CrossTalk

May 2007 The Journal of Defense Software Engineering Vol. 20 No. 5

Software Acquisition

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE
MAY 2007

2. REPORT TYPE

3. DATES COVERED
00-00-2007 to 00-00-2007

4. TITLE AND SUBTITLE
CrossTalk: The Journal of Defense Software Engineering. Volume 20, Number 5, May 2007

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
OO-ALC/MASE,6022 Fir Ave,Hill AFB,UT,84056-5820

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**32**	

Software Acquisition

Open Forum

On The Cover

Cover Design by
Kent Bingham

Additional art services
provided by Janna Jensen

Departments

CrossTalk

Co-Sponsors:
DoD-CIO *The Honorable John Grimes*
NAVAIR *Jeff Schwalb*
76 SMXG *Kevin Stamey*
309 SMXG *Randy Hill*
402 SMXG *Diane Suchan*
DHS *Joe Jarzombek*

Staff:
Managing Director *Brent Baxter*
Publisher *Elizabeth Starrett*
Managing Editor *Kase Johnstun*
Associate Editor *Chelene Fortier-Lozancich*
Article Coordinator *Nicole Kentta*
Phone (801) 775-5555
E-mail crosstalk.staff@hill.af.mil
CrossTalk Online www.stsc.hill.af.mil/crosstalk

CrossTalk, The Journal of Defense Software Engineering is co-sponsored by the Department of Defense Chief Information Office (DoD-CIO); U.S. Navy (USN); U.S. Air Force (USAF); Defense Finance and Accounting Services (DFAS); and the U.S. Department of Homeland Security (DHS). DoD-CIO co-sponsor: Assistant Secretary of Defense (Networks and Information Integration). USN co-sponsor: Naval Air Systems Command. USAF co-sponsors: Oklahoma City-Air Logistics Center (ALC) 76 Software Maintenance Group (SMXG); Ogden-ALC 309 SMXG; and Warner Robins-ALC 402 SMXG. DHS co-sponsor: National Cyber Security Division of the Office of Infrastructure Protection.

The USAF Software Technology Support Center (STSC) is the publisher of CrossTalk, providing both editorial oversight and technical review of the journal. CrossTalk's mission is to encourage the engineering development of software to improve the reliability, sustainability, and responsiveness of our warfighting capability.

S T S C

Subscriptions: Send correspondence concerning subscriptions and changes of address to the following address. You may e-mail us or use the form on p. 19.

517 SMXS/MXDEA
6022 Fir AVE
BLDG 1238
Hill AFB, UT 84056-5820

Article Submissions: We welcome articles of interest to the defense software community. Articles must be approved by the CrossTalk editorial board prior to publication. Please follow the Author Guidelines, available at <www.stsc.hill.af.mil/crosstalk/xtlkguid.pdf>. CrossTalk does not pay for submissions. Articles published in CrossTalk remain the property of the authors and may be submitted to other publications.

Reprints: Permission to reprint or post articles must be requested from the author or the copyright holder and coordinated with CrossTalk.

Trademarks and Endorsements: This Department of Defense (DoD) journal is an authorized publication for members of the DoD. Contents of CrossTalk are not necessarily the official views of, or endorsed by, the U.S. government, the DoD, the co-sponsors, or the STSC. All product names referenced in this issue are trademarks of their companies.

CrossTalk Online Services: See <www.stsc.hill.af.mil/crosstalk>, call (801) 777-0857 or e-mail <stsc.webmaster@hill.af.mil>.

Back Issues Available: Please phone or e-mail us to see if back issues are available free of charge.

Being a *Smart Buyer*

All of us want to be *smart buyers* whether we are spending our own money or we are guardians of taxpayer dollars. One of the articles in this month's issue compares buying a car to acquiring software or software development services. This is an excellent analogy. Today, most people are very competent car shoppers. They use tools such as the Internet, car magazines, buying guides, and lessons learned from others. People have a much better understanding of financing, options, fuel economy, predicted reliability, and so forth, in spite of all the models available.

Acquiring software intensive systems (SIS) for large tactical weapons or information systems is, of course, much more difficult than buying a car. For one thing, software is part of a system (we usually acquire systems, not software) and this system is often part of a larger system. As our systems become more complex, we see the bulk of the functionality being implemented in software rather than hardware. Another aspect affecting the complexity of acquiring an SIS is that not all system functionality is known at time of contract award.

Despite complexity and uncertainty, acquirers of an SIS must be *smart buyers* in order to deliver products that meet user requirements, that are delivered on time, and that are within budget. For the acquisition of an SIS program, office personnel must be skilled in the following areas:

- **Recognizing and selecting a competent supplier.** The response to a Request for Proposal (RFP) allows potential suppliers to describe their experience and expertise with software development. The acquirer must know how to craft the RFP to ask for the right information and then analyze that information once it is supplied to select the *best* source.
- **Defining software requirements and managing changes.** The early and unambiguous definition of system requirements is key to any acquisition. If the scope of requirements changes are not consistently tracked, assessed, controlled, and applied to revised software estimates, then the likelihood of program failure (cost/schedule/quality) increases significantly.
- **Accurately predicting cost and schedule and managing the risk associated with those predictions throughout the various phases of the program, especially when unrealistic cost and schedule constraints are imposed on a program.** An excellent article in this month's issue (*Controlling Software Acquisition Costs with Function Points and Estimation Tools*) addresses the use of function points and other estimating tools to better evaluate cost and schedule bids from offerors.
- **Using Earned Value Management (EVM) as applied to software consistently and correctly, to determine status of the development.** A useful EVM system is closely tied to good estimating and an accurate, detailed work breakdown structure.

Within the Navy, a renewed emphasis is being placed on improved acquisition office processes by instigating awareness, education, training, cultivation of competent leaders/managers who understand software, adherence to rigorous processes, and consolidated, clear-cut guidance. For this to be successful, senior leadership is critical. In May 2006, Dr. Delores Etter, Assistant Secretary of the Navy for Research, Development, and Acquisition kicked off a Software Process Improvement Initiative. Key elements of this initiative include required training for Navy senior acquisition personnel, use of a process model as a tool in software developer source selections, and more consistent processes for RFP preparation and contractor selection.

In this issue, the United States Army Program Executive Offices provide a unique perspective on software acquisition in *Software Acquisition in the Army*, noting the similarities and differences in the issues programs face. Key aspects of software assurance are addressed in *Software Assurance: Five Essential Considerations for Acquisition Officials*. The added complexity and interdependencies inherent in joint program acquisition is presented in *The Acquisition of Joint Programs: The Implications of Interdependencies*. Finally, four key recommendations in *Defense Acquisition Performance Assessment – The Life-Cycle Perspective of Selected Recommendations* are discussed from the perspective of their implementability.

Just as with buying a car, being a smart buyer doesn't eliminate the risk of getting gouged. But being a smart buyer does maximize your chances of success. As you read through this month's CrossTalk articles, I hope you will become a smarter buyer.

Tony Guido

Tony Guido
Head, Software Engineering Division
Naval Air Systems Command

Software Acquisition in the Army

Elizabeth Starrett
CrossTalk

There has been much discussion lately regarding the Global War on Terror's (GWOT) financial ramifications to the United States Army. While all of the Department of Defense (DoD) is challenged financially during the ongoing war, the Army appears to be most effected [1, 2]. *As* CrossTalk *prepared this issue on Software Acquisition, we thought* CrossTalk *readers would benefit from a discussion of this challenge, providing additional perspectives to acquisition efforts.*

CrossTalk contacted representative Army Program Executive Offices (PEOs) that deal with software and asked for their perspectives on several acquisition topics. We received responses from the following five PEOs (see the sidebar on page 5 for a brief description of each):

- Ammunition.
- Command, Control, and Communications-Tactical (C3T).
- Enterprise Information Services (EIS).
- Future Combat Systems Brigade Combat Team (FCS BCT).
- Ground Combat Systems (GCS).

We asked each of them the same five questions. The following are those questions and their answers:

Q: What is the biggest software acquisition challenge you are currently facing?

Ammunition: Our biggest challenge is to acquire and maintain (throughout the life cycle) safe, reliable, supportable, and modifiable systems that meet user requirements in an environment of rapid technological advances and complex regulations and policies which are, in many cases, overly broad. As an example, information assurance (IA)-related requirements are applicable equally to all systems (business, command and control, weapons in a tactical environment, etc.). However, due to the differing operational environments and system capabilities, the threats and vulnerabilities for business systems, command and control systems, and weapons systems are different, and the use of broad IA regulations and policies can create additional, and in many cases, unnecessary costs.

C3T: As the needs of our warfighters are rapidly evolving to address unique wartime challenges, the process for inserting software enhancements into Programs of Record (PORs) to satisfy these new requirements must be timely. In order to meet urgent needs, users will sometimes develop home-grown tools and software or contract developments that may not fully consider the implications of operating in a tactical environment. Any fielded solution needs to recognize unique tactical capability demands, such as the need for efficient use of limited tactical bandwidth, interoperability with Army-provided systems, and long-term sustainment, as would be required within the normal acquisition process. Our challenge is to immediately recognize these high-priority unit needs, fully understand and document the impacts, and drive the appropriate acquisition approvals, while retaining the warfighter's confidence that the process can respond with the right solution at the right time. Anything that can be done to make the acquisition process more timely and efficient contributes greatly to mission success.

EIS: Clearly our greatest challenge is helping to take the Army Business Mission Area (BMA) into net-centric operations and warfare. This starts with Army Business Transformation and the efforts of Mike Kirby, Deputy Under Secretary of the Army Business Transformation and the Lean Six Sigma program. It makes sense to spend the time necessary to lean out our business processes before we start buying solutions. As a matter of fact, the new solutions we need to be net-centric fall into the category of enterprise solutions. These solutions are different in that they are transformational in nature and present a whole constellation of issues we have not had to deal with in the past. Enterprise Resource Planning (ERP) and Service Oriented Architecture (SOA) are two examples. Both require a massive amount of hard work in the functional community before implementation of any software can be done effectively and efficiently. A lot of hard and expensive lessons have been learned in the private sector with the use of transformational Information Technology (IT). We do not want to miss any of these lessons as we build out the BMA. We have noted that most of the failures here have little or nothing to do with technology. The failures involve change management, governance and policy, and decision making, as well as other things we have not really dealt with before. For example, in an SOA environment, it is not about the technology as much as it is about the way you do business and how you manage the technology. This can be a monumental change for any organization, and an absolute if we are going to be net-centric.

FCS BCT: Our biggest challenge is the execution of the FCS BCT program. Technical complexity, distributed workforce, the use of commercial off-the-shelf (COTS) products, complex integration of software systems, and the long-term schedules required for ultra large software systems present a significant software acquisition challenge.

GCS: One of our biggest challenges is the synchronization of multiple sub-systems (including their support software) received from various contractors and other government agencies. The Software Blocking Initiative was supposed to ease this problem, but software synchronization remains a serious challenge.

Q: How is the GWOT affecting your organization?

Ammunition: The GWOT has produced a great urgency to quickly deliver safe, reliable, quality systems that meet users' needs. It forces a focus on continual improvement aimed at increasing system operational effectiveness while reducing overall time to field. This is not a trivial endeavor given the increasing complexity of systems and software and the complicated regulations and policies that must be adhered to.

C3T: The GWOT began as our modernization efforts, initiated as part of Force XXI, were nearing fruition. It quickly became apparent that the digital battle command software tools that were part of that initiative would become a decisive element of the fight. In a brief and historically significant period, the Army went from a small group of select units that experimented with digitization to a fully interoperable modular force operating digital command posts and related systems. For example, our Army in Iraq today operates from a *common operating picture* based on Army Battle Command System (ABCS) Version 6.4. That *common operating picture* is fed into our Command Post of the Future which allows geographically dispersed units to collaboratively visualize and plan the operational battlespace. Blue force tracking tracks and displays our platform locations in near real-time and the Joint Network Node connects our command posts using Internet Protocol-based satellite communications nodes. As the systems engineer involved with the technical challenges of integrating these C3T systems, it is hard to imagine another scenario that would have had more of an impact on how PEO C3T operates. The GWOT sharpened our focus on the task at-hand, direct support to the warfighter, while simultaneously driving groundbreaking work that transformed our tactical IT.

EIS: GWOT has refocused a great deal of resources and that means schedules slide to the right. We completely understand the constraints that everyone has to absorb with the current situation. The large enterprise systems acquisitions by their very nature are resource intensive.

FCS BCT: The GWOT serves to refine the picture of the future threat. This has highlighted the need to incrementally field capability to the current force to help prosecute the GWOT. Funding the GWOT has resulted in funding decrements to my organization.

GCS: The GWOT has had a significant impact on PEO GCS. Prior to the GWOT or Operation Iraqi Freedom (OIF), the Abrams tank program and Bradley vehicle program were downsizing (due to large funding cuts and natural attrition in personnel) in anticipation of new FCS vehicles that were on the drawing boards. Since the GWOT and OIF, billions of extra dollars have been pumped into Program Manager (PM) Heavy Brigade Combat Team to modernize these existing weapons platforms and enhance crew protection from enemy attacks. This has created some (temporary) acquisition and engineering staffing problems due to a shortage of experienced personnel (because of the prior downsizing/retirement of key, experienced personnel). We are coping, but everybody is extremely busy.

U.S. Army Organization Descriptions

The following organizations provided feedback to the questions submitted:

Ammunition <http://peoammo.army.mil>
The mission of the Ammunition PEO is to develop and procure conventional and leap-ahead munitions to increase combat power to warfighters. The PEO has been delegated as the Single Manager for Conventional Ammunition (SMCA) mission and therefore procures conventional ammunition items that have been transmitted to the SMCA for services.

Answers provided by Robin Gullifer, Program Executive Office Ammunition, Program Management; Heather Vimba, Program Executive Office, Chief Information Officer; John Scibilia Armament Research Development and Engineering Center Software Engineering Center.

Command, Control, and Communications Tactical (C3T)
<http://peoc3t.monmouth.army.mil>
The mission of the PEO C3T is to rapidly develop, field, and support leading edge, survivable, secure and interoperable tactical, theater and strategic command and control and communications systems through an iterative, spiral development process that results in the right systems, at the right time and at the best value to the warfighter. Today PEO C3T is involved in critical work supporting GWOT efforts through fielding situational awareness systems. These systems show a visual representation of friendly and enemy forces on computer screens inside vehicles and command posts and help to prevent fratricide or friendly fire incidents.

Answers provided by BG Nickolas G. Justice, Deputy Program Executive Officer C3T.

Enterprise Information Services (EIS) <www.eis.army.mil>
The mission of the EIS PEO is to provide joint service and Army warfighters with information dominance by developing, acquiring, integrating, deploying, and sustaining network-centric knowledge-based IT and business management systems, communications and infrastructure solutions through leveraged commercial and enterprise capabilities that support the total Army. This information dominance enables the Army to achieve victory. PEO EIS develops, acquires and deploys tactical and non-tactical IT systems and communications.

Answers provided by Dr. Chip Raymond, Director, Army Enterprise Solutions Competency Center.

Future Combat Systems Brigade Combat Team (FCS BCT) <www.army.mil/fcs>
The primary mission of the PM FCS BCT is to develop, produce and field a fully capable and sustainable FCS BCT that is compliant with the Joint Requirements Oversight Council approved Operational Requirements Document by the year 2014. A key objective of the FCS Program is to successfully develop an integrated BCT that is net-centric, lightweight, overwhelmingly lethal, rapidly deployable, self-sustaining, and survivable. The FCS-equipped BCT will be enabled by a fully integrated network that will increase connectivity and intelligence sharing within combat formations, while providing unprecedented situational awareness to soldiers in the field.

Answers provided by Edgar L. Dalrymple, PM FCS BCT, Associate Director, Software and Distributed Systems.

Ground Combat Systems (GCS) *<www.peogcs.army.mil>*
The mission of GCS is to maintain a total Army perspective in managing the development, acquisition, testing, systems integration, product improvement, and fielding that places the best ground combat and support systems in the hands of our soldiers. They serve as the System of Systems Integrator of the GCS for the armed forces and lead the Army transformation toward future systems as they evolve to the objective force while maintaining a current combat ready force. Their Abrams tanks, Bradley Fighting Vehicles and Paladins provide battlefield superiority in Iraq . The Stryker family, the Joint Lightweight 155mm Howitzer and Unmanned Ground Vehicles are evolving toward the Stryker and Objective Forces.

Answers provided by Mike Olsem, Senior Systems Engineer, SAIC.

Q: How are open source software and open architectures influencing your acquisition efforts?

Ammunition: In order to reduce cost and effort for compatibility, we make extensive use of COTS products in our systems. We have only just begun to look more closely at open source software and open architectures to determine how they might fit within our acquisition of systems. The mission, safety, and IA critical nature of our systems weigh heavily in determining what COTS products and open source software and architectures may be appropriate to incorporate.

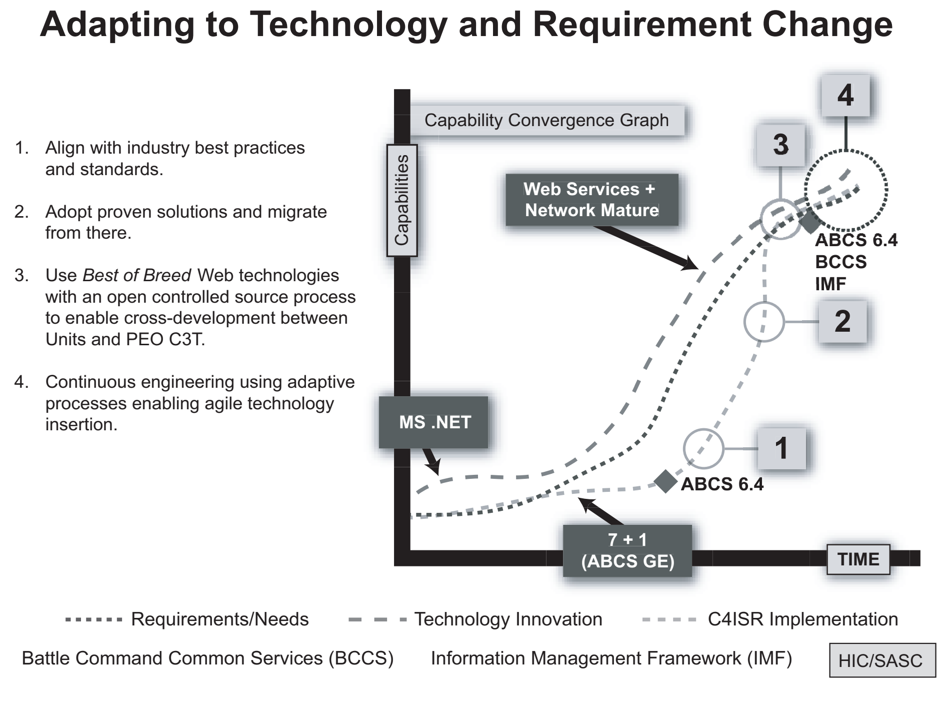
C3T: The fundamental concepts behind open source software and open architectures have driven our Battle Command software technical vision and associated acquisition efforts. As depicted in Figure 1, our original acquisition efforts focused on satisfying the critical subset of requirements for high intensity conflict. Building on this foundation, we opened up our architecture by implementing a common set of COTS/government off-the-shelf services across our tactical operation centers (Point 1 on the figure represents the Battle Command Common Services [BCSS] platform for distributing services to Battle Command users). We extended this service implementation by incorporating a community contribution model for the development of Web capabilities (Point 3 on the figure represents our Information Management [IM] Framework). By incorporating the open source model and an open architecture as depicted in the figure, we have improved our acquisition process by delivering warfighter-required capabilities and partnering with the user community in order to support requirements for full spectrum operations.

