Image designed by Harambee Dennis

TRAINING ARCHITECTURE PRACTICES IN ARMY ACQUISITION: AN APPROACH TO TRAINING SOFTWARE ARCHITECTURE PRACTICES IN U.S. ARMY ACQUISITION

Stephen Blanchette and John Bergey

Technology management skills in the Department of Defense (DoD) are not keeping pace with the advanced systems acquired by the DoD, especially as software becomes more prevalent in those systems. For a number of years, software architecture practices have been identified as enablers of program success, yet evidence suggests that too little attention is paid to the topic. The Army Strategic Software Improvement Program (ASSIP) seeks to dramatically improve the acquisition of software-intensive systems, in part through improved acquisition workforce skills. Through ASSIP, the Army has begun to build a level of technical expertise in modern software architecture practices within its acquisition community. This article discusses the training component of the ASSIP Software Architecture Initiative.

Since late 2002, the Software Engineering Institute (SEI), a federally funded research and development center operated by Carnegie Mellon University, has been working with (then) Assistant Secretary of the Army for Acquisition, Logistics & Technology (ASA[ALT]) Claude Bolton,[1] in a strategic partnership aimed at improving the Army's ability to acquire systems that are highly dependent on software (often called software-intensive systems [SIS]).[2] Through this partnership, known as the Army Strategic Software Improvement Program (ASSIP), the Army and

the SEI are engaged in several initiatives designed to enhance the Army's ability to be a "smart buyer" of software-intensive systems.

The need for ASSIP is readily apparent. Much has been made about the lack of technical depth demonstrated by the Department of Defense (DoD) in managing ever more complex system development programs (Government Accountability Office, 2007; Government Accountability Office, 2008). In fact, in 2007 the Pentagon initiated a study of the skills of its acquisition workforce (Erwin, 2007). One contributing factor to the shortage of technical skills is the growing presence of software in virtually every major system, from tanks to bombs to bullets, procured by DoD.

Early ASSIP investigations into Army SIS acquisition indicated, among other things, that while software architecture practices were deemed important for SIS programs, methods and skills to carry out those practices were perceived to be inadequate. Hence, the ASSIP formulated an initiative to raise the organic capabilities of Army acquisition in this important software development area. This article describes the work done to begin developing a foundation for an organic Army software architecture capability.

WHY SOFTWARE ARCHITECTURE?

One might question the focus on software architecture capabilities within the Army's acquisition workforce, but the reason becomes obvious when viewing architecture in terms of program success. First, a software architecture is the set of system structures that consist of "software elements, the externally visible properties of those elements, and the relationships among them" (Bass, Clements, & Kazman, 2003, p. 21). The software architecture underpins a system's software design and code; it represents the earliest design decisions—ones that are often difficult to change later (Bass et al., 2003), so getting the architecture "right" has enormous implications for the software and for systems reliant upon that software. It then follows that solid software architecture practices are essential to successful software-intensive programs.

In 2000, the DSB pointed to software architecture as "a central theme for software reuse, product lines, and greater exploitation of commercial technology and practices" (Defense Science Board Task Force, 2000, p. 3).

In fact, the importance of software architecture practices has been known for quite some time. In 1994, the Defense Science Board (DSB) highlighted the potential for software architecture and product line techniques to reduce cost and cycle times

(Defense Science Board Task Force, 1994). In 2000, the DSB pointed to software architecture as "a central theme for software reuse, product lines, and greater exploitation of commercial technology and practices" (Defense Science Board Task Force, 2000, p. 3). Further, a 2001 Army lessons-learned workshop focusing on software upgrade programs concluded, in part, that architecture is "a key technical focus for the system" (Anderson et al., 2001, p. 14), making special note of the criticality of the architecture in determining the future ability to upgrade the system (Anderson, et al., 2001).

Given that software architecture practices have been linked to successful SIS acquisition as noted above, one might have expected that such practices would be prevalent in Army (and other Services') acquisition programs. However, such is not the case. In 2002, the DoD Tri-Service Assessment Initiative (TAI) highlighted poor software architecture practices as one *systemic causal factor* of software-intensive systems issues, based on TAI assessments of 21 DoD programs (Charette, McGarry, & Baldwin, 2003; McGarry, 2002). Simply performing a task called "software architecture" is insufficient to leverage the benefits of software architecture. In fact, the act of producing an architecture is inadequate; both acquirer and supplier should also conduct an evaluation of the architecture's quality and robustness to ensure suitability for current and future needs as the system evolves. Indeed, a few of the larger defense contractors routinely employ some form of architecture evaluation (Bass, Nord, & Wood, 2006).

