Tackling the Cost Challenges of System of Systems

Arlene F. Minkiewicz
PRICE Systems

Systems & Software Technology Conference
Monday, 1 May 2006
Track 1: 3:55 – 4:40 p.m.
Ballroom A

The Department of Defense and its contractors are currently facing unprecedented challenges in planning projects involving groups of systems integrated into one large system of systems (SOS). These challenges are intensified by the fact that these systems tend to be heavily software-dependent. Often planners must decide which configuration of platforms best meets mission needs with respect to affordability, performance, and risk in the very early stages of a project from top-level requirements. This article presents research of the cost issues associated with delivery of SOS capabilities. It starts with a discussion on what SOS are and areas where SOS projects vary from typical system development and deployment. New and expanded contractor roles and activities are presented, highlighting how these drive cost differences from traditional system projects. Guidelines are provided for performing high-level analysis of SOS costs to enable decision makers to perform trade-offs between various configurations in order to pursue the most affordable solution that will meet mission needs.

The Department of Defense (DoD) has migrated from a platform-based acquisition strategy to one focused on delivering capabilities. Instead of delivering a fighter aircraft or an unmanned air vehicle, contractors are now being asked to deliver the right collection of hardware and software to meet specific wartime challenges. This means that much of the burden associated with conceptualizing, architecting, integrating, implementing, and deploying complex capabilities into the field has shifted from desks in the Pentagon to desks at Lockheed Martin, Boeing, Rockwell, and other large aerospace and defense contractors.

In "The Army's Future Combat Systems' [FCS] Features, Risks and Alternatives," the Government Accounting Office states the challenge as:

> …14 major weapons systems or platforms have to be designed and integrated simultaneously and within strict size and weight limitations in less time than is typically taken to develop, demonstrate, and field a single system. At least 53 technologies that are considered critical to achieving critical performance capabilities will need to be matured and integrated into the system of systems. And the development, demonstration, and production of as many as 157 complementary systems will need to be synchronized with FCS content and schedule. [1]

The planning, managing, and execution of such projects will require changes in the way organizations do business. This article reports on ongoing research into the cost challenges associated with planning and executing a system of systems (SOS) project. Because of the relatively immature nature of this acquisition strategy, there is not nearly enough hard data to establish statistically significant cost-estimating relationships. The conclusions drawn to date are based on what we know about the cost of system engineering and project management activities in more traditional component system projects augmented with research on the added factors that drive complexities at the SOS level.

The article begins with a discussion of what an SOS is and how projects that deliver SOS differ from those projects delivering stand-alone systems. Following this is a discussion of the new and expanded roles and activities associated with SOS that highlight increased involvement of system engineering resources. The focus then shifts to cost drivers for delivering the SOS capability that ties together and optimizes contributions from the many component systems. The article concludes with some guidelines for using these cost drivers to perform top-level analysis and trade-offs focused on delivering the most affordable solution that will satisfy mission needs.

Related Research

Extensive research has been conducted on many aspects of SOS by the DoD, academic institutions, and industry. Earlier research focused mainly on requirements, architecture, test and evaluation, and project management [2, 3, 4, 5, 6, 7, 8]. As time goes on and the industry gets a better handle on the technological and management complexities of SOS delivery, the research expands from a focus on the right way to solve the problem to a focus on the right way to solve the problem *affordably*. In the forefront of this cost-focused research is the University of Southern California's Center for Software Engineering [9], the Defense Acquisition University [10], Carnegie Mellon's Software Engineering Institute [11], and Cranfield University [12].

What Is an SOS?

An SOS is a configuration of component systems that are independently useful but synergistically superior when acting in concert. In other words, it represents a collection of systems whose capabilities, when acting together, are greater than the sum of the capabilities of each system acting alone.

According to Mair [13], an SOS must have most, if not all, of the following characteristics:

- Operational independence of component systems.
- Managerial independence of component systems.
- Geographical distribution.
- Emergent behavior.
- Evolutionary development processes.