The important part of our IM Framework is that the applications and Web parts can then be managed and distributed to the community, letting the soldiers get back to doing their jobs and mitigating risk described in the first question. Units that rotate into an operational theater occasionally find out about some of the tools and technology developed in-theater that they *fall in on* only after they deploy. By adopting the IM Framework, we facilitate the timely distribution of capabilities across the Army so that the generating force can assess and exercise the capabilities being used by deployed units.

EIS: That depends. We would like to use more of these, but we also must keep in mind that security is a critical issue for us. We think that the Army will baseline on a federated architecture SOA. SOA is all about open architecture and the industry has set the basic standards needed to make this work well. We have security concerns, however, and will need to sort though that in the fullness of time. It would be nice to use open source stuff but we need to be cautious.

FCS BCT: The FCS BCT program developed its most foundational software component, Systems of Systems Common Operating Environment (SOSCOE), to follow the design principles of an open architecture. This has allowed the judicious selection and use of many COTS and open source software components. This has allowed accelerated development schedules and the potential migration of SOSCOE to other Army systems, representing an opportunity for increased interoperability and war-fighting capability.

Figure 1: *Satisfying a Critical Subset of Requirements*



Adapting to Technology and Requirement Change

1. Align with industry best practices and standards.
2. Adopt proven solutions and migrate from there.
3. Use *Best of Breed* Web technologies with an open controlled source process to enable cross-development between Units and PEO C3T.
4. Continuous engineering using adaptive processes enabling agile technology insertion.

Capability Convergence Graph
Capabilities
Web Services + Network Mature
4
3
ABCS 6.4
BCCS
IMF
2
MS .NET
1
ABCS 6.4
7 + 1
(ABCS GE)
TIME
Requirements/Needs
Technology Innovation
C4ISR Implementation
Battle Command Common Services (BCCS)
Information Management Framework (IMF)
HIC/SASC

GCS: No effect at all since we support highly customized weapons platforms with highly customized support software. FCS is more affected than we are as they plan future weapons systems since their stated goal is to make more usage of COTS and open architectures. But the Army must fight with what we have and our current weapons systems do not use open source or open architectures.

Q: What is your favorite government acquisition success story?

Ammunition: The PM for Intelligent Munitions System made a decision early in the contracting process to maintain a mirror software support environment at the Armament Software Engineering Center and to require periodic software drops so the Army software engineers could have better insight into the progress being made by the contractor, allow for the government to conduct independent testing of safety-critical software, and to ensure proper transition of the software from development to maintenance. This mirror lab is currently paying additional dividends. It will be used to speed up testing, thereby reducing overall schedule.

C3T: Adaptability. The urgent operational needs from our OIF and Operation Enduring Freedom commanders and the Army's conversion to a modular force structure meant transitioning developmental projects into widely fielded and fully supported systems in short order. Blue force tracking, the Joint Network Node, and Command Post of the Future are just a few examples of our recent successes in making that happen. Fundamentally, during wartime, we need to shift our mindset from a focus on ongoing development of our major PORs to a focus on how we can meet warfighter needs in time to make a positive difference.

EIS: Army General Fund Enterprise Business System (GFEBS). Here is an example of a transformational technology being applied in an effective and efficient way. Starting at the very top, the program has the complete, dedicated support of the functional business owner. This is one of those lessons learned. If the business owners do not support a transformational IT, it will fail. The Army financial business owners made the decision early on to put a lot of effort into the transformation. GFEBS is an ERP, which comes with a host of best business practices. This means change and change management. Although GFEBS is early in its acquisition cycle, it has all the hallmarks of a successful transformational technology implementation. Since there are no lessons learned with the second kick of a mule, we have learned well.

FCS BCT: SOSCOE Make/Buy. Per direction of the Assistant Secretary of the Army (Acquisition, Logistics & Technologies), the FCS BCT program instituted a comprehensive process to evaluate COTS products for purchase (buy), versus the development (make) of custom software products. The process is based on requirements-driven, markets surveys, and life-cycle cost/benefit analysis. There are multiple layers of management review that ensure effective technical and program management oversight. To date, this process has allowed SOSCOE to be developed from approximately 80 percent COTS software.

GCS: One of the best software success stories we have comes from the Abrams System Enhancement Program (SEP) tank. Abrams SEP Version 1 tank software (using circuit boards from contractor A) was quickly and easily ported to the Abrams SEP Version 2 tank (using circuit boards from contractor B). Using a software development approach known as *layering the software*, the porting of software between dissimilar circuit boards was made possible by isolating the software from the hardware dependencies.

Q: If you could make one change in the way the government procures software, what would it be? Why?

Ammunition: I believe we actually need to work on two changes:

1. I would consolidate and simplify regulations and policies with respect to *software* acquisition and recommend that Army PMs use a standard acquisition process model such as the Software Engineering Institute Capability Maturity Model Integration (CMMI®) acquisition model that is due to be released in 2007. The CMMI acquisition model or some reasonable alternative would ensure that acquisition best practices are used for procuring software-intensive weapon systems.
2. Ensure that software centers are involved from program start to ensure the RFP and Statement of Work properly considers software, data rights, and software supportability. Too often, data rights are not properly considered in the solicitation of a contract. Without data rights, the Army is accepting the risk of not being able to support its system if the contractor goes out of business, changes its business, or defaults.

   This over-reliance on a single contractor is not a good business practice and can lead to cost and schedule overruns or, at worst, the inability to maintain the system software.

C3T: It is hard to pick just one as we have learned so many lessons over the last 20+ years on how to do software better. So I will convey my top three interrelated changes.

1. Acquisition processes, to a large degree, are driven by principles established to acquire and manage risks associated with the acquisition of platforms/hardware. Programs are funded as new platforms with unique requirements to be tested pass/fail. The focus is on up-front risk, to get it right the first time, prior to making expensive production decisions. While a good model for platforms/hardware, generally for software this is not the best approach. And increasingly, more systems are becoming software-intensive, if not wholly software, with sought-after warfighter capabilities that are not necessarily new or unique, but evolved. Such capabilities would be better provided (in terms of cost, schedule, and risk) as integrated pieces of software – reused where possible. Software is really a continuous, evolutionary development that is not complete until a system is retired. And then, much of the software should be considered for reuse on the replacement system. To maximize effectiveness, the acquisition process (and life-cycle model) must be one where the Army can accept software as is, build on it incrementally over the life cycle, and do so in an agile manner.
2. We need to use a more holistic strategy (which, by the way, is not necessarily supported by current planning, programming, budgeting, and execution system and acquisition policies/processes). That is, more and more sought after capabilities, like net-centricity, are not systems but are concepts implemented through numerous technologies, systems, and supporting infrastructure. Similarly, related families of systems (system-of-systems) utilize many of the same/similar functions. However, we keep paying for the same/similar functions to be built over and over. By taking product line approaches and leveraging SOAs, we can build/buy once, centrally manage the software, and successfully reuse these software assets. These approaches, when implemented well, have proven track records in achieving the better, faster, cheaper objectives we espouse and can deliver significant increases in return on investment. This of course brings certain business challenges such as incentivizing industry to reuse rather than rebuild and would require procurement of certain essential government rights, source code, and supporting documentation from prime contractors.
3. We have to take life-cycle software management seriously (with a focus on the sustainment phase). In recent industry surveys, we have validated the fact that industry, having made significant investments in particular software systems, continuously evolves those systems through an aggressive software sustainment program, ensuring that a system continues to fulfill its needs over time, and eliminating need for unnecessary replacement (sustainment = maintenance + modernization). New capabilities (particularly software) can often be inserted into current systems faster, cheaper, and with less risk than procuring entirely new systems from scratch. System replacement (new development) carries a high risk in terms of time and cost. It is not unusual for a company to expend 80 percent of a software system's life-cycle cost in the sustainment phase. In the DoD, we typically see the reverse (80 percent through production and only 20 percent leveraging that capability investment). All of these lessons raise the fundamental question that must be addressed for each new capability: When should we procure new software versus evolve/reuse existing software? The answer to which has major implications.

EIS: A *software depot.* A consolidated, centralized *store* for Army software. One buyer, one seller. Software is one of those

® CMMI is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

Coming Events

June 11-14
DNCO 2007 Defense Network Centric Operations
Alexandria, VA
www.wbresearch.com/DNCO/index.asp

June 16-20
ICS 2007 21st Association for Computing Machinery International Conference on Supercomputing
Seattle, WA
http://ics07.ac.upc.edu/

June 18-21
2007 Systems and Software Technology Conference
Tampa Bay, FL
www.sstc-online.org

June 24-27
2007 ASEE American Society for Engineering Education Annual Conference and Exposition
Honolulu, HI
www.asee.org/conferences/annual/2007/index.cfm

June 25-28
Better Software Conference and Expo 2007
Las Vegas, NV
www.sqe.com/bettersoftwareconf/

June 25-28
SERP 2007 International Conference on Software Engineering Research and Practice
Las Vegas, NV
http://people.cs.und.edu/~reza/SERP07.html

***Coming Events:* Please submit coming events that are of interest to our readers at least 90 days before registration. E-mail announcements to: nicole.kentta@hill.af.mil.**

unique commodities that ought to be managed at the enterprise level. We know this and we are moving in that ultimate direction. Once we have the *depot* operational, we will have a reasonable chance to manage software like we do repair parts. There must be a lot of savings with that kind of approach.

To elaborate, there are a number of ongoing activities that aim at managing software at the enterprise level. We believe that business software (that is the functional, network, and enterprise software) and IT systems could be managed in the same way as we successfully manage our logistics base in the Army. The highest value target, for example, could be centralized license asset management. If I know where all the licenses are, know where they are needed, and know what is on the shelf (you cannot scan a network to locate these), then I think I could cross-level throughout the enterprise and drive down the total cost incurred when everyone buys their own licensed software. With the maturing capability of Web 2.0 and *software as a service*, we will one day be able look across the enterprise and see where our assets are being used and better manage them. That is a long way off. There are a lot of early efforts under way to do centralized management. In this era of constrained budgets, it might make sense to increase our focus in this area. It is just good business to do this and see how much money we can really save.

FCS BCT: The government, at least the Army, needs to stop buying software exclusively from the traditional defense contracting base. These companies have the overhead costs of manufacturing companies, yet software development should carry a far smaller overhead burden. Most defense contractors are still managed by manufacturing engineers or business managers. Very few of them have software management expertise. By using software-only suppliers who have relevant domain experience and lower-cost government labs, the cost of software can be reduced. This is especially true now that the hardware used by these systems is becoming more standard off-the-shelf types of technologies.

GCS: For our major weapons systems, we typically do not procure software. Instead, we procure systems and subsystems which contain software. However, we recognize that software is a critical component to the modern tanks, cannons, and troop transport vehicles (Bradley and Stryker). Thus, we would like more emphasis on a better, more formal, and documented process for integrating software upgrades into existing platforms (refer to the software synchronization problems in the first question).

Summary

As I considered the responses to the questions provided, I was struck by the contrast of diversity and similarities in the answers provided by the PEOs. For example, while addressing the *what is the biggest software acquisition challenge you are currently facing* question, the challenges mentioned ranged from technical challenges to business process issues and combinations of both. The criticism of the Software Blocking policy struck me because I have heard this criticism from multiple organizations.

In the second question *how is the GWOT affecting your organization*, one organization is delivering systems more quickly while another's schedules are sliding. One PEO is dealing with a decrease in funding while another is dealing with increased funding. Clearly, the GWOT is delivering a major impact on all of the organizations, and that impact appears to be dependent on how close the product – specific to each PEO – is to the fight. Easily assumed, all of the PEOs are busier due to the on-going GWOT.

Among the PEOs, there seems to be a growing acceptance to open source software and COTS products over time. While security still weighs into these decisions, organizations focusing on new acquisitions are considering the potential benefits of COTS and open source options more readily.

I was especially eager to read about the success stories from the PEOs. The stories discuss acquisition methodologies that look outside the box and have subsequently gotten greater value, through inventive means, for the taxpayer.

As we conclude with requests for change, most of the PEOs suggest ideas that will simplify and consolidate the acquisition process. Hopefully, by sharing this acquisition information, the PEOs' requests for change will be categorized beneath the success stories of the future.◆

References

1. Rogin, Josh. "Schoomaker: No Need to Trim FCS." Federal Computer Week Oct. 2006 <www.fcw.com/article96406-10-09-06-Web>.
2. Russell, Alec. "U.S. Army Chief Refuses to Submit Pentagon Budget." The New York Sun 26 Sept. 2006 <www.nysun.com/article/40306>.

Controlling Software Acquisition Costs With Function Points and Estimation Tools

Ian Brown
Booz Allen Hamilton

Too often, organizations that contract for software development services are at the mercy of vendors for cost and schedule estimates. Once a program office releases a request for proposal (RFP) for software development, it must somehow evaluate the validity of cost and schedule estimates that come back with the proposals. Or, a program might have a limited budget or schedule but not a clear understanding of what amount of development is actually feasible within these limitations. This article proposes an approach that can help buyers of software take control of this situation by providing the ability to objectively evaluate software development proposals, select the best value for their needs, and effectively manage acquisition costs from kick-off to product delivery.

Just a few years ago, purchasing a new car was often a lopsided affair. A buyer might know what kind of car he wanted to buy and how much he could afford, but the sellers held all the cards because they controlled the situation with information. They knew cost details – sticker price, invoice, incentives, kickback numbers – all of which provided them tremendous advantage in the transaction. They had information on how specific features were priced and which ones generated the most profit. A buyer might ask for power seats and windows, a CD player, antilock brakes, and passenger side air bags, and the salesperson might give a price of $5,000. How was the buyer to know whether or not that price was too high or if it was a great deal? And what if the seller offered to throw in the special undercoating and super-absorbent floor mats – which the buyer does not need – for *free*? It was very difficult for a buyer to understand if he was getting the features for which he asked and needed at a fair price. He might be able to shop around, but in the end, not having good information as a point of reference, it was difficult to assess if the transaction was fair.

These days, however, information is more readily available for car buyers. Car pricing internet sites have become valuable sources of information for consumers. Car buyers can now prepare more effectively for the acquisition process by arming themselves with independent, comparative cost information *before* the assessment and negotiation activities begin. Overall, consumers are much more likely to be able to buy the car of their choice – with the features wanted and needed – at a fair price.

In many ways, acquiring software or software development services compares to the *old way* of buying cars. Access to information is rarely equal, and it typically does not favor the buyer. Once an RFP for software development is released, a program office can become completely dependent upon vendors' estimates of cost and schedule. Or, a program might have a limited budget or schedule but not a clear understanding of what amount of development is actually feasible within these limitations. When assessing proposals from vendors, programs are faced with several of the following questions:

- Have we been offered a reasonable price?
- Has this project been deliberately underbid?
- Is the proposed schedule realistic?
- How do we know we are getting the functionality we have asked for and need?

With these kinds of uncertainties, how can a program make informed decisions when purchasing software or software development services? A program must take control of the situation to more effectively assess whether submitted proposals are realistic while having a clear understanding of what functionality should be included in the delivery.

The purpose of this article is to provide an approach that can help buyers of software objectively evaluate software development proposals, select the best value for their needs, and effectively manage acquisition costs from kickoff to product delivery. The foundation of this methodology is the ability to objectively size the developed software and to understand the potential ranges of cost and schedule that could result.

This article proposes a particular methodology to estimating software development cost and schedule *in the context of independent evaluation of vendor proposals*. It is not the only valid software estimation methodology available to organizations, but experience has shown that this specific methodology is very well-suited for this particular situation, for reasons that are discussed later.

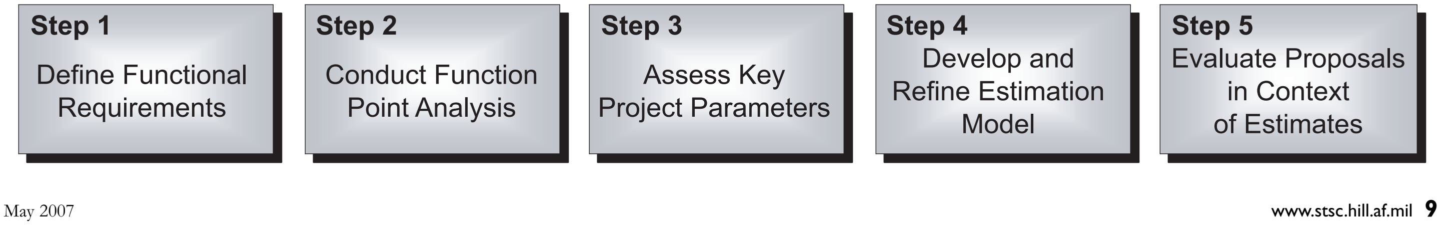
Methodology

The following five-step approach (Figure 1) is designed to be as objective as possible:

- **Step 1:** Define Functional Requirements.
- **Step 2:** Conduct Function Point Analysis (FPA).
- **Step 3:** Assess Key Project Parameters.
- **Step 4:** Develop and Refine Estimation Model.
- **Step 5:** Evaluate Proposals in Context of Estimates.

Generating cost and schedule estimates without intimate knowledge of a development organization's historical performance can be extremely challenging. This methodology combines standardized software measurement techniques with structured, well-documented estimation tools to enable true independent estimation. The goal is not to produce the *right* answers in terms of cost and schedule, but

Figure 1: *Software Acquisition Cost and Schedule Estimation Framework*


Step 1
Define Functional Requirements
Step 2
Conduct Function Point Analysis
Step 3
Assess Key Project Parameters
Step 4
Develop and Refine Estimation Model
Step 5
Evaluate Proposals in Context of Estimates

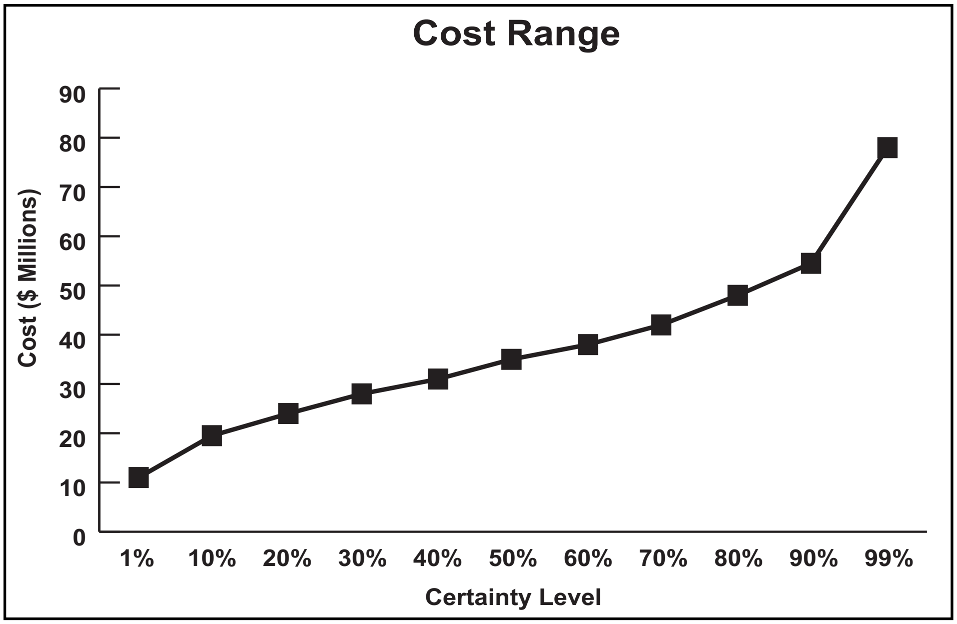


Figure 2: *Cost S Curve*

rather to understand what a reasonable range of answers might be and how vendor responses to a RFP (typically submitted as point estimates) fit within that range. Additionally, this methodology can help a program understand the relative cost and schedule risk it accepts by selecting one proposal over another.