A recent SEI analysis of 18 software architecture evaluations performed by the Institute between 2000 and 2005 showed that over half of the evaluations revealed significant program risks driven by an organization's failure to appreciate the extent of the software architecture effort, as evidenced by lack of training, lack of tools, and poor planning. Further, about two-thirds of the risks discovered were risks of *omission* (architectural decisions either not made or not captured, for example) (Bass, Nord, & Wood, 2006). These observations are consistent with earlier reports that indicated organizations pay insufficient attention to software architecture practices, and suggest that architecture evaluators must be experienced enough to probe the architecture in detail rather than accept it at face value.

A review of findings from initial ASSIP data gathering efforts proved consistent with the studies noted above. For instance, acquisition professionals held the general impression that prime contractors' software architecture abilities were about average (Kasunic, 2004), suggesting a need for architecture evaluations to reduce associated program risk. Yet, according to interviews with some key programs and with the Army's Program Executive Officers (PEOs), government program office staffs were not sufficiently skilled to evaluate software architectures (Keeler, 2005; Blanchette, 2005).

Thus, software architecture is an acknowledged good practice in SIS programs, but one that is rarely executed effectively or evaluated rigorously.

SOFTWARE ARCHITECTURE INITIATIVE

Given that software architecture was one of the technical challenge areas facing Army program management offices (PMOs), the logical next step was to consider

what could be done to help PMOs use software architecture to their advantage. The SEI had been working with a few Army PMOs individually on software architecture issues (Bergey et al., 2005; Clements & Bergey, 2005; Clements, Bergey, & Mason, 2005), and while these efforts were successful, they were point solutions to a more systemic problem.

Understanding the significance of the studies discussed above for Army acquisition as a whole, Bolton charged the SEI to develop an ASSIP initiative to address the problems noted in software architecture. The resulting Software Architecture Initiative was approved by the ASSIP Action Group for implementation in fiscal year 2004. A training component formed the core of the initiative.

THE TRAINING PROGRAM

The SEI already had available a training curriculum for software architecture and, since it was designed to be taught either at SEI facilities or onsite at customer locations, it easily served as the basis of the ASSIP Software Architecture Initiative. The curriculum consists of six courses:

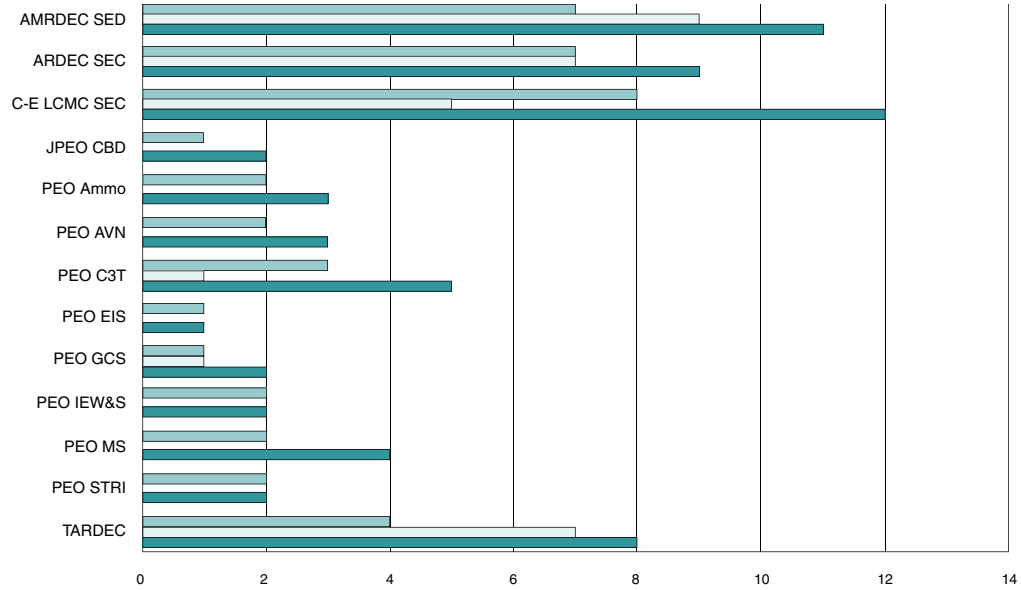
- Software Architecture: Principles and Practices
- Documenting Software Architectures
- Software Architecture Design and Analysis
- Software Product Lines
- SEI Architecture Tradeoff Analysis Method® (ATAM®) Evaluator Training
- ATAM Leader Training