For the purposes of this research, this definition has been expanded to explicitly state that there be a network-centric focus that enables these systems to communicate effectively and efficiently.

Today, there are many platforms deployed throughout the battlefield with limited means of communication. This becomes increasingly problematic as multiple services are deployed on a single mission as there is no consistent means for the Army to communicate with the Navy or the Navy to communicate with the Air Force. Inconsistent and unpredictable means of communication across the battlefield often results in unacceptable time from detection of a threat to engagement. This can ultimately endanger the lives of our service men and women.

One example of an SoS that the Army is currently envisioning is the Warfighter Information Network-Tactical (WIN-T), which is a communication system designed for reliable, secure, and seamless video, data, imagery, and voice services to enable decisive, real-time combat actions. This SOS promises full two-way communication between platforms and across services, making it possible for information to be shared and processed in time to make a real difference in the outcome. The cloud is being lifted from the battlefield!

How Different Are SOS Projects?

How much different is a project intended to deliver an SOS capability from a project that delivers an individual platform such as an aircraft or a submarine? Each case presents a set of customer requirements that need to be elicited, understood, and maintained. Based on these requirements, a solution is crafted, implemented, integrated, tested, verified, deployed, and maintained. At this level, the two projects are similar in many ways. Dig a little deeper and differences begin to emerge. The differences fall into several categories: acquisition strategy, software, hardware, and overall complexity

The SOS acquisition strategy is capability-based rather than platform-based. For example, the customer presents a contractor with a set of capabilities to satisfy particular battlefield requirements. The contractor then needs to determine the right mix of platforms, the sources of those platforms, where existing technology is adequate, and where invention is required. Once those questions are answered, the contractor must decide how best to integrate all the pieces to satisfy the initial requirements. This capability-based strategy leads to a project with many diverse stakeholders. Besides the contractor selected as the lead system integrator (LSI), other stakeholders that may be involved include representatives from multiple services, Defense Advanced Research Projects Agency, prime contractor(s) responsible for supplying component systems, and their subcontractors. Each of these stakeholders brings to the table different motivations, priorities, values, and business practices – each brings new people management issues to the project.

Software is an important part of most projects delivered to DoD customers. In addition to satisfying the requirements necessary to function independently, each of the component systems needs to support the interoperability required to function as a part of the entire SOS solution. Much of this interoperability will be supplied through the software resident in the component systems. This requirement for interoperability dictates that well-specified and applied communication protocols are a key success factor when deploying an SOS. Standards are crucial, especially for the software interfaces. Additionally, because of the need to deliver large amounts of capability in shorter and shorter timeframes, the importance of commercial off-the-shelf (COTS) software in SoS projects continues to grow.

With platform-based acquisitions, the customer generally has a fairly complete understanding of the requirements early on in the project with a limited amount of requirements growth once the project commences. Because of the large scale and long-term nature of capability-based acquisitions, the requirements tend to emerge over time with changes in governments, policies, and world situations. Because requirements are emergent, planning and execution of both hardware and software contributions to the SOS project are impacted.

SOS projects are also affected by the fact that the hardware components being used are of varying ages and technologies. In some cases, an existing hardware platform is being modified or upgraded to meet increased needs of operating in an SOS environment, while in other instances brand new equipment with state-of-the-art technologies is being developed. SOS project teams need to deal with components that span the spectrum from the high-tech, but relatively untested to the low-tech, tried-and-true technologies and equipment.

Basically, a project to deliver an SOS capability is similar in nature to a project intended to deliver a specific platform except that overall project complexity may be increased substantially. These complexities grow from capability-based acquisition strategies, increased number of stakeholders, increased overall cost (and the corresponding increased political pressure), emergent requirements, interoperability, and equipment in all stages from infancy to near retirement.