Step 1 – Define Functional Requirements

Although this article focuses on software estimation for acquisition, requirements definition must be included as a critical step. Functional software requirements have to be defined, documented, and baselined. This is an essential foundation of the acquisition process. Requirements fidelity is key to ensuring that vendors understand what must be delivered. Requirements documentation should be provided to all potential bidders along with time for review and clarification. Requirements definition should happen before anything else. The importance of this step cannot be over-emphasized, as requirements are an integral part of nearly every aspect of delivery of the project, and they are necessary to conduct an initial FPA.

Step 2 – Conduct FPA

In order to estimate the cost and schedule of a development effort, one has to know the size of the intended software. Software size, too frequently overlooked in estimation exercises, is often expressed in source lines of code (SLOC). Many organizations have success with this size measure as a basis of estimate, but this measure has some inherent difficulties associated with it, especially in the context of the methodology proposed here [1]. Different organizations count SLOC differently – there are no industry-defined standards that identify what should be counted and what should not. This makes an accurate, independent assessment of SLOC size difficult to produce. So, although it is a valid measure of software size, SLOC does not lend itself to the nature of this particular type of estimate and independent analysis.

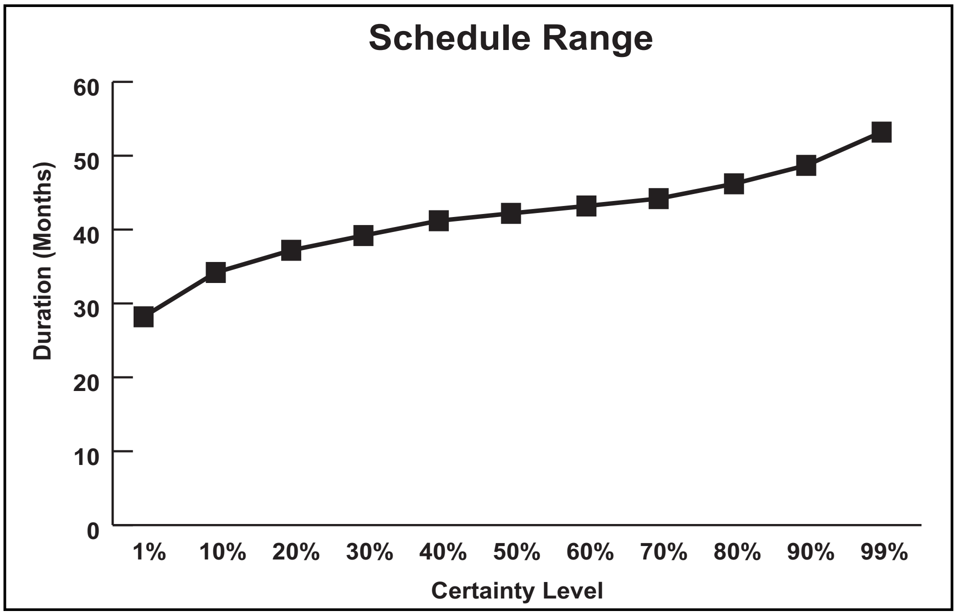
Function points, on the other hand, are a standardized unit of measure as recognized by the International Organization for Standardization 20926:2003. The function point standard is maintained by the International Function Point Users Group (IFPUG) in a voluminous counting practices manual; IFPUG offers a certification program that recognizes experts in the field as certified function point specialists. Function points measure software size independently of technology, platform, or programming language. In short, function points objectively define the size of an application that is to be developed based on defined functional requirements. They can also help identify gaps in requirements analysis, avoiding early introduction of defects [2].

To take this a step further, if an FPA is conducted prior to releasing the RFP, the results can be provided to all interested vendors to provide a common assumption of size so that all bidders can work from consistent information in developing their responses.

Step 3 – Assess Key Project Parameters

Key project parameters define characteristics of important cost and schedule drivers for the development effort. These parameters include high-level assumptions, such as the platform, programming languages, application type, reuse, development standards, commercial off-the-shelf (COTS) use, and staffing approach. If known, more precise parameter values for specific product and performance attributes can be identified in order to tailor the estimate to more specific project characteristics, such as development environment, personnel skills, organizational process maturity, security requirements, and system volatility. These inputs should be expressed as ranges (low, middle, high) to account for uncertainty and potential variation among bidding organizations. The less that is known about these more specific factors, the wider the range of assumptions should be.

Figure 3: *Schedule S Curve*



Step 4 – Develop and Refine Estimation Model

The outputs of steps 1 through 3 are relatively meaningless on their own. Only when they are combined as inputs to an estimation tool do they produce relevant, useful information. In this type of estimation exercise, leveraging a parametric tool to generate cost and schedule estimates is particularly important. Other estimation methodologies (analogy, wideband Delphi, cost estimating relationships, etc.) do not provide the flexibility needed to establish this proposal assessment framework. An estimation tool provides the flexibility to apply generalized assumptions where necessary and specific assumptions where appropriate. A trained, experienced user should be involved to make sure that the tool is used properly and the results are interpreted correctly. This methodology does not lend itself strictly to COTS acquisition, as most estimation tools are best suited for estimating effort on projects with custom development.

The model should be constructed with a work breakdown structure (WBS) that reflects the components of the software. For example, if the application will have a user interface with an Oracle back-end, then these two components should be modeled as separate WBS elements in the parametric tool. The WBS should also reflect any expected modular or incremental development strategies that might be proposed. Function points should then be allocated to the appropriate WBS elements or increments as accurately as possible.

Step 3 identified ranges of key project parameters. Applying these inputs to the estimation model will generate cost and schedule ranges with corresponding probabilities. Estimates in this form can be expressed as S graphs and are the linchpin of this evaluation structure. Using ranges in this way, as illustrated in Figures 2 and 3, provide the context and framework for evaluation of different bids from different vendors.

The estimates produced by the tool, however, should not just be accepted without consideration. Analysts should apply cross-checks, known analogies, or expert opinions to test the outputs for reasonability. The estimates should also be compared to the expected or known budget or schedule for the project. Ideally, those numbers will fall somewhere within the estimated ranges, preferably in the higher certainty levels. However, if the budget or schedule falls outside of the relevant range, the program office should review the model and key project parameters. Oftentimes this exercise can highlight some assumptions that might be incorrect. For example, the initial model may have assumed the project would be staffed to minimize the development schedule. This staffing approach generally increases effort and cost (as well as the associated risk), so if the estimated range is too high for the known project budget, perhaps the project should be modeled to optimize the effort (resulting in fewer staff, lower cost, and longer schedule).

If this review *still* results in an estimated range that does not include the necessary budget or schedule, then the project scope must be evaluated. Too often, a program office will set a project up for failure by demanding that the full set of software requirements be developed within a budget or period of performance that is simply unrealistic and unattainable. This methodology can help to avoid these situations by raising a red flag early in the process. Requirements should be evaluated and prioritized, and then the overall scope of the proposal should be reduced or phased in such a way to more likely fit the required budget and schedule. The powerful combination of function points (linked directly to requirements) and a parametric estimation tool make these *what if* scenarios possible.

Step 5 – Evaluate Proposals in Context of Estimates

This is the *payoff* step. Steps 1 through 4 prepare a program office for proposal evaluation. The cost and schedule estimates represented by the S curve charts

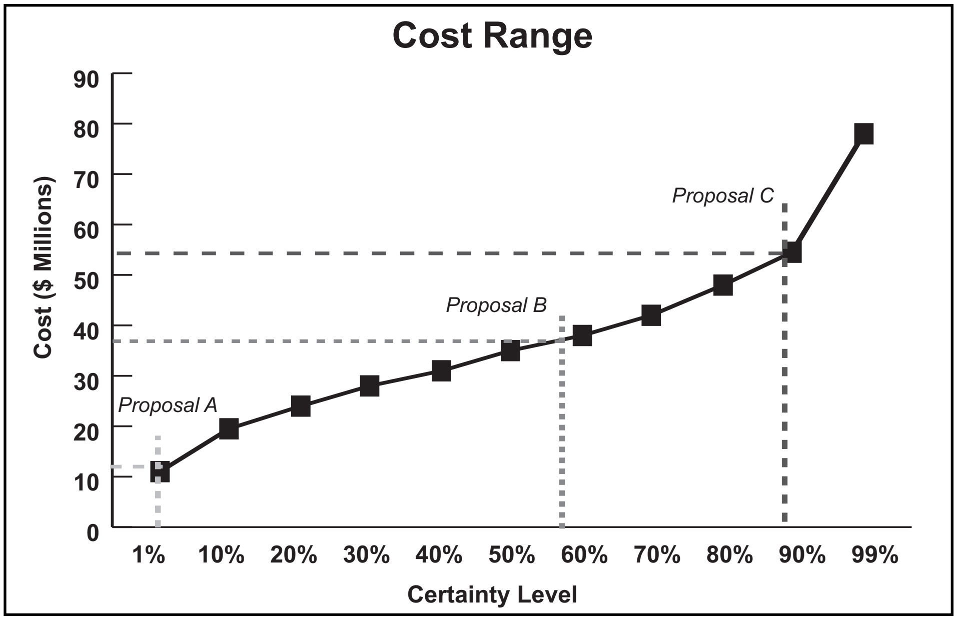
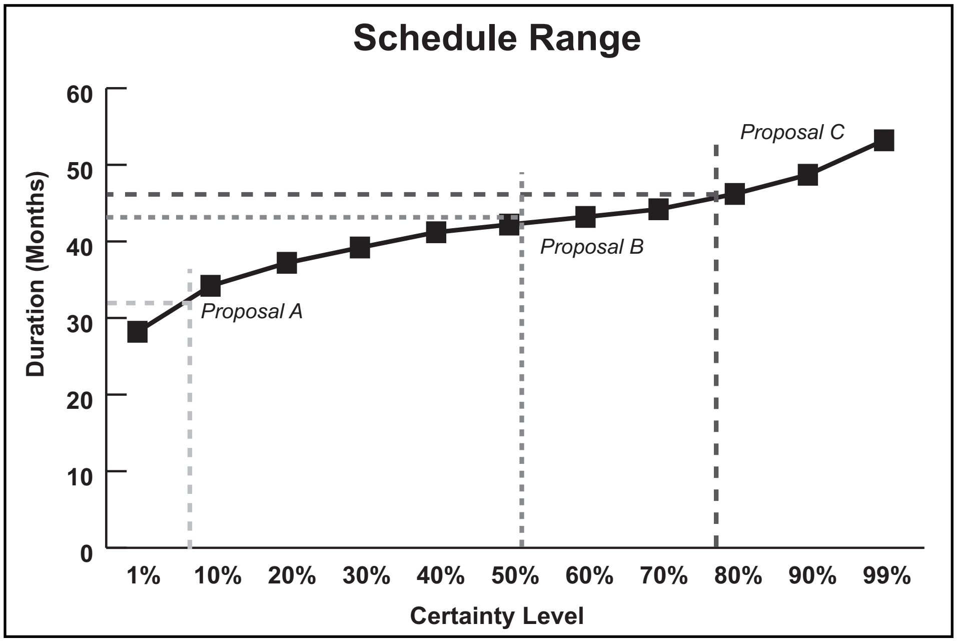


Figure 4: *Cost Evaluation Framework*

Figure 5: *Schedule Evaluation Framework*



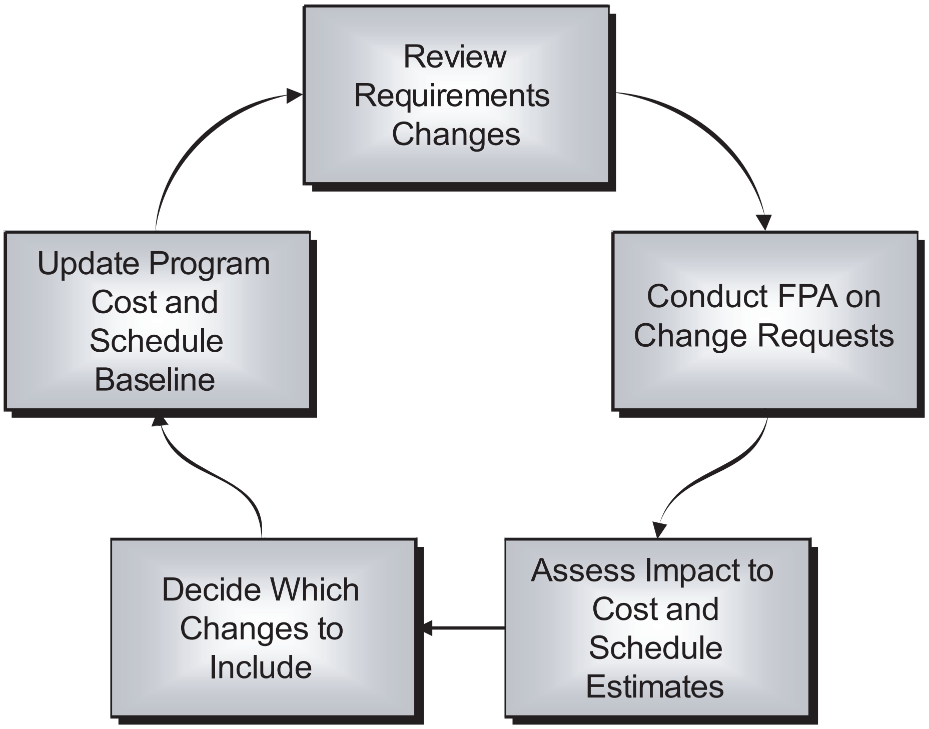

Review Requirements Changes
Conduct FPA on Change Requests
Assess Impact to Cost and Schedule Estimates
Decide Which Changes to Include
Update Program Cost and Schedule Baseline

Figure 6: *Iterative Cost Estimation Process*

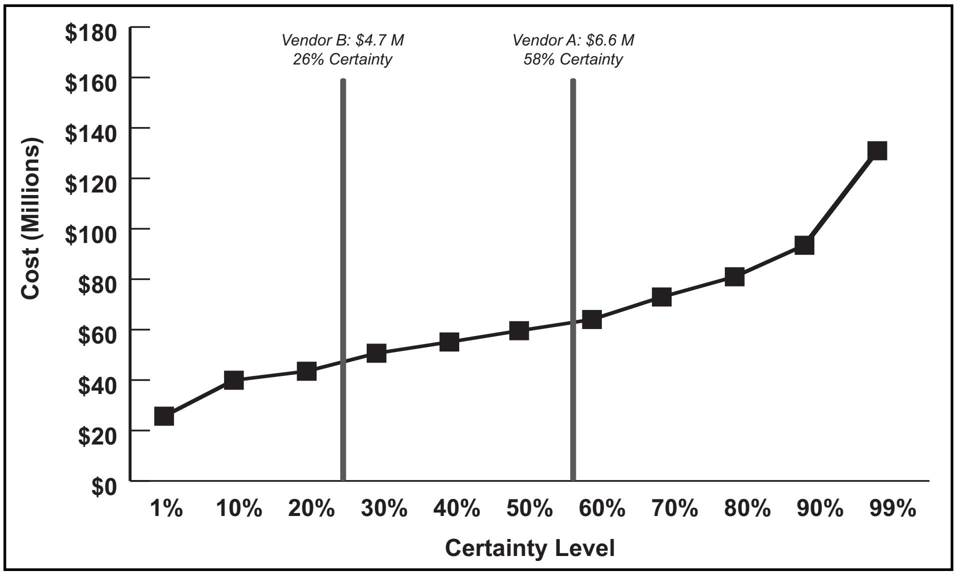
provide the framework against which actual proposals can be compared. When vendor responses are received, the cost and schedule proposed by each vendor can be mapped to a point on the independent cost and schedule curves (see Figures 4 and 5, page 11). Comparing certainty levels across the estimate continuum provides a more informed understanding of the relative value or risk of any given proposal.

Any given certainty level is interpreted as the likelihood that the project can be completed at or below that cost or schedule. The higher the certainty level, the more likely the project can be completed within that estimated cost and schedule. This comparative analysis can provide a tremendous amount of information to a program office. It can help identify *price to win* proposals that are overly optimistic as well as more conservative proposals that might be overpriced. This framework allows the program to control the amount of cost and schedule risk it accepts when awarding the work by providing a context to the winning bid that is based on robust quantitative analysis.

Other Critical Considerations

When evaluating the submitted proposals within this framework, there are several items that absolutely must be kept in mind. These critical considerations can significantly impact the value of the analysis and the quality of the information that results.

Figure 7: *Financial Management System Consolidation Cost Framework*



- **Width of Cost and Schedule Ranges.** These should increase with more uncertainty in an acquisition. How much is *really* known about the project should be carefully evaluated at any given point.
- **Contract Type.** The nature of the contract type may influence the proposals received in response to the RFP. Time and materials or cost plus award fee contracts are more likely to be priced aggressively (lower certainty levels) than fixed price proposals.
- **Cost Versus Effort.** Parametric tools actually calculate effort and then multiply effort by labor rates to arrive at cost estimates. Evaluating proposals based on effort estimates normalizes for differences in labor rates and highlights which vendor is actually proposing the most efficient solution. Note, however, that highly skilled and experienced vendors will likely have higher hourly rates.
- **COTS Usage/Software Reuse.** Some proposals may have assumed more software reuse or COTS applications than others. In this situation, the program office may want to run multiple scenarios with varying levels of COTS components or other software reuse assumptions. This produces multiple evaluation frameworks but allows for more appropriate apples-to-apples comparisons.

Post-Contract Award

The main purpose of this methodology is to enable more informed decisions when evaluating and awarding the initial development contract. The benefits, however, do not have to end there. The same estimation methodology can be applied in an iterative fashion throughout the entire contract life cycle to manage the project with quantitative data. The baseline FPA can serve as the basis of the initial contract, establishing threshold delivery rates or other relevant performance metrics. The contract could also implement a progressive fee structure for functionality added or changed in later development phases.

As part of a change control process, recurring FPA, coupled with updates to the estimation model, can evaluate the potential impact of proposed changes to functional or technical requirements on the project cost and schedule. The estimation process at this point should become cyclical in nature, reviewing change request, and revising size, cost, and schedule estimates based on the methodology (see Figure 6). *Go/no go* decisions regarding these changes can be based on quantitative analysis instead of guesswork. This approach is one way to help keep requirements volatility and *scope creep* under control. Finally, function points and a robust estimation model can provide the data essential to earned value management by establishing well-documented baseline cost and schedule plans, providing the ability to update these plans when requirements change, and effectively assessing *percent complete* or the value that has been delivered at any point in time.

Example

A client organization desired to merge multiple financial management systems with overlapping functionality into a single, consolidated system. The business owners developed a master set of requirements that any solution would have to meet, then released an RFP for open bids. Two vendors responded, both of whom assumed they would be able to leverage some amount of functionality from existing proprietary systems. Vendor A bid a significantly higher price than Vendor B, but the contracting organization was unsure which would provide the best value and was wary of the risk of focusing on the low bid.

Consultants completed an FPA of the master requirements to establish the baseline size, then conducted an independent gap analysis of each vendor's existing systems. Then, following the methodology set forth in this article, the consultants set up the RFP analysis framework that enabled the client to put each bid into context.

As the client suspected, Vendor B did offer a rather aggressive price bid, driven lower by an assumption that significant legacy reuse would be possible (see Figures 7 and 8). The probability of delivery for the bid amount was lower than desired (~26 percent certainty). Vendor B's schedule estimate was actually more conservative, but the client identified the lower cost estimate and the high reuse assumption as a potential risk for the project. Vendor A provided an estimate that fit into the evaluation framework at a more conservative level (~58 percent). Vendor A's schedule estimate was more aggressive than Vendor B's, but it was still within a reasonable range in the framework. Cost risk was more critical to the client organization than schedule risk. This approach allowed the client to make an informed decision to award the contract to Vendor A based on quantitative analysis. The client could justify the higher price bid while understanding where critical risk areas were in the acquisition strategy.