Through a series of special offerings, the SEI delivered the curriculum at the Army software engineering centers (SECs) using the same materials and instructors as the publicly offered classes. The SECs provided the most central location for many participants since most of the Army's program offices are located in close proximity to one of the SECs. Students who completed the prescribed course sequences earned certificates just as if they had attended the regular public offerings.[3]

Generally, each course had 30 slots available to Army personnel engaged in acquisition or acquisition support roles. The ASSIP allocated the slots equitably among the PEOs, PMOs, and SECs. Due to the nature of the coursework, the more advanced ATAM Evaluator and Leader courses had a limit of 15 students. To better serve the specific needs of the Army, the ASSIP made those slots available to the SECs first because they are positioned to provide evaluation support across many programs. The PEOs and PMOs took advantage of the few slots in the courses not filled by SEC personnel.

The training program enjoyed strong participation, a good indication of both need and interest within the Army acquisition community. In fact, demand exceeded expectations and forced the waiving of class size restrictions in a few instances. Additionally, participation was broad: 9 of the 11 PEOs (including subordinate PMOs) and all of the Army's software centers had students who took part in the program. Sixty-four Army technical professionals attended at least part of the curriculum, with most earning at least one certificate. Figure 1 summarizes these results.[4]

FIGURE 1. SUMMARY OF ASSIP ARCHITECTURE TRAINING RESULTS



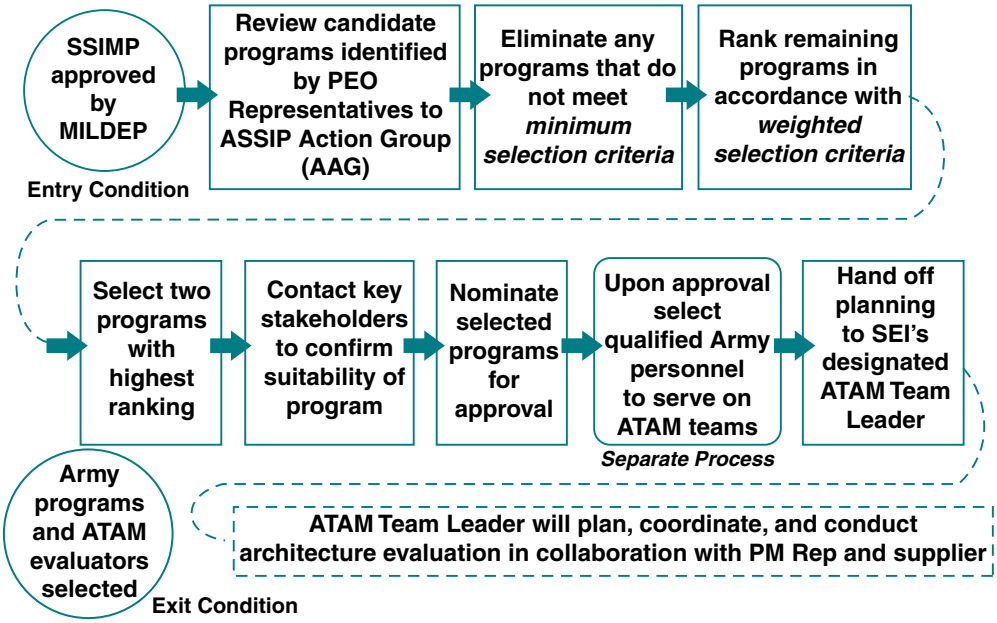
	TAR-DEC	PEO STRI	PEO MS	PEO IEW&S	PEO GCS	PEO EIS	PEO C3T	PEO AVN	PEO Ammo	JPEO CBD	C-E LCMC SEC	ARDEC SEC	AMRDEC SED
Software Architecture Professionals	4	2	2	2	1	1	3	2	2	1	8	7	7
ATAM Evaluators	7	0	0	0	1	0	1	0	0	0	5	7	9
Total Participants	8	2	4	2	2	1	5	3	3	2	12	9	11

OPPORTUNITIES FOR PRACTICE

Obviously, training is a necessary step toward building a skill level. However, training alone is not sufficient; to truly develop competence, trainees must be able to practice their newly acquired skills. To that end, the ASSIP Software Architecture Initiative added a limited skill-building component in FY05. The initiative sponsored several ATAM-based software architecture evaluations, and as a prerequisite, required the participation of trained Army evaluators on the evaluation teams.

