, production, and final deployment, the project is doomed to failure. Every management textbook in the world expounds this fact in detail, yet weak planning is still a major cause of cost overruns and poor performance in all types of industry.

Information Exchange. While formality and procedure allow tight control of the requirements, informality and open communications are the key to efficient design and problem resolution. Not only must the contractor communicate effectively with the customer, but the various elements of the contractor's organization (management, system engineers, unit designers, etc.) must all talk to each other in order to completely understand the requirements. In every program there are stated goals and hidden goals, real requirements and perceived requirements; it all depends on where the observer is looking from. Communications and open channels between all participants, regardless of title or rank, are absolutely essential to all phases of the job.

Technology Base. "Technology" in this context means more than the hardware and software that can be employed in a design solution; it encompasses the organizations and information architectures as well. As a system becomes larger and more complex, so too does the technology or "knowledge base" required to fully define the implementation of system requirements. Such a base might include other contractors, national resources (e.g., the Space Transportation System), specialized electronic devices, etc. In short, practically any conceivable problems, and even a few inconceivable ones, can come up in systems design. To deal effectively with them, the systems engineering team must have the knowledge and experience to recognize solutions from a wide selection of possibilities.

Dedication and Staffing. Finally, the one factor that takes system engineering from an abstract concept to a practical reality is the dedication of the people involved. In order to even begin a design for a complex system, a design team is required. Not a single guru and a few part-time acolytes, but a team of committed managers and engineers with experience in real-world problem solving, technical breadth and clearly defined roles in the systems engineering process. Without this core team, the continuity and rigor required by the process to ensure a coherent, effective solution cannot possibly exist.

Just as planning is the key to a successful project, leadership is the key to a successful team. The complexity of the designs under discussion are such that (typically) a wide range of talents are needed to arrive at a solution. This diversity can be dangerous without direction, because diversity is just a polite name for chaos waiting to happen. A group with a broad technical background, when presented a problem without leadership, will always seek to maximize its entropy. The project staff must be directed and focused at all times in order to move through the systems engineering process. After all, efficiency and minimal engineering costs concern the entire group. The depth necessary to perform the detailed designs need not come from the systems staff, however; this is often not possible given the generalist nature required of them. Most companies employ a unit engineering staff to design the components of the complete solution, which simply reflects the top-down design approach of breaking each requirement down into smaller and smaller functional blocks.

An important factor to consider is time. It may take several months or even years to complete the design of a complex system, so continuity becomes a factor in the staffing of the design team. The deleterious effects of change on an organization are well known, and so are those of miscommunication. The training of systems engineers, whether through formal schooling or on-the-job education, is the first step toward building a self-perpetuating, self-replicating design methodology. Experienced staff members are able to produce more and overcome obstacles better than those less experienced; reinventing the wheel is avoided. Additionally, experienced people add synergy to the team by virtue of shared experiences. Synergism in the design process is how the engineering leverage of systems engineering is released, by the magnification of individual efforts. A fringe benefit of this magnification is growth in the individuals involved. The less experienced become more experienced and leadership skills are developed and honed. Not only does the design process (and product) continue to improve but, through continuity and growth, the staff benefits personally as well.

What about the individual roles of the staff members? The need for a broad knowledge base, for generalists, is clear, but what do they do? As in any team-building situation, all members need clearly communicated job descriptions and management expectations; this applies to all members of the project team from the most senior manager to the last clerk. Once the work has started, they need tangible feedback on what is going correctly, according to expectations, and what is not. The immediate benefit to the organization is clear. Job satisfaction increases, and with it, a concomitant rise in overall productivity. Again, the process, when properly managed, feeds upon itself to work more efficiently.

COST VS. BENEFITS OF FULL-SCALE SYSTEMS ENGINEERING

The requirements levied upon systems design for very large projects are simple: provide full customer satisfaction on time and on budget for a set of diverse and complex functional specifications and interconnections. Likewise, the technology appropriate to this task is (hopefully) equally clear: employ a formal, full-scale systems engineering approach to meeting this challenge.

Costs:

- Management must be willing to allow group synergy to make decisions; the "group think" approach is mandatory.
- Personnel must be dedicated and immersed in the systems engineering of a single system. Teamwork and continuity must be fostered and preserved.
- The systems engineering organization can exhibit all the negative aspects of a bureaucracy if not managed precisely.
- Careful, rigorous planning is required for all aspects of the program up-front, before the work begins, which often means extra bidding expense.

Benefits:

+ Customer satisfaction is enhanced through demonstrated performance and the opportunity for full customer involvement in the design process.
+ Manageability is improved by accurate, more complete planning and a well-defined staff structure.
+ Contingencies are worked out in advance, resulting in fewer surprises during the design, test and production phases.
+ Better cost performance is achieved due to reduced redesigns, reworks and "patches."

After an analysis of the costs and benefits of implementing a systems engineering solution to a complex design problem, it becomes apparent that the benefits outweigh the costs, especially in terms of the potential for productivity and cost improvements. The chief drawback of this method is that it is difficult to implement in organizations that do not already practice some form of systems engineering, due to the cultural adjustments that are often necessary. Once the need for a rigorous design methodology is apparent, the systems engineering process of requirements analysis, technology assessment, solution synthesis and performance verification can be utilized to provide an efficient, cost-effective solution to the managerial and technical challenges.

The author wishes to thank Dr. Thomas A. Brackey, W. Richard Brown, and Gávien Miyata of the Hughes Aircraft Company for their support and mentorship in several complex design projects.

REFERENCES

Chase, W.P. Management of *System Engineering,* as quoted in Hughes, Seminar.

Defense Systems Management College. *Systems Engineering Management Guide,* U.S. Air Force, 3 October 1983.

Hughes Aircraft Company. *Systems Engineering Seminar for General Motors*, internal memorandum, 1987.

-----. "S&CG Practice 5-0-53," internal memorandum, 21 July 1987.

-----. "Systems Engineering Division Mission, Goals, and Objectives," internal memorandum, 8 October 1987.

IEEE *Standard Dictionary of Electrical and Electronics Terms*, IEEE Press, Third Edition, 1984 (ANSI/IEEE Std 100-1984).

IEEE Spectrum special report, *On Good Design*, Volume 24, Number 5, May 1987.

U.S. Government MIL-STD-499.