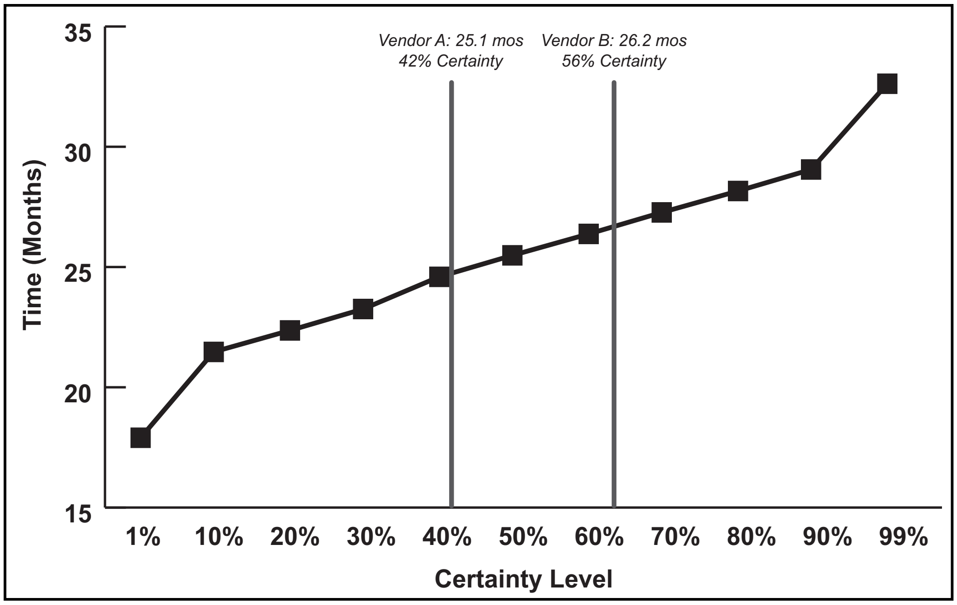


Figure 8: *Financial Management System Consolidation Schedule Framework*

Summary

The more information a program office has during the software acquisition process, the better the chances are for having the answers to the questions that the beginning of this article laid out. The reasonableness of the proposed price will be known, thanks to the context provided by the independent estimate. Proposals that are intentionally low-bid in order to win the contract can be identified and filtered prior to award. Conducting an FPA will help ensure completeness of requirements to improve the probability of delivery of full functionality. The methodology also allows a program office to conduct a *self-examination* to make sure that the planned budget and duration are reasonable and do not doom the project for cost and schedule overruns before it even starts. This methodology enables the contracting program office to more effectively control the balance of information and, in turn, produce acquisition results that benefit all stakeholders – program office, user community, and vendor alike.◆

References

1. Schofield, Joe. "The Statistically Unreliable Nature of Lines of Code." CrossTalk Apr. 2005 <www.stsc.hill.af.mil/crosstalk/2005/04.html>.
2. Dekkers, Carol, and Mauricio Aguiar. "Applying Function Point Analysis to Requirements Completeness." CrossTalk Feb. 2001 <www.stsc.hill.af.mil/crosstalk/2001/02.html>.

Additional Reading

1. Boehm, Barry W., Ellis Horowitz, Ray Madachy, and Donald Reifer. Software Cost Estimation With COCOMO II. Prentice Hall, 2000.
2. Galorath, Daniel D., and Michael W. Evans. Software Sizing, Estimation, and Risk Management. Auerbach, 2006.
3. IFPUG. IT Measurement: Practical Advice from the Experts. Addison-Wesley Professional, 2002.
4. Jones, Capers. Estimating Software Costs. McGraw-Hill, 1998.
5. McConnell, Steve. Software Estimation: Demystifying the Black Art. Microsoft Press, 2006.

About the Author

Ian Brown is a senior associate with Booz Allen Hamilton. With 10 years of experience in software measurement and analysis, he leads the firm's Quantitative Software Analysis capability. Brown is a member of the board of directors of the IFPUG, has spoken at several international software conferences, and is a Certified Function Point Specialist. Brown earned a bachelor's degree from Cornell University and a master's degree in public policy from Harvard University.

Booz Allen Hamilton
8283 Greensboro DR
Booz 2036
McLean, VA 22102
Phone: (703) 902-4971
Fax: (703) 902-3634
E-mail: brown_ian@bah.com

Software Assurance: Five Essential Considerations for Acquisition Officials[1]

Mary Linda Polydys
National Defense University

Stan Wisseman
Booz Allen Hamilton

Software Assurance (SwA) is a key element of national security; it is critical because dramatic increases in business and mission risks are attributable to exploitable software [1]. *A recent Chief Information Office (CIO) Executive Council poll indicated that the top two most important attributes of software are reliable software that functions as promised and software free from security vulnerabilities and malicious code. The acquisition process can be leveraged to achieve these important attributes. As part of the Department of Homeland Security (DHS) and Department of Defense (DoD) SwA initiative, a working group developed a guide,* Software Assurance in Acquisition: Mitigating Risks to the Enterprise *<https://buildsecurityin.us-cert.gov>, for acquisition officials on how to incorporate SwA considerations in key decisions throughout the acquisition process.*

Dependency on information technology (IT) makes SwA[2] [2] a key element of national security. IT in critical information infrastructures is composed of systems, system of systems, and family of systems (SoS/FoS). Most of these systems involve integrating a complex value chain of commercial off-the-shelf (COTS), government off-the-shelf (GOTS), open-source, embedded, and legacy software. Attackers exploit unintentional vulnerabilities or insert intentional vulnerabilities into these software components.

In a 2006 poll taken by the CIO Executive Council on the impact of software flaws, vulnerabilities, and malicious code, respondents indicated that the top two most important attributes of software are *reliable software that functions as promised* (95 percent of respondents) and *software free from security vulnerabilities and malicious code* (70 percent of respondents) [3].

SwA in the Acquisition Process

A broad range of stakeholders now need justifiable confidence that the software which enables their core business operations can be trusted to perform (even with attempted exploitation) and contribute to more resilient operations. In SoS/FoS, multiple software suppliers are usually involved. Therefore, the responsibility for SwA must now be shared by acquisition officials and supply chain constituents – building the assurance case starts with the acquisition process. To that end, acquisition officials[3] involved in the purchase of software services or products have a responsibility to factor in SwA to reduce the risk of exploitable software being passed to users.

However, there is a growing concern that acquisition officials are not aware of this responsibility and are not prepared to exercise SwA due diligence in the buying process. To assist acquisition officials in understanding and exercising SwA due diligence, a guide [4] was developed by a working group (as part of a larger SwA[4] initiative) on how to incorporate SwA considerations in key decisions throughout the acquisition process.

This article provides a summary of five essential SwA considerations that acquisition officials should include in their decision-making. These considerations are extracted or synthesized from the acquisition guide developed by the working group. The acquisition guide provides more detailed discussion and explanation along with additional considerations.

Five Essential SwA Considerations in Acquisition Decision-Making

SwA considerations should be included in each phase of the acquisition process from the initial acquisition strategy and plan, requirements development, contract or purchase, and contract administration through follow-on software support efforts. The objectives of these SwA considerations are to ensure the delivery of *reliable software that functions as promised* and *software free from security vulnerabilities and malicious code.*

Essential Consideration #1 – Build Security In: Create Acquisition Strategies and Plans That Include Essential SwA Considerations

To *build security in*, SwA considerations should be planned from the inception of a software or software-intensive system acquisition through delivery and post-release support. The Federal Acquisition Regulation (FAR) requires that an acquisition plan be developed for all acquisitions and that all plans discuss how agency information security requirements are being met [5]. The Defense Acquisition Guidebook requires program managers to develop an Acquisition Information Assurance (IA) Strategy as part of their Acquisition Strategy [6]. Whether developing a strategy or plan in accordance with the FAR, Defense Acquisition Guidebook, or another directive, SwA should be part of the discussion on how information security requirements are to be met. To that end, the strategies or plans might include a discussion on the participation of SwA subject matter experts in the acquisition process, initial SwA risk considerations, plans for including SwA requirements, SwA considerations in contractor selection, and SwA considerations in contract administration and project management.

Acquisition officials should require the participation of SwA subject matter experts in the acquisition process from planning, requirements development, source selection, contract award through contract administration, and project management. This is essential not only for establishing appropriate SwA requirements, but also in evaluating potential contractors and ensuring that secure software is delivered. Acquisition strategies and plans should state the level of SwA expertise required as well as specific statements of involvement. An example: This acquisition requires support from an SwA subject matter expert. This individual will develop the SwA requirements, evaluate the SwA aspect of proposals, and monitor the assurance case proving the delivery of SwA requirements during contract performance.

Strategies and plans should include an initial discussion on risk management. For information assurance/security, the security category (SC) (based on a range of risk levels) should be included in strategies and plans. The Federal Information Processing Standard Publication (FIPS Pub) 199 [7] as mandated by the Federal Information Security Management Act (FISMA) of 2002 requires that a security category be designated for each software-intensive system. The DoD Instruction (DoDI) 8500.2 [8] provides security categorization[5] rules for DoD software-intensive systems using Mission Assurance Categories (MAC) and confidentiality levels. The FIPS Pub 199 states that security categories should be based on the mission that the software is to support, the environment in which the mission is performed, and, generally, the kind of information that is generated and maintained to support the mission (e.g., medical, privacy, classified, time sensitive, warfighter combat information, financial, security management, etc.). Security categorization includes an assessment of three security objectives defined in FISMA: confidentiality[6], integrity[7], and availability[8] [9]. Two examples follow:

- **EXAMPLE 1 – From FIPS Pub 199:** A law enforcement organization managing extremely sensitive *investigative information* determines that the potential impact from a loss of confidentiality is high, the potential impact from a loss of integrity is moderate, and the potential impact from a loss of availability is moderate. The resulting SC of this information type is expressed as: SC investigative information = (*confidentiality*, HIGH), (*integrity*, MODERATE), (*availability*, MODERATE).
- **EXAMPLE 2 – [NOTIONAL], MAC, and Confidentiality Level:** A system must provide access to sensitive and classified combat support data. There must be uninterrupted service and data availability. The loss of confidentiality and integrity are unacceptable and could include the immediate and sustained loss of mission effectiveness. The resulting MAC and confidentiality level is expressed as: *Confidentiality*: TOP SECRET; *MAC I*: Requires the most stringent of protection measures.

Acquisition strategies and plans should include statements of critical, high-level SwA considerations. These high-level statements guide the ultimate detailed statement of requirements. Acquisition officials developing acquisition strategies and plans should rely heavily on the SwA personnel assigned to the acquisition. Three examples follow:

- **EXAMPLE 1 – COTS Software:** In order to ensure that COTS is consistent with the overall security requirements of the software-intensive system, SwA personnel assigned to this acquisition will provide requirements to ensure delivery of COTS that has specified pre-set security settings. In addition, requirements will mandate that testing of the specified pre-set software be accomplished on the operating system and platform proposed for production.
- **EXAMPLE 2 – Software Development or Systems Integration:** To manage the development and delivery of SwA requirements, an SwA case shall be developed that presents a convincing argument the software will operate in an acceptably secure manner. To support the SwA case, definitive evidence (e.g., processes, procedures, test results, etc.) shall be produced to present a convincing argument that the software will be acceptably secure throughout its life cycle, including termination. The security stakeholders (e.g., accreditors) will evaluate the SwA case in determining that the software will function as expected and be as free of vulnerabilities as possible.
- **EXAMPLE 3 – Generally:** The software shall address the required security properties and functionality, relevant laws, regulations, standards, and other legal and societal requirements. In addition, independent verification and validation (IV&V) shall be performed on the code to determine the software's security posture. This IV&V shall be performed by a qualified SwA IV&V entity.

High-level statements on how SwA is to be considered in the selection of contracts should also be included in acquisition strategies and plans. As an example: Due diligence questionnaires will be used to solicit answers from offerors on their SwA practices. The answers will be part of the evaluation plan.

Lastly, high-level statements should be included in acquisition strategies and plans on how SwA requirements are to be monitored during contract performance, for example: SwA personnel will monitor the delivery of SwA requirements.

Essential Consideration #2 – Require Secure Software: Include SwA Requirements in Software Requirements Document

The security category is the basis for SwA requirements. The FAR requires that federal agencies use FIPS pubs for IT standards and guidance [10]. The FIPS Pub 200 includes guidance on minimum security requirements for federal information and information systems [11]. The National Institute for Standards and Technology Special Publication (NIST SP 800-53) provides specific security control requirements based on security category [12], and the DoDI 8500.2 contains security control requirements based on mission assurance category for the DoD. The guide for acquisition officials includes additional sources for SwA requirements, as well as some examples. Table 1 shows examples of general requirements of SwA that acquisition officials should consider, including statements of work or terms and conditions. Table 2 shows specific requirements of SwA.

Essential Consideration #3 – Be an Educated Consumer: Ask the Right Questions During the Contracting Process

Knowing what to ask and asking the right questions regarding offerors' SwA environments is essential in determining how well offerors' meet business and technical goals for SwA. The guide for acquisition

Table 1: *Examples of General SwA Requirements*

General Requirements
• Definitions relative to SwA for a common understanding.
• A full explanation of the SC.
• Assurance case that addresses the SwA requirements (see more in Essential Consideration #4).
• SwA acceptance criteria (associated with the SwA case).
• SwA risk management that includes a formal program for risk management.
• Consideration for auditing code for security by an independent body.
• Software Architecture that includes SwA components.
• A security test plan that defines the approach for testing SwA requirements.
• Configuration guidelines for all security configuration options.
• Legal responsibilities relative to SwA.
• Qualifications and required SwA training of software personnel.
• Identification of key security personnel.
• Required information relative to foreign ownership, control, or influence.

officials includes sample software due-diligence questionnaires for various types (e.g., COTS only, software integration services, software development, etc.) of software acquisitions. These questionnaires provide the acquisition official a means to gather, in advance, some of the information needed to make a decision about whether it offers the process capabilities to deliver *reliable software that functions as promised* and *software free from security vulnerabilities and malicious code*.

Questionnaires may be used informally or incorporated into a formal process. For example: Informally, the buyer of COTS software may apply the questions to conduct market research into COTS products available to satisfy an organization's software requirements. Formally, the acquisition official may incorporate questions into a Request for Information or Request for Proposal (RFP) as part of a major software-intensive system acquisition. Answers to the questions form a basis for evaluating offers.

Questions in a software due diligence questionnaire may be organized into categories that represent a logical grouping of SwA concerns such as organization background, software production policies, software pedigree, assurance, preventive measures, quality control, operations and support, and service governance. Table 3 lists a partial set of example questions that may be used in the acquisition of custom software development services (the guide includes the comprehensive set of questions).

Essential Consideration #4 – Demand Delivery of Secure Software: Ensure SwA Requirements Are Met During Contract Administration and Project Management

Acquisition officials should ensure that all the SwA requirements are adequately monitored and implemented. This includes work plan management, assurance case management, software risk management, and final acceptance of the software product or service.

Acquisition officials must ensure that SwA requirements are specifically included in a contract work plan and/or work breakdown structure, if required. SwA subject matter experts should be used to ensure that SwA requirements are included in the work plan.

Table 2: *Examples of Specific SwA Requirements*

Specific Requirements
• A server-side software application shall never rely on the client to perform input validation. The server application should always validate any input it receives, regardless of whether that input was previously validated by the client.
• The software application shall verify that the actual results match the expected results criteria.
• The software application shall prevent any entity from performing application functions that entity's authorizations do not explicitly permit it to perform.
• Server/Web service that authenticates based on role or group authentication shall perform individual authentication first.
• Authentication technology shall be implemented based on published open standards.
• Code shall meet organizational and industry standards, conform to a consistent style guideline (code format), and shall be well documented.
• Appropriate security metrics shall be used during security review/audit in the software life cycle to measure the degree to which security criteria/requirements have or have not been satisfied.
• Security testing shall be performed both on individual units/components and on the whole integrated software application.
• Error messages shall not reveal more details than necessary about the software application.
• No software developer *backdoors*, debug interfaces, or unauthorized access paths shall be present in the production version of the software.
• After it goes into production, the software application's security posture shall be periodically reviewed to ensure that new vulnerabilities have not emerged.
• The software application shall continue functioning, possibly in a degraded mode, when subjected to input patterns that indicate a denial of service attack.
• Any COTS software shall be configured in accordance with security configurations specified in the statement of work. The contractor shall provide written assurance that the software operates as intended and as initially configured with each subsequent software release.

Acquisition officials must ensure that the SwA case is managed in accordance with the contract and should be managed as part of the acquisition risk management strategy. The development of an SwA case is an iterative process throughout a system's life cycle and contains a plethora of claims and evidence types not collated or contained together. Therefore, the SwA case must be developed and managed in such a fashion that all evidence is able to be preserved, traced, and accessed. Throughout the acquisition life cycle, SwA case reports – as stipulated in the contract – should be delivered at key project milestones. These reports should be reviewed by appropriate SwA subject matter experts for issues and recommendations. Acquisition officials must ensure that periodic reviews of the SwA case are transparent and any corrective actions are followed to a conclusion prior to acceptance of the case argument. Example issues related to SwA case management during contract performance include the following:

- **Performance.** Is the SwA case development progressing in accordance with contract requirements? Are project technical milestones incorporating SwA case review? Does the SwA case comply with contract requirements, including regulations and certification requirements?
- **Resources.** Has the contractor allocated appropriate, qualified personnel to the task? Is the SwA case being developed with appropriate tools? Is the SwA case budget realistic?
- **Quality.** Is the supplier engaging the right acquisition officials to review the acceptability of the SwA case? Are corrective actions being followed up adequately? Are the contractor's claims, arguments, and evidence sufficiently robust and commensurate with risk?
- **Time.** Is the SwA case development on schedule and fully integrated with software system development?

Final acceptance should be based on the acceptance of the final SwA case. Criteria for acceptance should be explicit and included in the SwA case.

Essential Consideration #5 – Continue SwA for the Life of the Software: Maintain SwA in Follow-On Support

Follow-on support is the logistics tail in the acquisition of software. Additional contracts are often awarded to provide support during this phase. There should be ongoing analyses to ensure that security requirements remain adequate. To that end, acquisition officials should ensure that the assurance/security requirements implemented and accepted in previous contracts flow to the follow-on contract efforts. Additionally, acquisition officials should ensure that contract language is in place to guide the transition process from an incumbent contractor to a new contractor responsible for follow-on support.

Information systems are typically in a constant state of migration with upgrades to hardware, software, or firmware and possible modifications to the surrounding environment of the system. Weak change/configuration control procedures can corrupt software and introduce new security vulnerabilities. Therefore, acquisition officials should ensure that strong change/configuration control flows to follow-on contract efforts.

Patches and upgrades make direct changes to software and potentially the configuration of the operating system to which they are applied. Changes may degrade performance, introduce new vulnerabilities, or reintroduce old vulnerabilities. In order to understand patch risks, the patch process must be examined in some detail during the initial acquisition and again when follow-on support contracts are awarded. One of the most common patch failures stems from a lack of encryption and authentication in the implementation phase. Suppliers should provide updates in a secure fashion. There should be no doubt that the source is legitimate and the update's integrity is maintained in transit.