New and Expanded Roles and Activities

Understanding the manifestation of these increased complexities on a project is the first step to determining how the planning and control of an SOS project differs from that for a project that delivers one of the component systems. One of the biggest and most obvious differences in the project team is the existence of an LSI. The LSI is the contractor tasked with the delivery of the SOS that will deliver the capabilities the DoD customer is looking for. The LSI can be thought of as the *super* prime or the *prime of prime* contractors. He or she is responsible for managing all the other primes and contractors and ultimately for fielding the required capabilities. The main areas of focus for the LSI include:

- Requirements analysis for the SOS.
- Design of SOS architecture.
- Evaluation, selection, and acquisition of component systems.
- Integration and test of the SOS.
- Modeling and simulation.
- Risk analysis, avoidance, and mitigation.
- Overall program management for the SOS.

One of the primary jobs of the LSI is completing the system engineering tasks at the SOS level.

Focus on System Engineering

The following is according to the Encyclopedia Britannica:

> … system engineering is a technique of using knowledge from various branches of engineering and science to introduce technological innovations into the planning and development stages of systems. Systems engineering is not as much a branch of engineering as it is a technique for applying knowledge from other branches of engineering and disciplines of science in an effective combination. [14]

System engineering as a discipline

first emerged during World War II as technology improvements collided with the need for more complex systems on the battlefield. As systems grew in complexity, it became apparent that it was necessary for there to be an engineering presence well versed in many engineering and science disciplines to lend an understanding of the entire problem a system needed to solve. To quote Admiral Grace Hopper, "Life was simple before World War II. After that, we had systems [15]."

With this top-level view, the system engineers were able to grasp how best to optimize emerging technologies to address the specific complexities of a problem. Where an electrical engineer would concoct a solution focused on the latest electronic devices and a software engineer would develop the best software solution, the system engineer knows enough about both disciplines to craft a solution that gets the best overall value from technology. Additionally, the system engineer has the proper understanding of the entire system to perform validation and verification upon completion, ensuring that all component pieces work together as required.

Today, a new level of complexity has been added with the emerging need for SOS, and once again the diverse expertise of the system engineers is required to overcome this complexity. System engineers need to comprehend the big picture problem(s) whose solution is to be provided by the SOS. They need to break these requirements down into the hardware platforms and software pieces that best deliver the desired capability, and they need to have proper insight into the development, production, and deployment of the component systems to ensure not only that they will meet their independent requirements, but also that they will be designed and implemented to properly satisfy the interoperability and interface requirements of the SOS. It is the task of the systems engineers to verify and validate that the component systems, when acting in concert with other component systems, do indeed deliver the necessary capabilities.

Cost Considerations of SOS Projects

An SOS is a collection of existing, upgraded, and new systems that are required to work together to accomplish specific objectives. Clearly the costs of developing and acquiring component systems is one important cost consideration, but since estimating system costs is a fairly mature discipline, this article focuses on the additional costs associated with the delivery of capabilities made possible when a configuration of such systems works as a system.

Mastering the cost questions in an SOS project first requires establishing a link between the increased complexities and the participation of system engineers in the project. A traditional parametric estimating methodology for hardware or software systems relies on a quantification of the size and complexity of the system being developed. Size is driven by weight for hardware and source lines of code or function points for software. Project, process, and organizational factors drive complexity. Assigning a size and complexity to an SOS is a bit trickier. Traditional size measures alone are not adequate for estimating the size of system engineering tasks and with many participating organizations, the process and organizational factors can vary substantially within the project team.

> ***"Mastering the cost questions in an SOS project first requires establishing a link between the increased complexities and the participation of system engineers in the project."***

Tricky or not, being able to properly size the SOS part of a project is crucial to successfully determining what it will cost and how long it will take to deliver. It is also a crucial step in being able to make trade-offs in order to deliver a solution that not only meets requirements, but also satisfies affordability constraints. As with all estimating, the challenge in sizing an SOS is being able to translate what is known early on in the project into information that represents useful project characteristics as the project evolves.