As one might imagine, more programs were nominated for ATAM evaluations than could be accommodated, which made selecting among them a non-trivial task. As Figure 2 shows, once programs had been nominated, two experienced SEI staff members followed a process to rank the programs based on a set of criteria developed for ASSIP. The process consisted of two passes. The first pass pre-screened the nominated programs to ensure that they were ready and able to participate in an ATAM evaluation. Those that were not were eliminated from further consideration. The second pass ranked the remaining programs, with preference given to those where the

FIGURE 2. SELECTION PROCESS FOR PROGRAMS PARTICIPATING IN ASSIP-SPONSORED ATAMS



potential impact to the Army would be relatively higher and that presented the best opportunities to promote broad application of architecture practices (Blanchette & Bergey, 2007).

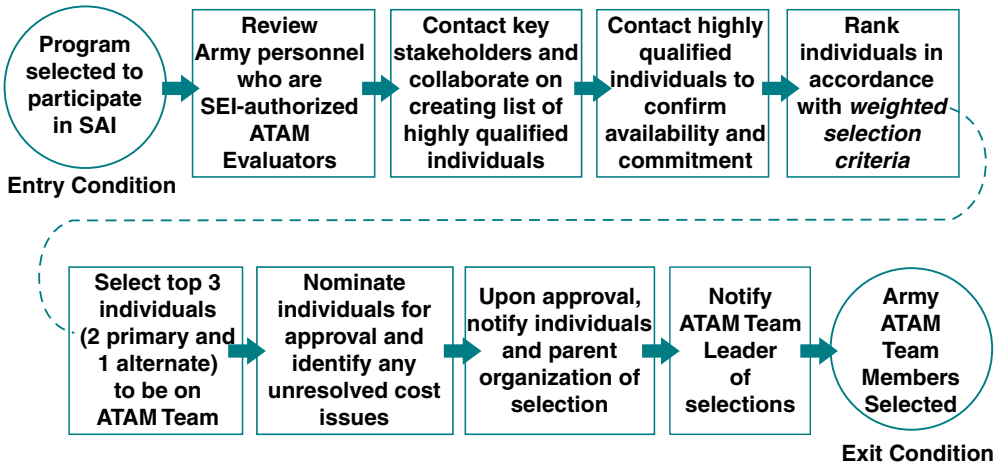
No less challenging was the effort to select evaluation team members from the pool of newly trained Army personnel; there were far more volunteers than slots available. The SEI used a process similar to the program selection process to select Army participants for the evaluation teams. Figure 3 depicts the participant selection process. Screening criteria for Army participants emphasized general technical competence in architecture practices as well as domain knowledge in the type of systems being evaluated. Ranking criteria focused on individuals who had knowledge of the program to be evaluated and who planned to become ATAM Lead Evaluators (because ATAM participation is one of the steps to becoming a Lead Evaluator) (Blanchette & Bergey, 2007).

To ensure the integrity of the ATAM, each Lead Evaluator had the final vote on whether a program was ready for an evaluation and whether the selected Army team members were adequately prepared. From FY05 through FY06, five ASSIP-sponsored evaluations provided seven students with the opportunity to hone their skills.

TAKING STOCK—A WORKSHOP

A key factor in any training program is evaluating effectiveness. Obviously, educating 30 technical professionals per year and providing practice opportunities for only one quarter of them[5] represents only a small step towards building and sustain-

FIGURE 3. SELECTION PROCESS FOR ARMY PERSONNEL PARTICIPATING ON ATAM TEAMS



ing a significant software architecture capability within the Army. Thus, gauging the relative success of the training and determining the necessary follow-on actions were important next steps in the process.