Partial Set of Example Questions

- How does your company use security best practices that are designed to address security concerns in the software development life cycle (SDLC)?
- Are there formal software quality policies in place? How are they enforced?
- What measurement practices and data does your company use to enable realistic project planning, timely monitoring of project progress and status, identification of project risks, and effective process improvement?
- What training does your company offer related to defining security requirements, secure architecture and design, secure coding practices, and security testing?
- Describe the company's policy and process for professional certification of developers and ensuring certifications are valid and up-to-date.
- Are security requirements developed independently of the rest of the requirements engineering activities, or are they integrated into the mainstream requirements activities? Explain.
- What threat modeling process, if any, is used when designing the software? What analysis, design, and construction tools are used by your software design teams? What security design and security architecture artifacts are produced? How are they maintained?
- Does the software development plan include peer reviews for quality and security?
- Are tools provided to help developers verify that the software they have produced is free of defects that could lead to vulnerabilities? What are they?
- Explain how your company ensures that software is able to detect, recognize, and respond to attack patterns in input it receives from human users and external processes?
- Are static or dynamic software security analysis tools used to identify vulnerabilities in the software? If yes, what tools are used? What classes of vulnerabilities are covered?
- Are security-specific regression tests performed during the development process? How broad is the test coverage? How frequently are security-specific regression tests performed?
- What policies and processes does your organization use to verify that software components do not contain unintended, *dead*, or malicious code? What tools are used?
- Does your company perform background checks on members of the software development team? If so, are there any additional *vetting* checks done on people who work on critical application components, such as security? Explain.
- Does your company have formally defined security policies associated with clearly defined roles and responsibilities for personnel working within the SDLC, including work that is subcontracted or outsourced, along with management oversight and enforcement? Explain.

Table 3: *Partial Set of Questions for Custom Software Development Services*

Conclusion

Large numbers of vulnerable software-based systems exist today, in many cases due to the acquisition of vulnerable software. The rampant, worldwide increase in exploitation of software vulnerabilities demands that acquisition officials not only check for acceptable functionality, but also achieve acceptable SwA. Security cannot be *bolted on* after software services and products are delivered. To that end, acquisition officials must become educated consumers in the purchase of secure software, and each phase of the acquisition process must be leveraged to *build security in* to ensure the delivery of *reliable software that functions as promised* and *software free from security vulnerabilities and malicious code.*◆

References

1. President's Information Technology Advisory Committee. Cyber Security: A Crisis of Prioritization. Arlington, VA: National Coordination Office for Information Technology Research and Development, Feb. 2005 <www.nitrd.gov/pitac/reports/20050301_cybersecurity/cybersecurity.pdf>.
2. U.S. Committee on National Security Systems (CNSS). CNSS Instruction No. 4009, National Information Assurance Glossary. Fort Meade, MD: CNSS, 2006 <www.cnss.gov/Assets/pdf/cn ssi_4009.pdf>.
3. CIO Executive Council. New CIO Executive Council Poll Reveals CIOs Lack Confidence in Software. CIO Executive Council News Bureau, 2006 <www.cioexecutivecouncil.com/nb/>.
4. DoD and D HS. SwA Working Group. Software Assurance in Acqui-sition: Mitigating Risks to the Enter-prise (V1.0). Washington: DoD/ DHS, Mar. 2007 <https://buildsecurityin us-cert.gov>.
5. FAR. Part 7, Acquisition Planning, Subpart 7.105(b)(17). Washington: General Services Administration, DoD, and National Aeronautics and Space Administration, 2005 <www.arnet.gov/far/>.
6. DoD. Defense Acquisition Guidebook. Part 2.3, Systems Acquisition: Acquisition Strategy <http://akss.dau.mil/dag/DoD5000.asp?view=document>.
7. U.S. Department of Commerce. NIST. FIPS Pub 199, Standards for Security Categorization of Federal Information and Information Systems. Gaithersburg, MD: NIST, 2004 <http://csrc.nist.gov/publications/fips/index.html>.
8. DoD. DoD Instruction 8500.2, Information Assurance Implementation. E4, Baseline Information Assurance Levels. Washington: DoD,

2003 <www.dtic.mil/whs/directives/corres/ins1.html>.

9. FISMA of 2002. 44 U.S.C., Sec 3532 <www.access.gpo.gov/uscode/title44/chapter35_subchapterii_.html>.
10. FAR. Subpart 11: Selecting and Developing Requirements Documents, Subpart 11.102: Standardization Program. Washington: GSA, DoD, and NASA, 2005 <http://www.arnet.gov/far/>.
11. Department of Commerce. NIST. FIPS Pub 200, Minimum Security Requirements for Federal Information and Information Systems. Gaithersburg, MD: NIST, 2006 <http://csrc.nist.gov/publications/fips/index.html>.
12. Department of Commerce. NIST. NIST SP 800-53, Rev 1, Recommended Security Controls for Federal Information Systems. Gaithersburg, MD: NIST, 2006 (Final Public Draft) <http://csrc.nist.gov/publications/nistpubs/index.html>.

Notes

1. The views expressed in this article are those of the authors and do not reflect the official policy or position of the National Defense University, the DoD, or the U.S. government.
2. SwA is the level of confidence that software is free from vulnerabilities, either intentionally designed into the software or accidentally inserted at anytime during its life cycle, and the software functions in the intended manner (CNSS Instruction No. 4009).
3. The generic term *acquisition official* is used throughout this article to mean the contracting officers or purchasing officials and other members of the purchasing team. Members of the purchasing team may include personnel who develop requirements and statements of work for contracts, contracting officer representatives to include contracting officer technical representatives, or program/project managers.
4. In 2003, the DoD launched an SwA initiative led by Joe Jarzombek. This was joined in 2004 by the DHS to address concerns of poor-quality, unreliable, and non-secure software. In March 2005, Jarzombek moved to DHS as the Director for SwA, National Cyber Security Division (NCSD). He provides the leadership in the collaborative SwA efforts. Several working groups (with members across government agencies, industry, and academia) were established. The initial working groups for DHS including the following:
   - Software technology, tools, and product evaluation.
   - Software acquisition.
   - Software processes and practices.
   - Software workforce educational and training.

   The goal of the SwA Acquisition working group is to look at how to enhance software supply chain management through improved risk mitigation and contracting for secure software. The overwhelming recommendation of the group is the development of a guide that provides due-diligence questionnaires, sample templates, and sample language that could be used in statements of work, RFPs, and contracts.
5. The FISMA of 2002 requires the development of security categorization standards. The security categories are the basis for establishing information security requirements based on a range of risk levels. See FIPS Pub 199 for security categorization of information and information systems that form a basis for confidentiality, availability, and integrity requirements. Also see DoD 8500 policies regarding security categorization-mission assurance categories. The DoD has three defined mission assurance categories that form the basis for availability and integrity requirements. Confidentiality requirements are based on the security classification of information.
6. ... *preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information* [44 U.S.C., Sec. 3532]. A loss of *confidentiality* is the unauthorized disclosure of information.
7. … *guarding against improper information modification or destruction, and includes ensuring information non-repudiation and authenticity* [44 U.S.C., Sec. 3532]. A loss of *integrity* is the unauthorized modification or destruction of information.
8. … *ensuring timely and reliable access to and use of information* [44 U.S.C., Sec. 3532]. A loss of *availability* is the disruption of access to or use of the software-intensive system.

About the Authors

Mary Linda Polydys is currently the Department Chair of the Information Operations and Assurance Department, Information Resources Management, National Defense University and co-chairs the DHS NCSD SwA working group. For more than 33 years, she has provided the U.S. government expert services in information security and assurance education, information technology acquisition and project management, enterprise architecture, and data management. Polydys has a bachelor's degree in decision sciences and a master's degree in information systems from George Mason University.

Information Resources
Management College
National Defense University
FT Lesley J. McNair,
Marshall Hall
Washington, D.C. 20319
Phone: (202) 685-3889
Fax: (202) 685-3974
E-mail: polydysm@ndu.edu

Stan Wisseman is a senior associate at Booz Allen Hamilton and has 22 years of experience in the IA field. He currently co-chairs the DHS NCSD SwA Acquisition working group, leads the IA team for the U.S. Department of Transportation's Vehicle Infrastructure Integration project, and oversees a SwA practice. Wisseman holds Certified Information Systems Security Professional, Certified Information Security Manager, and Project Management Professional (PMP) certifications. He has a bachelor's degree in computer science from Texas A&M University and a master's degree in engineering management from Santa Clara University.

Booz Allen Hamilton
8251 Greensboro DR
McLean, VA 22102
Phone: (703) 902-4673
Fax: (703) 902-3281
E-mail: wisseman_stan@bah.com

Web Sites

Defense Acquisition Performance Assessment Project

www.acq.osd.mil/dapaproject/

The Defense Acquisition Performance Assessment project will provide the Secretary of Defense and the 2006 Quadrennial Defense Review with recommendations on how the DoD can improve the performance of the defense acquisition system for major programs, and a comprehensive and executable implementation plan to institutionalize these recommendations. The site provides links to numerous documents and information pertaining to acquisition improvements for the global warfighter.

Defense Acquisition Guidebook

http://akss.dau.mil/dag

The *Guidebook*'s purpose is to provide the acquisition community and industry partners with an interactive, online reference to policy, and discretionary best practices. Consider the *Guidebook* a valuable resource when planning your programs.

Defense Acquisition History Project

www.army.mil/cmh/acquisition/index.html

Initiated by the Historical Office, Office of the Secretary of Defense, the Acquisition History Project is a multi-year effort that will produce a six-volume history of defense acquisition from the end of World War II to the present. The volumes will focus on OSD-level policy direction and service-level execution of defense acquisition. Five historical volumes, covering the history of defense acquisition from 1945-2000, are scheduled for publication in late 2007. The sixth volume, an annotated collection of key primary documents related to the history of defense acquisition, will be released in 2008. Updated information about the ongoing project can be found on this Web site.

The NASA Goddard Space Flight Center (GSFC) SwA

http://sw-assurance.gsfc.nasa.gov

The NASA GSFC SwA Web site provides tools, procedures, and training materials for software and safety assurance personnel, software engineers, as well as program and project managers. Practitioner assets can be found for each of the five Software Assurance disciplines.

Software and Systems Process Improvements Networks (SPIN)

www.sei.cmu.edu/collaborating/spins

A SPIN is an organization of professionals in a given geographical area who are interested in software and systems process improvement. It is a practical forum for the interchange of ideas, information, and mutual support.

The Software Technology Support Center Technical Document Resource

www.stsc.hill.af.mil/resources/tech_docs/

An invaluable section of our Web site, this is a repository of guidelines, reports, and templates published over the last several years to help you succeed in defense software engineering. The U.S. Air Force's Software Technology Support Center is excited to provide an updated and condensed version of the Guidelines for Successful Acquisition and Management of Software Intensive Systems. These guidelines can be found by following the links on the above listed Web address along with prior editions. The goal of this project has been to provide a usable desk reference that would give a brief but effective overview of important software acquisition and development topics, provide checklists for rapid self-inspection, and provide pointers to additional information on the topics covered.

Acquisition Community Connection

https://acc.dau.mil/CommunityBrowser.aspx?id=25749

The Acquisition Community Connection Web site is a valuable forum created to support growth beyond internal resources. In an attempt to reduce cost for the DoD, the forum creates an opportunity for those within the acquisition industry to come together, share ideas, and discuss best practices and lessons learned. The forum is available 24/7 and much of the content is accessible without login, giving timely advice and collaborative ideas across organizational boundaries.

CrossTalk
The Journal of Defense Software Engineering

Get Your Free Subscription

Fill out and send us this form.

517 SMXS/MXDEA
6022 Fir Ave
Bldg 1238
Hill AFB, UT 84056-5820
Fax: (801) 777-8069 DSN: 777-8069
Phone: (801) 775-5555 DSN: 775-5555

Or request online at www.stsc.hill.af.mil

Name:____________________

Rank/Grade:____________________

Position/Title:____________________

Organization:____________________

Address:____________________

____

Base/City:____________________

State:__________Zip:__________

Phone:(____)____________________

Fax:(____)____________________

E-mail:____________________

Check Box(es) To Request Back Issues:

Jan2006	□	Communication
Feb2006	□	New Twist on Technology
Mar2006	□	PSP/TSP
Apr2006	□	CMMI
May2006	□	Transforming
June2006	□	Why Projects Fail
July2006	□	Net-Centricity
Aug2006	□	Ada 2005
Sept2006	□	Software Assurance
Oct2006	□	Star Wars to Star Trek
Nov2006	□	Management Basics
Dec2006	□	Requirements Eng.
Jan2007	□	Publisher's Choice
Feb2007	□	CMMI
Mar2007	□	Software Security
Apr2007	□	Agile Development

To Request Back Issues on Topics Not Listed Above, Please Contact <stsc.customerservice@hill.af.mil>.

The Acquisition of Joint Programs: The Implications of Interdependencies

Dr. Mary Maureen Brown
University of North Carolina at Chapel Hill

Robert M. Flowe
Office of the Secretary of Defense

Sean Patrick Hamel
Software Engineering Institute

The movement toward transformation and joint capabilities has created new challenges for program acquisition efforts. The research reported in this article examines the implications of joint capabilities on acquisition. In short, the research investigates how program interdependency, size, age, and developmental status influence the occurrence of programmatic breaches. The findings provide empirical evidence that metrics that are capable of measuring interdependency may prove fruitful as an early indicator of joint program acquisition shortfalls.

With weapon system investments expecting to top the $1 trillion mark for the 2003-2009 time period, unprecedented attention has been devoted to clarifying the determinants of program risk, failure, and success [1]. The difficulties associated with averting and predicting adverse program outcomes such as cost and schedule breaches is not only a source of external criticism [2] and internal attention [3], it has illuminated deficiencies in current practices of program management and oversight. To date, there is significant debate regarding the factors that influence the outcomes of Department of Defense (DoD) programs. As such, the root causes of cost growth, schedule delay, and poor performance have received increased attention over the years [4].

Adding fuel to the fire is the fact that DoD acquisition investments are increasingly concentrated in very large, complex *system of systems* and net-centric designs. Despite the ongoing acquisition difficulties (that stymied past) and current efforts, DoD is in the process of radical transformation to a world predicated on joint capabilities – thereby leaving managers scrambling to identify new processes and metrics to support the new joint acquisition paradigm. In the case of the transformation, acquisition goals are shifting from individualized single system solutions to universal solutions that service joint needs. As such, many changes are under way in the acquisition arena and the search is on for a clearer understanding of how various acquisition strategies either support or impede joint efforts.

It comes as little surprise that the acquisition of joint programs is considerably more difficult than those of single-service programs. The reasons for the increased difficulties are often attributed to diverse requirements and complex management structures. As a consequence, joint programs are often criticized for taking longer and costing more than single service acquisitions. Some argue that joint (or interdependent) programs are not unlike single system initiatives – not differing in any important respects. The difference is simply one of scale; long-standing programmatic activities remain salient. Others argue that interdependent programs differ from single system efforts in fundamental ways that demand unique programmatic strategies and methods.

The research described in this article seeks to shed light on the controversy by asking if *jointness* matters. The following discussion focuses on the results of a cross comparison of single and joint acquisition efforts. The research stems from the perceived need to improve the ability to accurately gauge the cost and schedule demands of joint efforts. The overall goal of the research was to empirically test whether joint acquisition efforts encountered greater difficulties than their single system counterparts – and if so, to shed light on the nature of those difficulties. As discussed further on, the research examined 84 Acquisition Category (ACAT) 1[1] weapon system programs that were under development during the 1997-2005 time period. This research hopes to contribute to an understanding of the underlying causal factors that challenge joint efforts for the purpose of finding strategies that can enhance the success rate of joint capabilities.

Table 1: *Variables and Definitions*

Variable	Definition
Program Status	Indicates whether the program is single or joint.
Program Size	Total program cost in constant dollars.
Program Stage	Current stage of the program in terms of development or production.
Program Age	Years since entering Milestone B[3] status.
Schedule Breach	A program receives a schedule breach when the schedule exceeds most recent APB[4] schedule estimate by six months.
RDT&E Breach	A program receives an RDT&E breach when the research, development, testing, and evaluation costs exceed 15 percent.
Other Breach	Summation of the number of program acquisition unit cost, average procurement unit cost, procurement, and Nunn-McCurdy Breaches.

Understanding *Jointness*: A Closer Look at Interdependence

The desire for joint capabilities mandates, by definition, the establishment of interdependencies. Interdependent activities are not new to DoD or to government in general. However, what is new is the scale to which interdependent actions are currently applied. For most organizations, interdependence is pursued as a means to leverage the collective assets of various organizations located at different points along the value chain. In the DoD arena, joint capabilities are actualized by establishing interoperable systems. And the efforts promise to offer significant benefits. For example, in the command and control process, military operations benefit when commanders can seek, synthesize, and disseminate several types of information that derive from different organizations. Experts in a variety of areas must collaborate to effectively create and execute battle plans. These experts may come from different disciplines (or specialties), different branches of the military, or even different countries. In short, joint capabilities are achieved through the interoperable systems that allow interdependent activities to occur.

Interdependency is typically defined

as the degree to which the performance of one activity (or system) relies on an external activity (or system) for its accomplishment [5]. Interdependencies can often take several forms. Most frequently they will be in the form of technical interfaces but they can also be financial, materiel, or task-based. And they do incur a cost that is not present in non-interdependent efforts [6]. Transaction costs are the costs that arise from the establishment and maintenance of interdependencies [7]. They are the costs associated with conducting the transactions that allow the transfer of data, capital, or labor, and they are often manifested in the form of labor costs and tend to be distributed across all labor categories. The search for joint solutions, the costs of bargaining and negotiation, and the ongoing costs of monitoring and enforcing the agreements of the interdependent activities are all manifestations of transaction costs.

To date, the central question that received scholarly and practitioner attention was not *what will a given interdependency cost?* but rather *under what situations will (or should) organizations incur the costs of interdependencies?* The failure to be concerned about the true nature of the costs was largely due to the presupposition that organizations would not engage in interdependent actions if the cost outweighed the benefits. This presupposition works well when organizations have the benefit of choice. However, in the government sector, and in the DoD's case specifically, legislative requirements eliminate the opportunity to choose the most efficient path. Government agencies are often asked to incur the costs of interdependent activities in return for the benefits of synergy. Little is actually known about how to estimate the cost, schedule, and risk of interdependent activities. The lack of metrics and techniques for gauging interdependencies (and their associated transaction costs) may prove especially problematic in light of the scale of the interdependent activities that are currently under way. Whether long-standing, single-system driven methods for estimating acquisition cost, schedule or risk remain salient predictors is a topic of much debate. Thus, the study of whether interdependent actions demand unique methods and metrics is an important, albeit over-looked, consideration.

The Research Study

To test whether single-system efforts differed from their joint counterparts, we examined 84 active DoD weapon system programs in terms of the number and type of programmatic breaches they encountered. In short, we examined the programs on the occurrence of schedule breach and RDT&E cost breach. We restricted the analysis to the study of programmatic breaches[2] because they provided significant indicators of the program's fitness. As such, they provide good insight into the extent to which schedule and cost problems occur. Table 1 provides a definition of the variables used in the analyses reported below. The data were collected in the autumn of 2006 and all information was derived from quarterly Selected Acquisition Reports (SAR) and Defense Acquisition Executive Summary reports. The following section provides the findings of the investigation.