Toward this end, our research indicates that the number of unique interface protocols and the number of different component systems are the two best factors for determining the size of the SOS effort. In an SOS project, it is the LSI's job to define and design the infrastructure that will facilitate communication among the many component systems. The number of unique interface protocols is clearly a good start for determining problem space size. Augmentation of this number with the number of component systems that will be designed for or adapted to operate within this infrastructure provides an even better proxy for the size of the solution. This conclusion is consistent with the research done at the University of Southern California Center for Software Engineering on the Constructive System of System Integration Model [9].

The number of unique interface protocols drives the size of the integration and test effort. Our experience is that the effort for integration and test within a typical system ranges between 5 percent and 40 percent of the entire development effort of the system as the number of interfaces goes from few to many; this effect would be exaggerated in an SOS as complexity of the overall integration problem is greater. As the number of component systems increases, integration efforts increase in a non-linear fashion as a result of the diseconomy of scale brought on by project complexity. Additionally, the number of components will influence management and oversight costs in the form of added people and communication issues.

Size, of course, is only part of the puzzle. Multiple SOS within the same size range will only fall into the same cost range as a coincidence. For the sake of this discussion, consider the simplistic cost model that applies an exponent and a coefficient to a project size. In this context, the size is as described and the exponent and coefficient are determined by factors that determine project complexity. As such, it is necessary to assign relative complexity values to the various configurations. There are many factors that have a potential impact on complexity, some that are obvious early on in the project, and others that will emerge throughout the project life cycle. The ones that are available or predictable early on in the project and that appear to have the most significant impact on the amount of effort required for the SOS tasks include those in the following sections.

Number of Operational Scenarios

An operational scenario refers to a particular capability instance for some set of the component systems of the SOS. For example, the Coast Guard's Integrated

Deepwater System needs to include capability that can react to a terrorist threat, a person lost at sea, or a drug-smuggling operation. The number of operational scenarios impacts the coefficient in the cost equation discussed earlier as additional scenarios result in more time for requirements, design, and modeling and simulation. Depending on the similarities of the scenarios, the impacts to these activity's costs should represent increases between 10 percent and 50 percent.

Required Level for Acceptance of Key Performance Parameters

Key performance parameters associated with an SOS include things like detection effectiveness, survivability, and lethality. This factor could have substantial impact on both the coefficient and the exponent in the simple cost model mentioned earlier. System engineering activities associated with the SOS could double or triple or more as the detection effectiveness expectations move from available technology to state of the art. Use of immature technology on the Joint Tactical Radio System Program was cited as one of the main reasons for a $458 million development cost increase [16].

Number of Suppliers and Stakeholders

The number of players involved in an SOS project can increase the complexity and cost significantly. On a typical system project, people and communication issues can increase the cost of project management and oversight activities by as much as 60 percent. This effect can increase dramatically as the relatively well-known confines of the typical system are replaced with the much more expansive and undefined constraints on an SOS project.

Integration Complexity

Integration complexity is a quantification of the amount of integration each component is expected to require with the rest of the SOS. A SOS that requires highly complex integrations within and among each of its component systems could potentially see the integration and test activity costs increase an order of magnitude from an SOS where all of the integrations are simple, well-defined tasks.

Stability and Readiness of Components

As mentioned earlier, the technical immaturity of components can substantially impact system engineering tasks. Additionally, immature components can impact the overall schedule and cost of the SOS, since integration and test activities for various capabilities will be delayed until all required components are available. The WIN-T program was originally planned to deliver technologies not expected to mature until after production started. Such a strategy is guaranteed to lead to costly schedule delays.

Amount of COTS Capability

COTS components generally require modification, integration, and test as well as compromise on SOS requirements. When looking at the overall cost for an SOS, off-the-shelf components should decrease the cost compared to newly developed components. From the perspective of the LSI, however, they represent an increase of system engineering effort associated with requirements, design and integration, and test. This cost increase can be quite modest if the components and vendors are chosen wisely, but it could double the costs of these activities if poor choices are made.