In May 2007, the ASSIP conducted a software architecture workshop designed to determine the need for additional investment and actions in order to develop a sustained and truly organic software architecture capability within Army acquisition. Hosted by the SEI, the workshop enjoyed strong attendance, with nearly 20 government representatives participating. Moreover, participation was broad—5 of the 11 PEOs were represented by at least one of their programs, and two of the Army's software engineering centers were represented, as was the office of the ASA(ALT). In addition, both the office of the assistant secretary of the Navy for research, development, and acquisition chief engineer, and the office of the secretary of the Air Force for acquisition participated in the workshop.

The workshop allowed participants to learn about recent developments in software architecture and to hear from each other about their respective experiences in applying software architecture practices. More importantly, participants actively discussed the enablers and barriers to broadening adoption of software architecture practices within the Army and brainstormed the necessary steps to achieve that broadening.

Workshop attendees eagerly discussed their ideas and suggestions, several of which centered on training. One particularly hot topic was the DoD Architecture Framework (DoDAF). The DoDAF describes a six-step, data-focused approach to developing system architectures. Attendees expressed a need to understand how they could evaluate DoDAF architecture products in a rigorous manner. They also indicated a desire to understand the linkage between DoDAF architectural views and software architectures (Bergey et al., 2007).

Since it is desirable to be proactive and introduce software architecture practices early in the acquisition process so that they are appropriately applied and coordinated throughout the development life cycle, attendees suggested that training for contract

officers would be beneficial. Doing so would allow them to be familiar enough with the risk reduction concepts to incorporate many of the practices into a request for proposal or contract at the beginning of an acquisition. Attendees also noted that while training was good and necessary, it needed to be augmented with guidelines and support materials that would help government personnel apply the knowledge effectively (Bergey et al., 2007).

Finally, several attendees voiced the need to have software architecture practices incorporated into the Defense Acquisition University (DAU) curriculum in order to promulgate the practices more widely and ultimately achieve risk reduction in the software component of system acquisitions across the Services (Bergey et al., 2007). It should be noted that DAU does address software architecture practices and issues in its intermediate and advanced courses in software acquisition management; workshop attendees simply felt that a greater degree of technical depth was needed in those courses.

SOME LESSONS LEARNED

Based on results to date of the ASSIP Software Architecture Initiative, several lessons may be gleaned for organizations contemplating a similar educational program.

TRAINING

The training curriculum itself was the simplest part of the initiative to implement since it already existed in a format that lends itself to this application. In hindsight, though, the manner in which the training was offered and the overall planning for it could have been improved.

The Lead Evaluator course should have been offered only to those individuals who not only had an interest in becoming Lead Evaluators but who had the support of their organizations in satisfying all the criteria.

For instance, the initiative made the entire software architecture curriculum, including the ATAM Lead Evaluator course, available to all participants. However, actually becoming a Lead Evaluator requires satisfaction of several criteria beyond simply attending the course. In particular, candidates must satisfy course instructors that they possess not only the requisite technical skills but also the necessary leadership acumen to be an effective Lead Evaluator. They must have participated on an evaluation team. They must also undergo observation while leading an ATAM evalu-