The first research question sought to identify whether joint systems differed from their single system counterparts. In short, it attempts to address the controversy that the two (joint versus single efforts) are similar in all but scale. To test this assertion, the 84 programs were

Figure 1: *Single System vs. Joint System*

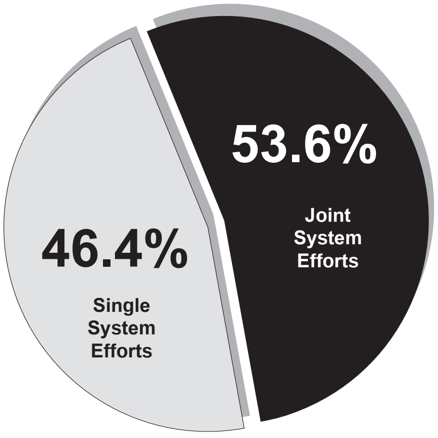


Figure 2: *Average Number of Breaches (1997-2005)*

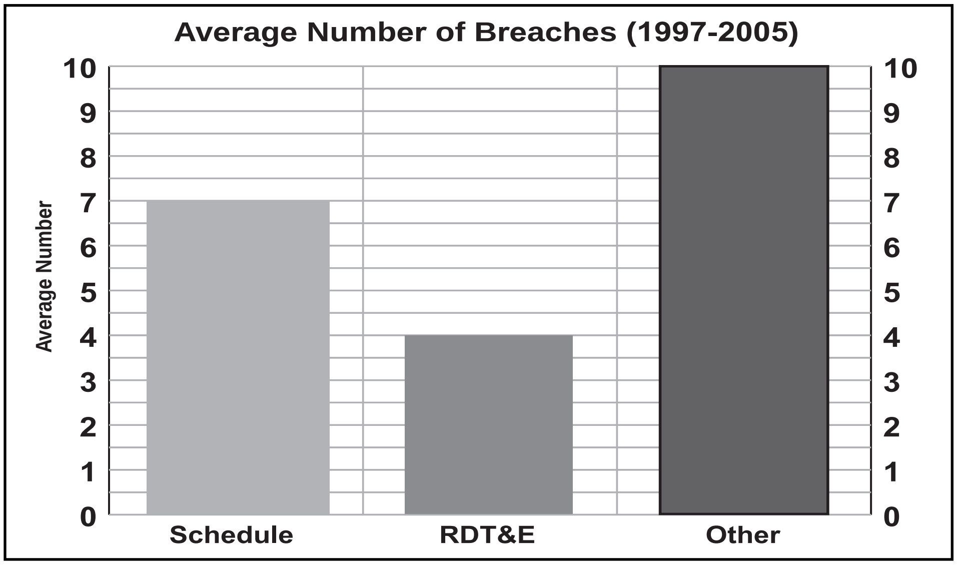
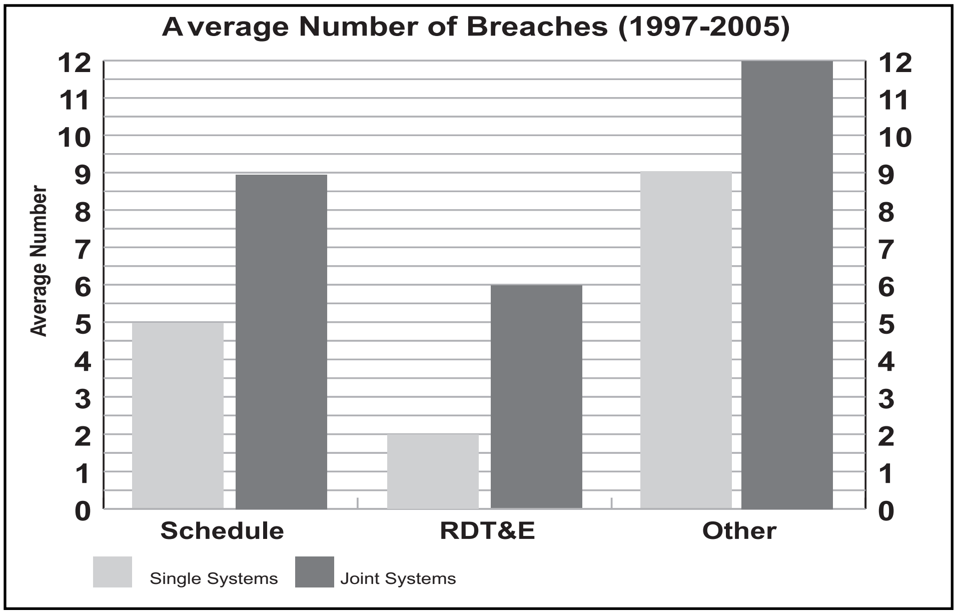


Figure 3: *Average Number of Breaches by Status (1997-2005)*



Variable	Mean Single Status	Mean Joint Status	Between Group Sum of Squares	Df	F Ratio	Sig
Schedule Breaches	4.58	8.6	341.75	1	5.19	.025
RDT&E Breaches	1.65	5.95	388.23	1	12.14	.001
Other Breaches	7.85	11.59	293.22	1	1.68	.198

A One-Way Analysis of Variance is a statistical procedure that tests the equality of three or more means at one time by using variances. The analysis of variance procedure compares the ratio of between group variance to within group variance. If the variance caused by the interaction between the groups is much larger when compared to the variance that appears within each group, then it is believed that the means are significantly different. Each sum of squares has corresponding degrees of freedom (Df) associated with it. The total Df is one less than the number of observations, N-1. The **F ratio** is the test statistic used to decide whether the sample means are within sampling variability of each other. That is, it tests the hypothesis H_0: $\mu_1 ... \mu_g$, thus it is equivalent to regression procedures in that it tests whether the model as a whole has statistically significant predictive capability.

Table 2: *ANOVA With Program Status a Factor Variable*

divided into two groups based on whether their SAR mission definition indicated any partnership relationships with any other MDAP programs. Thus, *Single System Efforts* did not indicate any partnership of jointness, whereas *Joint Systems Efforts* explicitly indicated a partnership/joint status. (A full list of the 84 programs can be found at <www.stsc.hill.af.mil/crosstalk/2007/06.stc.html>.) Of the 84 active programs, 46.4 percent (39 programs) were classified as single system efforts and 53.6 percent (45 programs) were identified as joint systems (see Figure 1, page 21).

In terms of the overall characteristics of the sample, the average age was six years. Approximately 54 percent were in the development stage and approximately 46 percent had already entered production and the average total program cost was $18 billion (in 2005 dollars). As noted in Figure 2 (see page 21), schedule breaches tended to be the most problematic for the programs under study (recall that *other* is the summation of a number of breaches, none of which in isolation rose to the level of RDT&E of schedule breaches). The average program experienced roughly seven schedule breaches and four RDT&E cost breaches over the 1997-2005 time frame. The average program also experienced roughly 10 *other* breaches (such as procurement, program acquisition unit cost and average procurement unit cost – see Table 1).

Furthering this examination, Figure 3 (see page 21) identifies the average number of breaches by the joint/single status. Single system efforts averaged roughly five schedule breaches, whereas the joint programs encountered approximately nine schedule breaches. The differences were statistically significant at conventional levels. Single systems also differed from their joint counterparts on RDT&E cost breaches. Single system programs had, on average, two RDT&E breaches, whereas joint efforts averaged six RDT&E breaches. With respect to the *other* breaches, single systems averaged nine breaches and joint systems averaged 12.

To test whether the differences were statistically significant, analysis of variance (ANOVA)[5] tests were conducted. As demonstrated in Table 2, the two groups (single versus joint) were statistically significant at conventional levels for both the schedule and the RDT&E breaches (p <.025 and <.001 respectively). While the two groups differed in terms of the number of *other* breaches, the differences failed to achieve statistical significance. Given that the status (single versus joint) of the program proved to be a significant indicator of schedule and RDT&E breach, a series of subsequent tests were conducted to test whether the program's status would continue to prove to be a good indicator of breach when controlling for the effects of age, size, and stage of development[6].

Data were obtained from the 2005 SAR on each of the program's age, size, and stage of development and three multivariate regression[7] tests were conducted. The first regression model tested for the influence of status on schedule breaches controlling for size, age, and stage. As demonstrated in Table 3, controlling for size, age, and stage, the program's status (joint/single) continued to be an indicator, albeit weak as noted by the p <.059, of schedule breach. The effects of status (holding constant size, age, and stage) on RDT&E cost breaches also proved robust. For these 84 cases, size, age, and stage did not prove to be significant indi-

Table 3: *Regression Results for the Effects of Size, Age, Stage, and Status on Programmatic Breach (1997-2005)*

Variable	Unstandardized Coefficients		Beta	t	sig
	b	Std. Error			
Dependent Variable: Schedule Breaches					
(Constant)	3.372	1.901		1.774	.080
Size	-8.17E-006	.000	-.037	-.323	.748
Maturity	.209	.218	.117	.959	.341
Stage	.390	2.027	.024	.192	.848
Status	3.628	1.896	.220	1.914	.059
Dependent Variable: RDTE Breaches					
(Constant)	.848	1.311		.647	.520
Size	-2.24E-005	.000	-.140	-1.282	.204
Maturity	.027	.151	.020	.178	.859
Stage	1.667	1.398	.138	1.192	.237
Status	4.539	1.307	.377	3.472	.001
Dependent Variable: Other Breaches					
(Constant)	3.757	3.051		1.231	.222
Size	-2.87E-005	.000	-.081	-.706	.482
Maturity	.448	.350	.154	1.279	.205
Stage	3.110	3.254	.116	.956	.342
Status	4.637	3.043	.173	1.524	.132

A linear regression line has an equation of the form $Y = a + bX$, where $\boldsymbol{X}$ is the explanatory variable and $\boldsymbol{Y}$ is the dependent variable. The slope of the line is $\boldsymbol{b}$, and the constant ($\boldsymbol{a}$) is the intercept (the value of $\boldsymbol{y}$ when $\boldsymbol{x}$ = 0). In the regression procedure, the unstandardized coefficient $\boldsymbol{b}$ is the slope of the independent variable. The standard errors of the regression coefficients are used for hypothesis testing and constructing confidence intervals. The Standardized coefficients (Beta) are what the regression coefficients would be if the model were fitted to standardized data. The **t** test is a test of significance. It tests the hypothesis that a population regression coefficient β is 0, that is, H_0: $\beta = 0$. It is the ratio of the sample regression coefficient $\boldsymbol{b}$ to its standard error. The significance is a function of the **t** value and provides an indication of the probability that the coefficient is due to random chance.

cators of the occurrence of either schedule or RDT&E breach. And none of the variables (size, age, stage, and status) provided predictive insight into the occurrence of other breaches.

Due to the finding that program status continued to provide insight into schedule breaches regardless of the size, age, or stage of development, the programs were examined in light of the average number of months of schedule delays that were experienced. For the programs that encountered a schedule slippage, the average slippage was 57 months. The average cost of the slippage was calculated by first dividing the total expenditures to date by the number of months the program received funding. The cost of slippage was then calculated for each program by multiplying the programs' average monthly expenditure by the number of months slipped. For those programs that encountered slippage, the average cost of slippage was $1,818 million.

Implications for Acquisition

The results of the research provide convincing evidence to suggest that, when considering program breach as an indicator of program fitness, joint programs do in fact differ from their single system counterparts. Joint systems encountered substantially more breaches than did single system efforts. And, the effects of status as a reliable indicator of breach, held true regardless of the program's size, age, or stage of development. These results suggest that joint programs, whether large or small, in development or production, and irrespective of age, are statistically more likely to encounter programmatic breaches than their single system counterparts.

The findings pose several salient issues for acquisition. First, the finding that joint programs differ from single systems and are more susceptible to schedule and cost breach is noteworthy. Jointness matters. Program managers involved in the acquisition of joint initiatives should heed these findings as a call to pay particular attention to attendant program risk factors as a means of mitigating potential problems. Unfortunately, the research cannot offer guidance on what forms of risk to monitor; it did not capture whether the typical single system risk factors continue to apply in a joint setting.

Furthermore, the research did not flesh out why joint efforts encounter more breaches. In other words, are there some unique characteristics of joint efforts that put them at higher risk? Or, might it be that the original estimation techniques were not well suited for joint efforts? In other words, is it the process of joint acquisition or is it the failure to accurately predict cost and schedule at the onset? This subtlety is an important distinction that warrants further investigation.

Second, little is known about the downstream cascading effects of interdependencies. Isolating the risk factors for the spill-over ramifications that one program may unintentionally impose against another downstream may be vitally important in a world of joint capabilities. Spillover effects could potentially explain why large programs breach despite the intense management oversight applied to them; it may be due to the spill-over effects of upstream, interdependent programs.

If joint capabilities is, in fact, a paradigm worth pursuing, then these findings indicate that further research on programmatic interdependencies is not just warranted, but imperative.

Third, the findings suggest that size, age, and stage offer little insight into the potential to breach. As such, size, age, and stage are not only *not* good early indicators, but they do not seem to offer any real immunity to breach. This finding tends to question the merits of the traditional approaches to classifying major defense acquisition programs for in-depth scrutiny solely on the basis of size or cost alone.

The results appear fairly clear – more applied research is definitely needed. Testing to see whether breaches are more related to one form of interdependency than another may prove helpful to acquisition managers that must navigate the choppy waters of joint efforts. It is quite plausible that different forms of interdependencies exist and that they will not all manifest the same influences. Additional research is also needed on how different forms of interdependency may interrelate to place a program at either higher or lower risk. Moreover, given the findings, further research may substantiate that interdependencies exhibit unique cost characteristics (see the discussion on page 21 on transaction costs) that may require distinct methods and metrics for estimating (and monitoring) program cost, schedule, and risk.

Finally, the findings indicate that the acquisition process is not impervious to the transformational activities underway in the DoD. As program structures change in important ways (i.e. from single to joint), it comes as little surprise that the management metrics, measures, and methods employed to undergird program acquisition would require modification. Additional insight into the specific nature of interdependencies and the management levers that act to tame the problems that interdependencies spawn, is clearly warranted.

While these findings provide important insights into the acquisition of joint programs, the findings should be interpreted with caution. The limitations of the study (for example, the one-point in time snapshot view, the limited manner of classifying interdependence or *jointness*, and the failure to include other important factors that may prove significant) cannot be overlooked. Nonetheless, the results provide reason to pursue the study of *interdependence* as a potential indicator of programmatic outcomes.

If *joint capabilities* is, in fact, a paradigm worth pursuing, then these findings indicate that further research on programmatic interdependencies is not just warranted, but imperative.◆

References

1. Pracchia, Lisa. "Improving the DoD Software Acquisition Processes." CrossTalk Apr. 2004 <www.stsc.hill.af.mil/crosstalk/2004/04>.
2. U.S. Government Accountability Office (GAO). "Best Practices: Better Support of Weapon System Program Managers Needed to Improve Outcomes." GAO-06-110. GAO, 2005.
3. Krieg, Kenneth. "Defense Reforms." Global Security, 2005 <www.globalsecurity.org/military/library/congress/2005_hr/050421-krieg.pdf>.
4. U.S. GAO. "DoD Acquisition Outcomes: A Case for Change." GAO-06-257T. GAO, 2006.
5. England, G.R. "Remarks by the Deputy Secretary of Defense." Military Communications 2006 Conference, Washington, D.C., 25 Oct. 2006 <www.defenselink.mil/Speeches/Speech.aspx?SpeechID=1059>.
6. Thompson, J.D. Organizations in Action: Social Science Bases of Administrative Theory. New York: McGraw-Hill, 1967.

7. Coase, R.H. "The Nature of the Firm." Economica 4.16 (Nov. 1937) <www.cerna.ensmp.fr/Enseignement/CoursEcoIndus/SupportsdeCours/COASE.pdf>.

Notes

1. Programs that are estimated to require an eventual total expenditure for research, development, testing and evaluation (RDT&E) of more than $365 million in fiscal year (FY) 2000 constant dollars or, for procurement, of more than $2.190 billion in FY 2000 constant dollars are classified as ACAT 1 programs.
2. The Acquisition Program baseline (APB) monitors program development metrics. Performance outside the predicted thresholds results in programmatic breaches. These breaches are viewed as unfavorable outcomes for program development.
3. Milestone B marks the beginning of the System Development and Demonstration stage of program acquisition. It is the first stage that requires a formal acquisition strategy that will be employed to track and monitor program progress.
4. The APB requires every program manager to document program goals prior to program initiation. Program managers identify set goals and a series of objective values that serve to provide thresholds for monitoring progress.
5. ANOVA is a statistical method that tests whether the averages (means) of two groups are statistically different. It does this by calculating a mean for the entire sample and then comparing it against the mean of each group. As such, it tests whether the individual group's mean differs from the entire population.
6. In modeling relationships between two variables, statisticians are often asked to test whether the relationship may actually be due to the actions of a third variable. For example, perhaps it is not the joint nature that leads to breach, but rather it is the size of the program. By including size, age, and stage in the multiple regression model, we can isolate the effects of jointness irrespective of the program's age, stage, and size. In this way we are *controlling* for any effects that size, age, or stage may impose on breaches.
7. Regression techniques test for the extent to which one variable is a direct function of another variable. In short, it examines how much of the dependent variable can be explained or predicted by knowing the value of the independent variable. It is capable of testing both the strength and the direction of the relationship between two variables. It is also capable of testing the effects of multiple variables on one dependent variable – in this case, it is referred to as *Multiple Regression*. For further insight into these techniques, see: Lind, Douglas, William Marchal, and Robert Mason. "Statistical Techniques in Business and Economics." 11th ed. New York: Irwin/McGraw-Hill, 2002 or <www2.chass.ncsu.edu/garson/pa765/index.htm>.

About the Authors

Mary Maureen Brown, Ph.D., is an associate professor at the School of Government at the University of North Carolina at Chapel Hill. She has more than 15 years experience in the design, development, and implementation of advanced information technology in government organizations. Brown also serves as a Visiting Scientist at the Software Engineering Institute (SEI) at Carnegie Mellon University, and as a Senior Fellow with the Center for Excellence in Municipal Management at George Washington University where she provides instruction and consultation to senior federal and state officials on the adoption and implementation of advanced technologies. Brown received her doctorate from the University of Georgia.

SEI/University of North Carolina
at Chapel Hill
CB #3330 Knapp-Sanders BLDG
Chapel Hill, NC 27599-3330
Phone: (919) 966-4347
Fax: (919) 962-0654
E-mail: brown@sog.unc.edu

Rob M. Flowe is an operations research analyst in the Office of the Secretary of Defense, Program Analysis, and Evaluation where he serves as a cost analysis improvement analyst for a variety of major defense acquisition programs. Flowe retired from the U.S. Air Force in 2003, having gained experience in the acquisition of space, Command, Control, Communications, Computer, and Intelligence, and software-intensive systems. He achieved his Level 3 certification in program management from the Defense Systems Management College in 1998, has a Bachelor of Science in Aerospace Engineering from Virginia Tech, and a Master of Science in Software Systems Management from the Air Force Institute of Technology.

Office of the Secretary of Defense
Program Analysis and Evaluation
Deputy for Resource Analysis
Weapon Systems Cost Analysis Div.
1800 Defense Pentagon
Washington, D.C. 20301-1800
Phone: (703) 697-3845
Fax: (703) 692-8054
E-mail: robert.flowe@osd.mil

Sean Patrick Hamel earned his master's degree from the University of Tennessee in 2004. Since graduation, he has been supporting research that is working to refine defense system costing methodologies. Hamel's primary interest with the research is to understand and identify systemic interdependencies and the related outcomes and implication of these relationships. Currently, he is planning to extend this experience and knowledge towards earning his doctorate in Public Administration starting in the fall of 2007.

SEI Research Affiliate
2316 Chapel Hill RD
Durham, NC 27707
Phone: (805) 588-6560
E-mail: sphamel@andrew.cmu.edu

Defense Acquisition Performance Assessment – The Life-Cycle Perspective of Selected Recommendations

Dr. Peter Hantos
The Aerospace Corporation

As a significant milestone in the Department of Defense's (DoD) continuous self-assessment process, an important document, the Defense Acquisition Performance Assessment (DAPA) report, was released in early 2006. The report – in its sweeping and integrated assessment – attempted to consider all critical aspects of defense acquisition and made recommendations for each of the major elements of the Defense Acquisition System (DAS). The author's goal in this article is to analyze the conceptual integrity of selected recommendations, using an approach that has been refined during the author's life cycle modeling research. Here, conceptual integrity refers to potential contradictions between the recommended actions that, when viewed independently from each other, appear to be viable. Why the life-cycle modeling focus? Life-cycle models represent the backbone of both acquisition and development processes, and this focus facilitates the analysis of concerns that crosscut in the impacted domains.