> *"Although there is an upfront investment in getting the architectures right and standardizing communication protocols, the payoff is significant during delivery of the initial operating concept and throughout the life of the SOS"*

Affordable SOS

When crafting a solution to deliver an SOS capability, there are things the LSI can do to ensure that it not only meets all performance requirements, but does so within affordability constraints. All possible solutions should be focused on the specified constraints for stated key performance parameters (KPPs). No solution should be presented that does not satisfy these constraints. Component systems that drive performance substantially above specified performance in these areas should be carefully scrutinized as well. All possible solutions should first be validated to ensure that they successfully address all KPPs and support all operational scenarios.

Care should be taken to utilize as many existing component systems as possible rather than developing new ones. When new component systems must be developed to deliver some currently non-existing capability or degree of performance, it is important to get the most from the technology investment. Attempts should be made to incorporate as much capability as practical into the new development to reduce the number of different component systems. Increases in complexity associated with technology readiness and component stability may be offset by size decreases if the number of required component systems can be reduced. At the same time, care should be taken to ensure that expectations for technology do not exceed practical limits on innovation imposed by schedule constraints on the program.

Well-thought-out architecture with simple communication protocols that meet many different needs will reduce the size of the SOS solution space. Although there is an up-front investment in getting the architectures right and standardizing communication protocols, the payoff is significant during delivery of the initial operating concept and throughout the life of the SOS. Emerging requirements will result in the addition of new component systems that must communicate with existing components.

The use of COTS hardware and software is a practical and necessary approach to accomplish the delivery of SOS capabilities in required timeframes. When possible, the same vendor should be considered for multiple components, parts, or software products. This reduces the number of vendors involved in the project, eases the effort to integrate between components, and could possibly result in favorable purchasing agreements based on bulk. Integration complexity can also be reduced through simple standards that are strictly enforced, effective risk management techniques focused on early identification and mitigation, and ongoing integration efforts.

Conclusion

Today, SOS solutions are replacing the existing post-World War II systems as the next generation of complex solutions supplied by contractors to the DoD. SOS projects require contractors to deliver capabilities rather than standalone systems. Contractors are left to decide on and acquire component systems, determine the best configuration for these component systems to achieve the required capabilities, and develop the best plan for interoperability among the component systems.

While there are some ways in which an SOS project is similar to a project that delivers a component system, there are many ways in which the two types of projects differ. Understanding these differences and how they affect the cost and effort associated with a project is crucial to proper planning and execution of an SOS project. A crucial difference is the requirement for increased involvement of system engineering resources throughout the life cycle of the SOS project. System engineers are involved in requirements elicitation and management, architecture decisions, test and evaluation, verification and validation, and technical oversight for the SOS project.

Cost drivers for an SOS fall into two categories: those that define the size of the system engineering tasks, and those that drive the complexity of the engineering and management tasks. Because the notion of capability-based acquisitions is still relatively immature, there is not the preponderance of data required to develop good, strong cost-estimating relationships for SOS project activities. Despite this, it is possible – and necessary – to begin estimating these projects today by incorporating estimating knowledge gained through years of system development augmented with information about the additional factors that influence SOS project size and complexities. Future directions for this research involve collecting data from evolving SOS projects as they reach milestones and use this data to refine, update, or replace cost-estimating relationships.◆