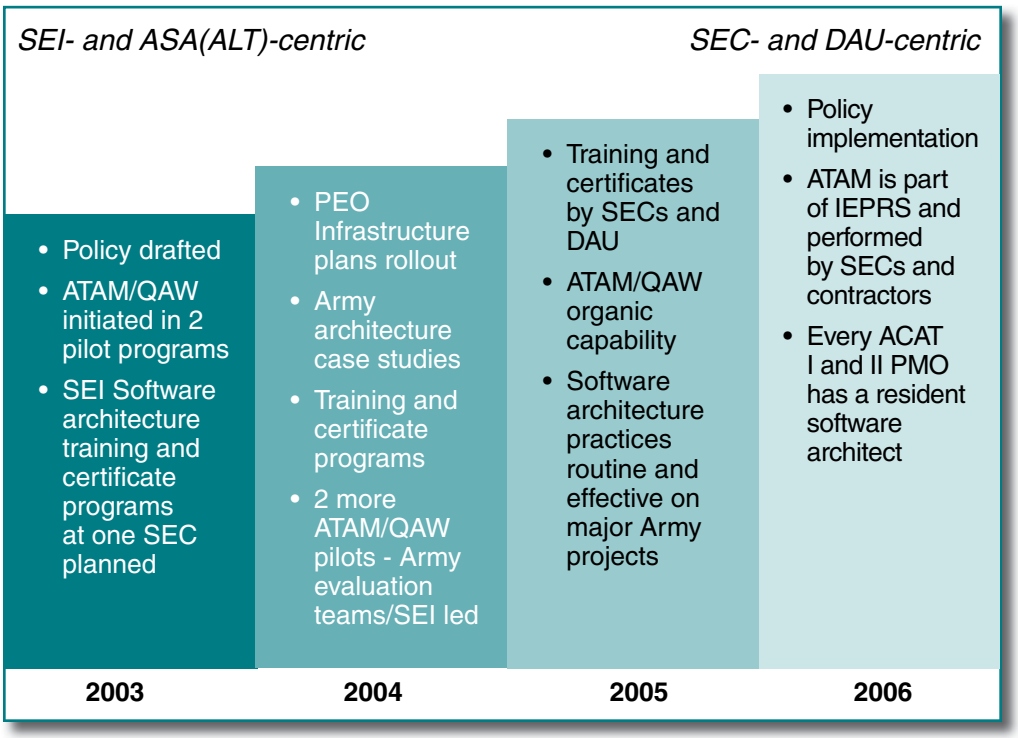
ation. Since there is a fee for the observation, the commitments of each individual's organization and supervisor are also required. Through the initiative, however, individuals were permitted to take the ATAM Lead Evaluator course without consideration for commitment to follow through with these additional steps. The Lead Evaluator course should have been offered only to those individuals who not only had an interest in becoming Lead Evaluators but who had the support of their organizations in satisfying all the criteria. An effective means of establishing organizational commitment is for prospective Army Lead Evaluators to include participation on an ATAM evaluation team (as an observed Lead Evaluator or as a regular team member) in their Individual Development Plans (IDPs).

As originally envisioned in FY03 and depicted in Figure 4, the ASSIP Software Architecture Initiative would have transitioned responsibility for the architecture training from an ASSIP-sponsored effort to the Army SECs and DAU by FY05. In retrospect, that schedule was too ambitious. Realistically, it was necessary to build a small cadre of trained professionals first and then demonstrate the utility of their training in order to develop the sort of groundswell of interest in software architecture practices that would support such a transition. In addition, transitioning the curriculum to external organizations requires careful planning.

PRACTICE

Arranging practice opportunities for students was one of the more challenging aspects of the initiative. For instance, although all of the nominated programs were in-

FIGURE 4. ORIGINAL TIMELINE



terested in receiving a free software architecture evaluation, a few were initially hesitant about having personnel from their PEO or SEC participate, while others objected to having Army personnel from unrelated or external commands on the team. Still others did not want the participation of *any* Army personnel. Although the objections were handled during the personnel selection process, the requirement for Army personnel to be involved needed to be more explicit in the program nomination process.

A focus of the ATAM evaluation is the actual on-site meetings, and it is easy to forget the other activities, such as pre-evaluation teleconferences and post-evaluation report development, that draw on a participant's time. These tasks are as essential to a successful outcome as the evaluation meetings, but are easily overlooked if there is a lengthy period between taking the ATAM Evaluator class and participating on a team. There were a couple of instances of misunderstanding about these points when recruiting participants for the ATAM evaluations. Although they were resolved without difficulty, reinforcing the requirements for participation up front, which is now part of the recruitment process, is a better approach.

The key difference between acquisition organizations is the manner in which they perceive software in their systems.

Finally, having the flexibility to adjust plans when situations dictate a change is essential; it is not in anyone's interests to cancel the limited practice opportunities. If a program selected for an architecture evaluation turns out not to be ready for the evaluation as planned, fallback options can be explored. For instance, if the architecture is not yet matured to a state where an evaluation would make sense, a quality attribute workshop, in which a team works with the program and its stakeholders to develop and prioritize non-functional requirements, might be substituted. The benefit of this approach is that evaluation team members are still able to practice techniques that they have learned through training, because eliciting quality attributes is an important step in an ATAM evaluation. Alternatively, if a program's architecture is sufficiently mature but not adequately documented for an evaluation, it is possible to postpone the evaluation while working with the program to improve its documentation.

PARTICIPATION

Not surprisingly, the SECs, due to their explicit focus on software, had the highest participation in the training program. The acquisition organizations were distributed relatively evenly in their course attendance, but at a much lower level than the SECs. When it came to nominating programs for architecture evaluations, however, those organizations that acquire software systems, communications systems, or electronics were more inclined to take advantage of the opportunity than those organizations that

chiefly acquired weapons systems (despite the fact that weapon systems were likely to contain large amounts of software, communications, and electronic components).