On June 7, 2005, Gordon England, Acting Deputy Secretary of Defense, authorized an assessment of the DAS, and created a panel to carry out the DAPA project. A detailed review that covers all aspects of the final report is beyond the scope of this article. Interested readers are invited to study the full text, which can be downloaded from the panel's Web site [1].

Of the panel's recommendations, the following four were selected for discussion on the basis of their life-cycle modeling aspects:

- Allowing program managers to defer non-Key Performance Parameter (KPP) requirements.
- Realigning Milestone B to occur at Preliminary Design Review (PDR) in the Defense Acquisition Management Framework.
- Improving the measurement of technology readiness.
- Making time (schedule) a KPP for the acquisition.

While each of these recommendations appears sound in the abstract sense, their implementation would pose serious challenges. The objective of this article is to identify inherent, life-cycle, structure-related problems with the Defense Acquisition Management Framework that would have to be resolved before attempting to implement the reviewed recommendations.

Because the article is concerned with cross-cutting issues, it did not seem effective to use the traditional approach of reviewing each recommendation in the order in which it is discussed in the DAPA report. Instead, a kind of *reverse* approach has been chosen. A comprehensive, albeit hypothetical, case study of a military space system is presented, and the potential impact of relevant DAPA recommendations on this sample acquisition is explored. The expectation is that the case study will demonstrate implementation ambiguities intrinsic to the panel's recommendations.

The Current Acquisition System

Figure 1 sets the context of the discussion. The diagram shows the interfaces and interactions among the three processes of the DAS: Planning, Programming, Budgeting, and Execution (PPB&E), Joint Capabilities Integration and Development System (JCIDS), and the *little a* acquisition process outlined in the DoD 5000.2 instruction. The shading in Figure 1 means to further emphasize that the article's analysis is only focusing on panel recommendations that are related to the *little a* dimension of the DAS. Since the case study is a military space acquisition example, a mapping of the DoD 5000.2 Defense Acquisition Management Framework [2] into the National Security Space Acquisition Policy 03-01 (NSSAP) acquisition phases [3] is needed. This mapping is shown in Figure 2 (see page 26). Note that the major phase gates are called *milestones* in DoD 5000.2 but are referred to as Key Decision Points (KDPs) in NSSAP 03-01. The content of the technical reviews is the same, as their names are similar, and all represent system-level reviews. In DoD 5000.2, these reviews are as follows: System Requirements Review (SRR), System Functional Review (SFR), PDR, and Critical Design Review (CDR). In NSSAP 03-01, System Design Review (SDR) replaces SFR. In both processes, IOC represents Initial Operational Capability.

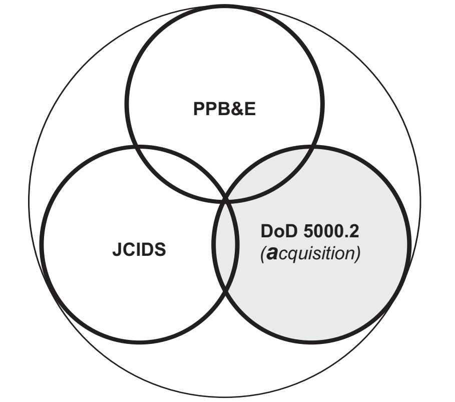
NSSAP 03-01, unlike DoD 5000.2, distinguishes between two acquisition models. One, the Small Quantity Model, is slated for the acquisition of the majority of space assets. The second, the Large Quantity Production Focused Model, is used for the acquisition of user equipment, terminals, etc. In Figure 2, the mapping for the Small Quantity Model is presented. What makes space systems different from the majority of weapon systems? First, they are highly software-intensive. Typical ground control systems have millions of lines of code, and even the spacecraft and satellite payload segments could easily contain a half-million lines of code. Second, satellite systems, along with their ground stations and boosters, are usually acquired in quantities of 10 or less due to the high expense of satellites and launch costs. These systems are practically custom-built rather than *mass-manufactured*, hence the need for the Small Quantity Model.

Space System Acquisition Case Study

This system would ultimately replace an existing network of military satellites that is slowly becoming obsolete. New, critical capabilities are planned. The final system in space would manage mixed missions, generations, and constellations of satellites. On the ground, a complex network of space/ground connections, mobile and permanent ground stations, and command and analysis centers are envisioned.

Evolutionary Acquisition (EA) has been chosen as the acquisition strategy. EA is

Figure 1: *Interaction Among PPB&E, JCIDS, and DoD 5000.2*



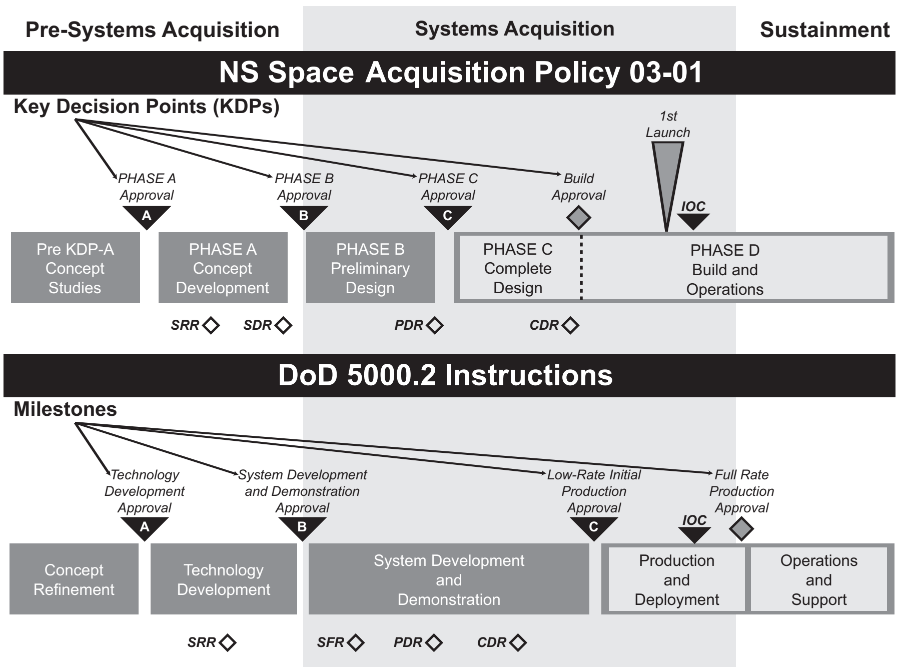

Pre-Systems Acquisition
Systems Acquisition
Sustainment
NS Space Acquisition Policy 03-01
Key Decision Points (KDPs)
1st Launch
PHASE A Approval
PHASE B Approval
PHASE C Approval
Build Approval
A
B
C
IOC
Pre KDP-A Concept Studies
PHASE A Concept Development
PHASE B Preliminary Design
PHASE C Complete Design
PHASE D Build and Operations
SRR
SDR
PDR
CDR
DoD 5000.2 Instructions
Milestones
Technology Development Approval
System Development and Demonstration Approval
Low-Rate Initial Production Approval
Full Rate Production Approval
A
B
C
IOC
Concept Refinement
Technology Development
System Development and Demonstration
Production and Deployment
Operations and Support
SRR
SFR
PDR
CDR

Figure 2: *Mapping of DoD 5000.2 Into NSSAP 03-01*

defined as an acquisition approach that delivers capability in an incremental fashion, recognizing the up-front need for future capability improvements. These future capabilities are to be contracted and delivered in the context of successive acquisition increments (Figure 3). As part of the acquisition strategy it is also decided that the contract in the first increment of the acquisition (to be referred as First Acquisition Increment) would have two major deliveries, in effect calling for the development and delivery of two system increments[1]. The planned content of these System Increments is as follows: The entire ground system (except for future mobile stations) would be developed in the first system increment. The operational acceptance test of this new ground system would involve the full control of selected, existing constellations. All new space assets (spacecraft and payload hardware/software) and the mobile stations would be delivered in the second system increment.

The plan is to first launch only a few prototypes of the new satellites, then decide about the acquisition of more satellites later. New requirements are expected for the ground system on the basis of experience gained during the launch and operation of the prototype satellites. Most likely, other mission and satellite payload capability requirements will also emerge, triggering the need for a generation of new satellites.

The program's acquisition strategy outlines a plan for soliciting bids from up to three contractors during the Pre-KDP-A Concept Study Phase, down-selecting to two at KDP-A, and making a final decision at KDP-B. This is an expensive but highly risk-aversive strategy to mitigate contractor uncertainties. Figure 4 illustrates a simplified life-cycle model, accommodating the first acquisition increment.

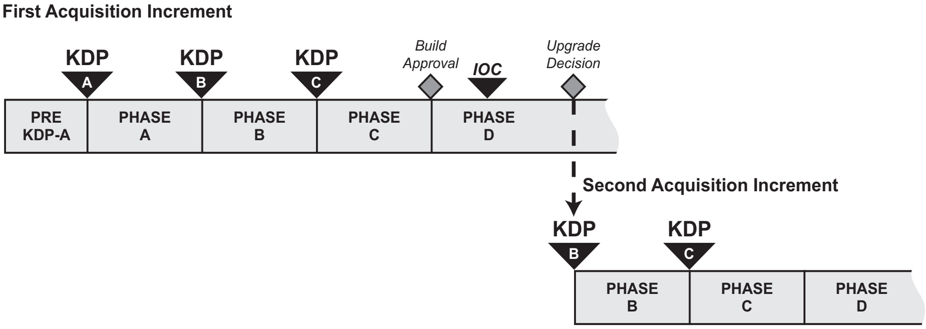
Figure 4 depicts several concurrent streams of events and their relationships, showing a notional alignment of the Milestone Decision Authority (MDA) actions with the decision-obligation-spending sequences of the PPB&E process. Congress allocates money for only one year's worth of activity. So PPB&E is repeated every year, and the appropriated funds, even though they belong to the same program, are in different spending states depending on when they were approved.

An explanation of the depicted contract actions is as follows: The program would require a Lead System Integrator (LSI) – sometimes called the *Prime* if the main contractor performs development tasks as well – and the contributions of, most likely, several sub-contractors. During the Pre-KDP-A period, three contractors are to provide concept studies. Following an evaluation of these studies, the MDA invites only Lead-2 and Lead-3 to continue. In Phase A only the potential leads compete, but upon entry to Phase B the selected lead chooses sub-contractor partners, hence the change to *team* designation.

With respect to funding, a naïve assumption is that work would only start after the budget and contracts are secured. In reality, companies that want to stay in the game have to be involved in continuous research and technology development even before the solicitations go out, and the funding of such activities must come from internal resources. These technology development and miscellaneous research activities are not shown in detail. For example, to bid for this project, Lead-1 (who ultimately was not invited to continue in Phase A) would already be engaged in relevant development activities. The same is true for potential sub-contractors. In Figure 4, the blocks with upward diagonal shading represent this early engagement. Some of the efforts during bidding are covered by the government, but it is not unusual for companies to pay for their expenses in an expectation of winning a lucrative long-term contract.

Study of the technical reviews in the overall life-cycle structure results in further controversies. These reviews – holdovers from the long-defunct Military Standard (MIL-STD)-1521B – are based on the Waterfall process, because in 1985, at the time of the last update of the standard, Waterfall was the only approved development life-cycle model for the DoD. (For further details, see [4].) For example, SDR is supposed to be a technical review of the system design supporting the MDA's decision-making at KDP-B, the entry to the preliminary design phase. The fact that system review is supposed to precede the start of preliminary system design is confusing, and neither the phase nor the review name/content is consistent with reality. Planning and conducting system PDR in Phase B is problematic as well. In Phase B, design and development of all segments progresses at different paces; total, vertical synchronization of reviews (i.e., lining up segment-level design

Figure 3: *Successive Acquisition Increments in Evolutionary Acquisition*



reviews for ground software, spacecraft software, spacecraft hardware, payload software, etc.) is simply not feasible. The first ground system increment must be almost ready for integration, and there must be substantial progress on the spacecraft and payload side as well. By the time system CDR comes, the disconnect is even more striking. The life-cycle modeling-based analysis shows the root cause for this disconnect. The first increment of the acquisition is a sequential structure by design, which via its naming conventions and phase descriptions enforces a Waterfall development life cycle. Such a life-cycle model is clearly inappropriate for a large scale, concurrent engineering project.

Figure 4 shows Spiral as the life-cycle model of choice for ground software development. Both DoD 5000.2 and NSSAP 03-01 state that Spiral Development (SD) is one of the main processes that perform EA. Are the depicted ground spirals what the government policies refer to? The answer is an unqualified *no*. From the earliest days, the prevailing misconception is that DoD 5000.2 *is* spiral development, where concept refinement is the first spiral, technology development is the second one, system development and demonstration is the third one, and so on. Also, entry criteria for every milestone (or for the corresponding KDPs in NSSAP 03-01) include required risk management activities (risk identification, risk reduction, and risk mitigation plans), reinforcing the notion that we are performing SD. At the same time, looking at the concise definition of the Spiral and its essential characteristics [5], it becomes clear that these activities are not what the successful application of spiral concepts assumes. The key risk-related mechanism that is unique to SD is embodied in (a) the concurrent engineering of all artifacts and (b) the risk-driven planning of the content, and consequently cost and schedule successive spirals. Having risk mitigation plans in the conventional sense is different from spiral planning. It involves the creation of additional plans to eliminate or gradually reduce the risk by having alternative course(s) of actions lined up in case the risk materializes or its likelihood drastically increases. A key element of such risk planning is that funding for alternative actions needs to be provided in addition to the allocated, regular cost of development.

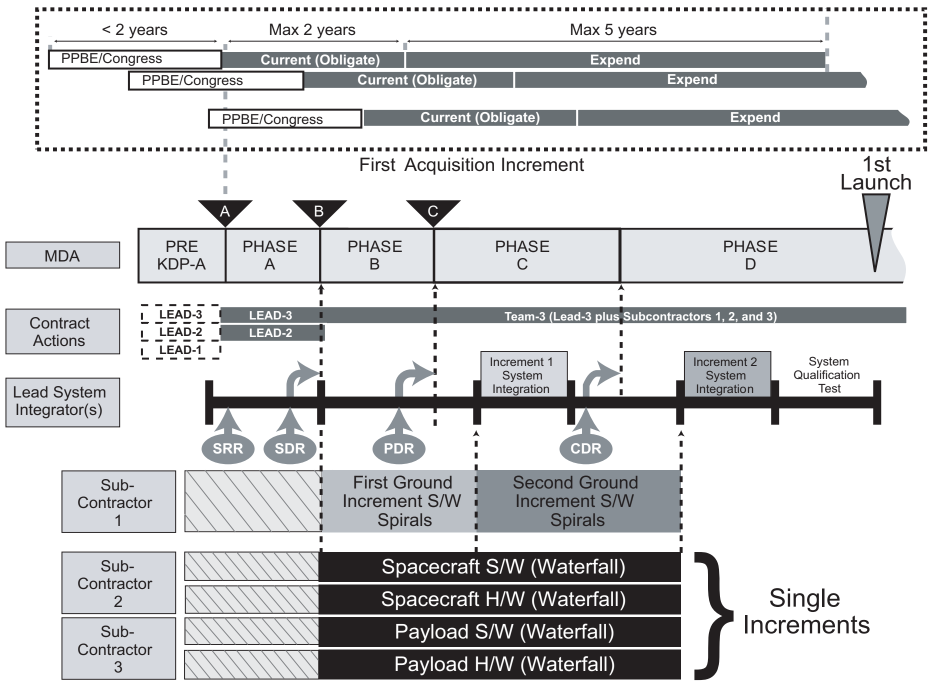


Figure 4: *Simplified Life-Cycle Model*

The applied SD method in this case study is a highly localized and not a system-level process, and it is not supportive of this program's EA strategy. While a detailed discussion of EA is beyond the scope of this article, some justification for this statement is needed. As NSSAP 03-01 states, during SD that supports EA, a desired capability is identified, but the end-state requirements are not known at program initiation. In our case study, not only system capabilities but detailed system requirements are also known prior KDP-B. In fact, even the high-level requirements for the two software increments are determined in advance and go on contract as well. Also, looking at Figure 3, it is becoming clear that development spirals (iterations) carried out during Phase B or even Phase C are far removed from the upgrade decision that triggers the second acquisition increment. The upgrade decision – besides new, emerging requirements – should be based on the status of current technology and user experiences gained during the operational phase and not on information gathered during earlier development spirals. The reader might also wonder, if this is the case, why SD was chosen by the case study's program manager for ground software development. Was it an arbitrary decision and was it a mistake? On the contrary, iterative development is the prudent strategy for this kind of large scale, concurrent engineering project, and SD is a well-known, *brand-name* iterative method. Quoting Martin Fowler's whimsical advice, *You should use iterative development only on projects that you want to succeed* ... [6].

As pointed out earlier, the acquisition life-cycle phases, the management commitment points, and their associated mandatory documentation represent a Waterfall sequence from the point of view of system development. This inability to reconcile the conflicting acquisition and development life-cycle models is one of the main reasons for the poor track record of the Spiral Model in defense acquisitions. In summary, applying spiral development in an acquisition increment to manage risks could be an effective project management strategy, but this strategy has nothing to do with the spiral process assumed in DoD 5000.2 or, for that matter, in the Defense Authorization Act of fiscal year 2003 that further specifies mandated characteristics of spiral development for major defense acquisition programs [7].

Deferral of Non-KPP Requirements

Allowing program managers to defer non-KPP requirements to later upgrades is an attractive proposition from the program manager's view. It provides an effective risk management tool by greatly expanding their decision-making authority and flexibility. In the context of our case study, how could the program manager using this newly acquired freedom reduce the scope of the first acquisition increment? Unfortunately, analysis shows there are not many opportunities after all. One possibility is to make the delivery of the first spiral of the ground system the first acquisition increment. This is a useful and complete capability (controlling the existing constellation of satellites), but it does not provide enough value to the customer, since there was already an operational system in place. In other words, the delivery of this new but compatible ground system is an excellent engineering objective, but insufficient as an acquisition objective. Also, it is not clear what we would do about spacecraft and payload development. They can not be deferred until after the delivery of the first increment of the ground system; that would push out the availability of new satellites with new capabilities to an unacceptable, distant period. On the other hand, if their development is started simultaneously with the ground system, at the time of ground system delivery they would be still in an incomplete, intermediate state of their Waterfall process-streams. Receiving documentation, prototype breadboards and models, and maybe some untested code would not be an acceptable acquisition value proposition either.

There are other considerations that would make the deferral of requirements difficult. For example, complex graphics and elaborate display designs are important in any ground system. As a requirements-pacing strategy, one might consider releasing the first version of the ground software with simplified user interfaces. This is an effective engineering approach, but it may backfire with end-users of the system. In similar situations, satellite operators forced to work with intermediate systems having limited capabilities created resentment and blocked buy-in when the final system became available.

In conclusion, the opportunity for delaying non-KPP requirements is great, but complex space systems might not always lend themselves to a feasible granularity of requirements for such deferral.