References

1. Francis, Paul L. The Army's Future Combat System's Features, Risks and Alternatives. Testimony before the Subcommittee on Tactical Air and Land Forces. Committee on Armed Services. House of Representatives, GAO-04-635T 1 Apr. 2004 <www.gao.gov/new.items/d04635t.pdf>.
2. Lamartin, Glenn. "The Role of T&E in the Systems Engineering Process." National Defense Industrial Association System Engineering Conference, 17 Aug. 2004 <www. acq.osd.mil/ds/se/speeches.htm>.
3. Hooks, Ivy. "Managing Requirements for a System of Systems." CrossTalk Aug. 2004 <www.stsc.hill.af.mil/crosstalk/2004/08/index.html>.
4. Krone, Roger. "Managing a Complex System-of-Systems." President's Commission on Moon, Mars and Beyond, 4 May 2004 <www.govinfo.library.unt.edu/moontomars/news/docs.asp>.
5. Martin, James N. "Modeling and Architecture Considerations for Systems of Systems." 2004 Systems and Software Technology Conference, Salt Lake City, UT, 21 Apr. 2004.
6. Carney, D., and P. Oberndorf. "Integration and Interoperability Models for Systems of Systems." 2004 Systems and Software Technology Conference, Salt Lake City, UT, 1 Apr. 2004.
7. Crossley, William A. "System of Systems: An Introduction of Purdue University Schools of Engineering's Signature Area." Engineering Systems Symposium, MIT Engineering Systems Division, Mar. 2004.
8. Conrow, Edmund H. "Risk Management for Systems of Systems." CrossTalk Feb. 2005 <www.stsc.hill.af.mil/crosstalk/2005/02/0502conrow.html>.
9. Lane, Jo Ann. "Factors Influencing System-of-Systems Architecting and Integration Costs." University of Southern California Center for Software Engineering <www.stevens-tech.edu/cser/authors/46.pdf>.
10. Flowe, R., and M. Spurlock. "Systems of Systems Research Project Overview." Office of the Secretary of Defense Program Analysis and Evaluation, Mar. 2004 <http://acc.dau.mil/simplify/ev_en.php >.
11. Zubrow, Dave. "System of Systems Integration Cost Driver Research." 37th Annual DoD Cost Analysis Symposium, Feb. 2004.
12. Adcock, Rick. "Principles and Practices of Systems Engineering." INCOSE UK Chapter Library, Nov. 2001 <www.incose.org.uk/library.htm>.
13. Mair, M.W. "Architecting Principles for System-of-Systems." Systems Engineer 1.4 (1998).
14. Encyclopedia Britannica Online <www.britannica.com>.
15. Wikipedia <http://en.wikipedia.org/wiki/system.engineering>.
16. Francis, Paul L. Defense Acquisitions, Future Combat Systems Challenges and Prospects for Success. Testimony Before the Sub-committee on Airland. Committee on Armed Services. U.S. Senate, Government Accounting Office, 16 Mar. 2005 <www.gao.gov>.

About the Author

Arlene F. Minkiewicz is chief scientist of the Cost Research Department at PRICE Systems. She is responsible for the research and analysis necessary to keep the suite of PRICE estimating products responsive to current cost trends. In her 20-year tenure with PRICE, Minkiewicz has researched and developed the software cost estimating relationships that were the cornerstone for PRICE's commercial software cost estimating model, ForeSight, and invented the Cost Estimating Wizards originally used in ForeSight that walk the user through a series of high-level questions to produce a quick cost analysis. As part of this effort she has invented a sizing measurement paradigm for object-oriented analysis and design that allows estimators a more efficient and effective way to estimate software size. She recently received awards from the International Society of Parametric Analysts and the Society of Cost Estimating and Analysis for her white paper "The Real Cost of COTS." Minkiewicz contributed to a new parametric cost estimating book with the Consortium for Advanced Manufacturing International called "The Closed Loop: Implementing Activity-Based Planning and Budgeting," and she frequently publishes articles on software estimation and measurement. She has also been a contributing author for several books on software measurement and speaks frequently on this topic at numerous conferences.

PRICE Systems
17000 Commerce PKWY STE A
Mt. Laurel, NJ 08054
Phone: (856) 608-7222
Fax: (856) 608-7247
E-mail: arlene.minkiewicz@pricesystems.com