Realignment of Milestone B

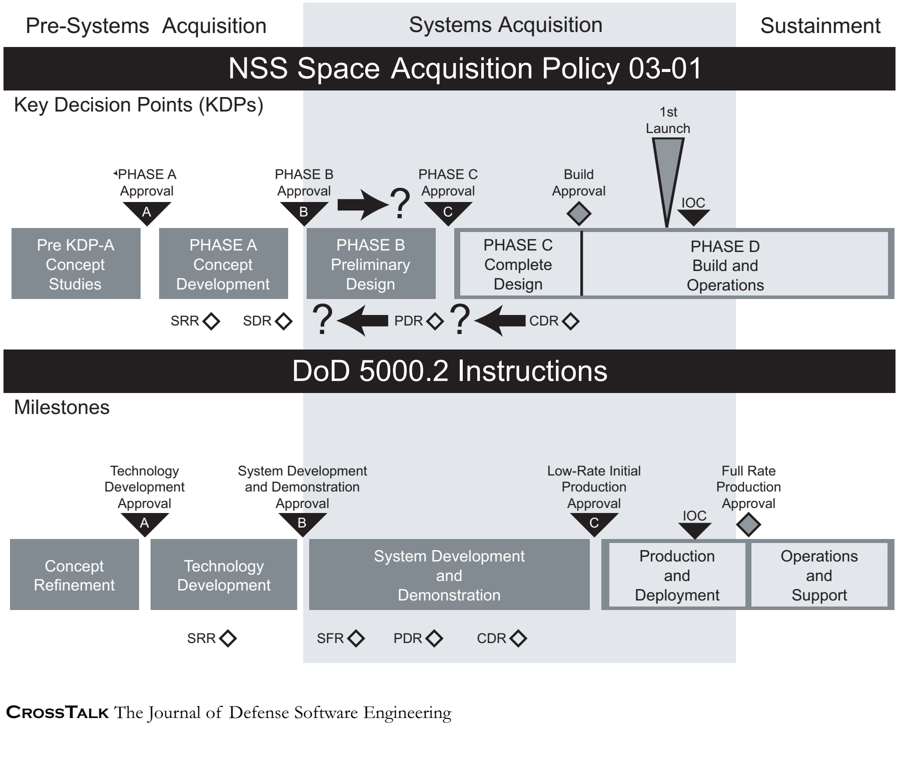
This DAPA recommendation calls for the realignment of Milestone B to occur at PDR, and the justification is as follows: The greatest trade space and the largest risk reduction opportunities exist between Milestone A and Milestone B, and the DoD places most program focus on Milestone B, because premature technology and system design decisions at Milestone B lead to technical problems during system design and development. Unfortunately, the term *realignment* is ambiguous due to lack of implementation details. Using the equivalent NSSAP 03-01 terminology, it needs to be clarified whether KDP-B should be moved forward or PDR moved backward (Figure 5).

Again, the phase definitions and reviews are in conflict. The declared objective of KDP-B is to gauge entry into Phase B. This phase-gate objective would indicate that we cannot talk about the move of KDP-B, only the move of PDR. However, if Phase C's objective is complete design, then PDR must immediately precede it. Moving up PDR means that its successful completion would lead us to complete design activities during a phase that is only designated for preliminary design. Finally, we are left with the delicate but unanswered question of positioning CDR. Would CDR move up as well? The unfortunate conclusion – again – is that the root cause of the problem is the ingrained Waterfall that is imposed on the developer by the acquisition models, and the planned move of decision points or reviews would not help either the MDA or the program manager.

Technology Readiness

As mentioned earlier, technology was identified as an important focus area for the DAPA inquiry. The findings state that there are no clearly definable measures of technology readiness, and the inability to define and measure technology readiness during Technology Readiness Assessments (TRAs) is the reason that immature technology is incorporated into plans prior to Milestone B. On the contrary, numerous sources are available to help with technology readiness assessments (see, for example [8], [9], and [10]). These referenced materials provide a workable version of Technology Readiness Levels (TRLs), applicable to the hardware elements of Ground, User, and Launch Segments of space systems. Even though there is some ambiguity regarding the use of these TRLs for assessing software in general and the hardware elements of the Space Segment in particular, still, measuring technology readiness should not be the main concern. While the exploration of all issues is beyond the scope of this article, the examination of the life-cycle dimension of TRA highlights the following, inherent problem of the Defense Acquisition Management Framework.

Figure 5: *Realignment Possibilities for KDP B*


Pre-Systems Acquisition
Systems Acquisition
Sustainment
NSS Space Acquisition Policy 03-01
Key Decision Points (KDPs)
1st Launch
PHASE A Approval
A
PHASE B Approval
B
?
PHASE C Approval
C
Build Approval
IOC
Pre KDP-A Concept Studies
PHASE A Concept Development
PHASE B Preliminary Design
PHASE C Complete Design
PHASE D Build and Operations
SRR
SDR
?
PDR
?
CDR
DoD 5000.2 Instructions
Milestones
Technology Development Approval
A
System Development and Demonstration Approval
B
Low-Rate Initial Production Approval
C
IOC
Full Rate Production Approval
Concept Refinement
Technology Development
System Development and Demonstration
Production and Deployment
Operations and Support
SRR
SFR
PDR
CDR

The applicable DoD policy for technology maturation at Milestone B is unambiguous (Chapter 5.3 of the DoD desk-book on TRA [12]): *All Critical Technology Elements (CTEs) should be identified and successfully demonstrated on a TRL 6 or higher before Milestone B.*

The concern relates to the execution of this policy. This simplified case study shows five concurrent engineering streams: ground software, spacecraft hardware, spacecraft software, payload hardware, and payload software (user systems and launch systems are also important segments of a total space system solution but were omitted for simplicity's sake). A TRA must be conducted for all segments in all domains. It is fair to assume that if KDP-B is the one and only phase gate to exit from concept development, then the enabling, critical technology elements of all concurrent processes must be at high TRL. Is this a reasonable assumption? What happens if some of the technologies are riskier than others and do not mature at the same pace? Clearly, this imbalance of concurrent engineering streams puts the predictability of the overall program in jeopardy. Or, theoretically, design of critical parts for the whole program could be forced to idle until the resolution of delinquent technology issues in the affected segments is completed, but that is obviously not a feasible option either.

Time Certain Development

One of the recommendations would

declare Time Certain Development as the preferred acquisition strategy by making time a KPP for the acquisition. First, when programs at Milestone A would be required to be budgeted on the basis of high-confidence estimates. Second, when the time-to-need and the current technology risk level are determined, the program should be time-constrained. Finally, technical performance should be traded-off to maintain this schedule. (see page 51 of the DAPA Report [1]). However, cost and schedule estimation in the presence of technology risks is difficult for various reasons. Theoretically, in all conventional parametric cost estimation models, cost, schedule, and performance can be seamlessly traded (although, this trade only works for routine, repeatable activities – in the case of software, for coding). The models establish an exponential relationship between performance and cost, and also between cost and schedule, to facilitate this trade. Fred Brooks pointed out an important and frequently overlooked fact in his classic book [11] that when a task cannot be partitioned because of sequential constraints, the application of more effort has no effect on the schedule. In terms of technology development, the process is inherently a sequence of learning steps, building on the results of previous experiments. This sequential process of experimentation and learning, combined with the probabilistic nature of success, make the implementation of Time Certain Develop-ment very problematic.

Conclusion

The acquisition life-cycle models of the DoD/NSSAP policies are inherently Waterfall, and as such, inadequate for the acquisition of large-scale, software-intensive systems, even if they are used with the intent of EA. The concerns raised in the case study indicate that consideration for an additional DAPA focus area, engineering, would be required to develop feasible changes to the DoD 5000.2 and NSSAP 03-01 policies. One can speculate that the absence of engineering considerations in the recommendations for industry is intentional; reflecting a hands-off approach by not constraining the contractor's engineering solutions. It is indeed desirable not to proscribe engineering processes in acquisition policy documents. Nevertheless, the case study convincingly demonstrates that current – not even state-of-the-art, but certainly state-of-the-practice – engineering methods, particularly integrated life-cycle models of concurrent engineering and iterative development, represent severe, hidden constraints, and they would have to be considered as key influencing factors during reworking the *little a* acquisition system.

Finally, the panel recommends the use of system dynamics to analyze the internal relationships of the acquisition system [12]. System dynamics is a modeling approach to studying complex systems via the identification and simulation of internal feedback loops of the system [13]. System dynamics is indeed the right tool for analyzing the tension resulting from unintended consequences of conflicting behaviors, but one could argue that before such a sophisticated and complex tool is unleashed, analyzing the life-cycle model structure of development should be satisfactory for identifying some fundamental, systemic conflicts.◆

Acknowledgements

This work would not have been possible without assistance from the following:

- **Reviewers:** Suellen Eslinger, Mary Jo Gura, and Dr. Leslie Holloway, The Aerospace Corporation.
- **Sponsor:** Michael Zambrana, U.S. Air Force Space and Missile Center.
- **Funding Source:** Mission-Oriented Investigation and Experimentation Research Program, Software Acquisition Task.

References

1. DoD. DAPA Project. Mar. 2006 <www.acq.osd.mil/dapaproject/>.
2. DoD. "DoD 5000.2 Instruction on the Operation of the DAS." May 2003 <www.akss.dau.mil/darc/darc.html>.
3. United States. Secretary of the Air Force. "National Security Space Acquisition Policy 03-01." Dec. 2004.
4. Hantos, P., "System and Software Reviews Since Acquisition Reform." Southern California Software Process Improvement Network, Apr. 2, 2004 <www.uces.csulb.edu/spin/media/pdf/SoCal-SPIN_PH_slides-04-02-04.pdf>.
5. Boehm, B., and W. J. Hansen. "The Spiral Model as a Tool for Evolutionary Acquisition." CROSSTALK May 2001 <www.stsc.hill.af.mil/crosstalk/2001/05>.
6. Fowler, M. UML Distilled. 2nd ed. Addison-Wesley, 2000.
7. United States House of Representatives. FY2003 Defense Authorization Act. Section 803 of 116 STAT. 2604 Public Law 107-314.
8. Mankins, J.M. "Technology Readiness Levels – A White Paper." National Aeronautics and Space Administration, 1995 <www.hq.nasa.gov/office/codeq/trl/trl.pdf>.
9. Graettinger, C.P., et al. Using the Technology Readiness Levels Scale to Support Technology Management in the DoD's ATD/STO Environments. CMU/SEI-2002-SR-027. Sept. 2002.
10. "Department of Defense Technology Readiness Assessment (TRA) Desk book." May, 2005 <www.akss.dau.mil/darc/darc.html>.
11. Brooks, F.P. The Mythical Man-Month – Essays on Software Engineering. Addison-Wesley, 1982.
12. Venture Services, LLC. "Monitor Government." Appendix D – Defense Acquisition Performance Assessment Project Report, A Baseline Literature Review. DAPA Project, Mar. 2006 <www.acq.osd.mil/dapaproject/>.
13. Sterman, J.D., Business Dynamics: System Thinking and Modeling for a Complex World. McGraw-Hill, 2000.

Note

1. Here, increment is used in two different contexts as is common in current software standards. In acquisition increment refers to contractual and user concerns, while in development increment refers to engineering and implementation concerns.

About the Author

Peter Hantos, Ph.D., is a senior engineering specialist at The Aerospace Corporation. He is the principal investigator of the Unified Life Cycle Modeling research, an effort to introduce comprehensive modeling and simulation approaches to software-intensive system development. Hantos has more than 30 years of experience as a professor, researcher, software engineer, and manager. Prior to joining Aerospace, as principal scientist at the Xerox Corporate Engineering Center, he developed corporate-wide engineering processes for software-intensive systems. Hantos holds masters of science and doctorate degrees in electrical engineering from the Budapest Institute of Technology, Hungary.

The Aerospace Corporation
P.O. Box 92957 - M1/112
El Segundo, CA 90009-2957
Phone: (310) 336-1802
Fax: (310) 563-1160
E-mail: peter.hantos@aero.org

Who Are Those Guys?

Welcome to Backfire, the interview within a journal cross-examining popular icons for software truth. This month we are talking with Butch Cassidy and the Sundance Kid who, despite the rumors, started Hole-in-the-Web LLC, a software company based in Bolivia.

Gary: Why start an identity theft software company?
Butch: I have vision, and the rest of the world wears bifocals.
Sundance: You just keep thinkin' Butch. That's what you're good at.

Gary: Do you know what you're doing?
Butch: Theoretically.

Gary: Theoretically?
Sundance: I'm the programmer.
Butch: Is that what you call programming?
Sundance: Is that what you call running a company? If I knew you were going to stroll...
Butch: You never could code, not from the very beginning!
Sundance: And you were all mouth!
Butch: We seem to be short on brotherly love around here.
Gary: Gentlemen, a little decorum please.

Gary: Do you enjoy the software business?
Butch: Boy, you know every time I see software code, it's like seeing it fresh for the first time. And every time that happens, I keep asking myself the same question: How could I be so darn stupid to keep coming back?

Gary: Percy, what went through your mind when your former employees asked you to join their software company?
Percy: Morons. I've got morons on my team.

Gary: Identity theft is lucrative; are you profitable?
Butch: Do you believe I'm broke already?
Gary: Really, why?
Butch: Well, I swear, Gary, I don't know. I've been working like a dog all my life and I can't get a penny ahead.

Gary: Sundance says it's because you're a soft touch, always taking expensive vacations, buying drinks for everyone and you're a rotten gambler.
Butch: Well, that might have something to do with it.

Gary: Does a secure internet affect your business model?
Butch: What happened to the old internet? It was beautiful.
Sundance: People kept phishing it.
Butch: Small price to pay for beauty.

Gary: Companies are doubling their efforts to secure their internet communications and servers. Are you concerned?
Butch: If they'd just pay me what they're spending to make me stop robbing them, I'd stop robbing them.

Gary: What's next for Butch and Sundance?
Butch: It doesn't matter. I don't know where we've been and I've just been there.
Sundance: Butch and me have been talking it all over. Wherever the hell Bangalore is, that's where we're off to.
Sheriff Bledsoe: They should have let themselves get killed a long time ago when they had the chance. See, they may be the biggest thing that ever hit the internet, but they're still two-bit outlaws. I never met a soul more affable than Butch or faster than the Kid but they're still nothing but two-bit outlaws on the dodge. It's over, their time is over and they're gonna crash hard, and all they can do is choose where.

Gary: Microsoft, Oracle and Yahoo among others have hired cyber-bounty hunters to shut you down. Are you concerned?
Butch: Maybe there's a way to make a profit in this. Bet on Microsoft.
Sundance: I would, but who'd bet on you?
Butch: Whatever they're sellin', I don't want it!

Gary: They appear to be skilled, well funded, and relentless.
Butch: They're wasting their time. They can't track us over the internet.
Sundance: Tell them that.
Butch: I couldn't do that. Could you do that? Why can they do it? Who are those guys?
Sundance: They're very good.
Butch: Don't they get tired? Don't they get hungry?
Sundance: They gotta be.
Butch: Why don't they slow up? Hell, they could even go faster, at least that'd be a change. They don't even break for Starbucks.

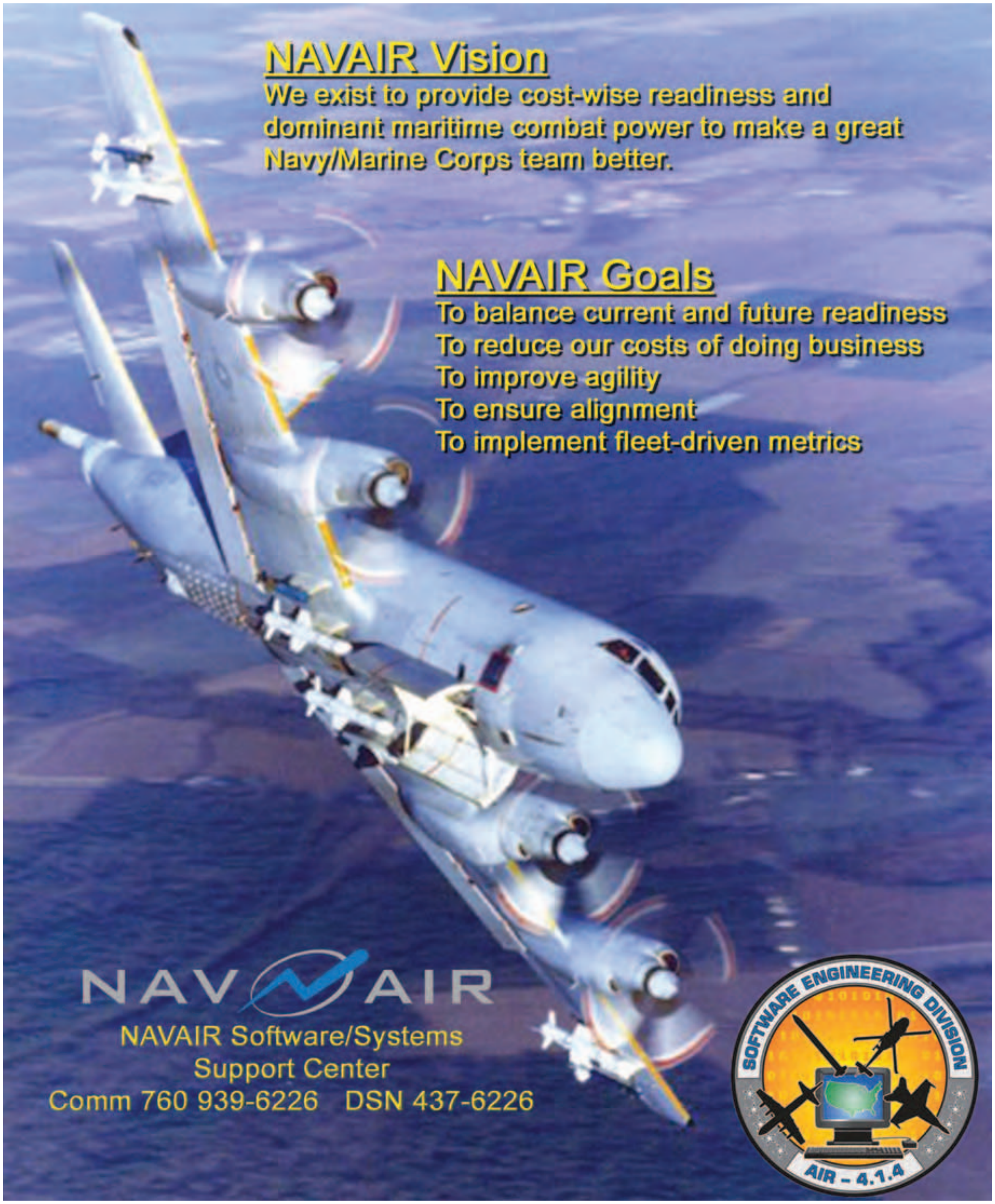
Sundance: Did you say they were hired permanent?
Gary: No, just until they destroy you.

Butch: Well, the way I figure it, we can either fight or give. If we give, we go to jail.
Sundance: I've been there already.
Butch: We could fight and they'll stay where they are with a honeypot. Or they could go for position with a honeymonkey and pick us off one at a time. Might even get a server overload started. What else can they do?
Sundance: They could surrender, but I wouldn't count on that.
Butch: Who are those guys?

Butch: Hey, wait a minute – you didn't say Google, did you?
Gary: Google? No.
Butch: Good; for a moment there, I thought we were in trouble.

Cyber-bounty Hunters: Fuego!

— Gary A. Petersen
Shim Enterprises, Inc.
gary.petersen@shiminc.com


NAVAIR Vision
We exist to provide cost-wise readiness and dominant maritime combat power to make a great Navy/Marine Corps team better.
NAVAIR Goals
To balance current and future readiness
To reduce our costs of doing business
To improve agility
To ensure alignment
To implement fleet-driven metrics
NAVAIR
NAVAIR Software/Systems Support Center
Comm 760 939-6226 DSN 437-6226
SOFTWARE ENGINEERING DIVISION
AIR – 4.1.4

CrossTalk / 517 SMXS/MXDEA
6022 Fir AVE
BLDG 1238
Hill AFB, UT 84056-5820

PRSRT STD
U.S. POSTAGE PAID
Albuquerque, NM
Permit 737

CrossTalk is co-sponsored by the following organizations:

NETWORKS AND INFORMATION INTEGRATION
ASSISTANT SECRETARY OF DEFENSE
DOD CIO

NAVAIR

76th SOFTWARE MAINTENANCE GROUP

309th SOFTWARE MAINTENANCE GROUP

402d SOFTWARE MAINTENANCE GROUP

DFAS

U.S. DEPARTMENT OF HOMELAND SECURITY
Homeland Security