The key difference between acquisition organizations is the manner in which they perceive software in their systems. Weapon systems acquirers tend to focus on the system in its totality; they view software as an *enabler* rather than a *driver* of system behavior, and perceive it as relatively less important. Acquirers whose systems are dependent on software for functionality were quick to appreciate the importance of software and the need for software architecture evaluations. These differences suggest that extra effort is necessary to reach out to organizations that tend to treat software as a less important implementation detail in their systems.

In the case of the Army, the workshop validated the need and importance of the architecture training program while also demonstrating a need to expand Army investment in it.

Perhaps the most significant outcome of the workshop was that a number of program offices indicated that they were only willing to pursue evaluation of their software architectures because ASSIP paid for the evaluations, yet *all* recognized the value of the evaluation afterwards (Bergey et al., 2007). The ability to turn "nay-sayers" into "yea-sayers" is powerful evidence of the architecture initiative's success. However, such findings also suggest that similar education programs must be prepared to counter lack of awareness among program managers about architecture evaluations through funding, policy, or both.

Additionally, it is important to evaluate the effectiveness of the program. In the case of the Army, the workshop validated the need and importance of the architecture training program while also demonstrating a need to expand Army investment in it.

WAY AHEAD

The success of the software architecture initiative led to a number of tasks for ASSIP in FY08 that seek to build on successes as well as to address the lessons learned and workshop results. Among the tasks in progress are the continued offering of the software architecture curriculum, the expansion of opportunities for students to apply techniques in several ASSIP-funded ATAM evaluations, and the introduction of opportunities for selected promising students to advance toward becoming SEI-certified ATAM Lead Evaluators. Additionally, a new course, aimed at acquisition executives to increase their awareness of the benefits of the disciplined use of software architecture practices, is being developed. The ASSIP also will conduct interviews of

program office personnel to collect lessons learned and develop case studies regarding software architecture practices, and will hold a workshop to explore and make clearer the relationships amongst different kinds of architecture, including software, systems, and enterprise architectures. Lastly, the SEI will collaborate with DAU in seeking opportunities to enhance the software training available to the Army (and DoD) technical workforce.

SUMMARY

There is no shortage of reasons for wanting to improve the technical skills of the government's acquisition practitioners, especially for those individuals who acquire software-intensive systems. Sound software architecture practices are widely recognized as helpful in developing such systems successfully, yet they represent one of the key areas where government expertise is lacking.

Through the ASSIP Software Architecture Initiative, the U.S. Army has succeeded in training a cadre of its acquisition professionals in the necessary skills to understand and evaluate software architectures. The initiative provided hands-on experience opportunities in addition to the classroom-based training.

In assessing the results of the Software Architecture Initiative, clearly more work remains to be done to achieve a truly organic software architecture capability in the Army. That work is underway as the Army continues its emphasis on improving the skills of its acquisition workforce.

Mr. Stephen Blanchette, Jr., is a senior member of the technical staff at the Software Engineering Institute, specializing in acquisition improvement initiatives for the U.S. Army. A graduate of both Norwich University and Embry-Riddle Aeronautical University, he has over 20 years' experience in the defense industry. He is a senior member of the American Institute of Aeronautics and Astronautics and the Institute of Electrical and Electronics Engineers.

(E-mail address: sblanche@sei.cmu.edu)

Mr. John Bergey is a senior member of the technical staff at the Software Engineering Institute, specializing in transitioning SEI product line and architecture-centric practices into acquisition practice across the Armed Services. Mr. Bergey is a graduate of Pennsylvania State University and the University of Michigan. Prior to joining SEI, he served over 25 years as a software division manager with the U.S. Navy.

(E-mail address: jkb@sei.cmu.edu)

AUTHOR **BIOGRAPHY**

ENDNOTES

1. Claude Bolton, ASA(ALT) retired in January 2008, but the ASSIP continues under Dean Popps, Acting ASA(ALT).
2. According to the Defense Acquisition University (DAU), a software-intensive system is one in which software represents the largest segment in one or more of the following criteria: system development cost, system development risk, system functionality, or development time (Defense Acquisition University, 2005).
3. Three certificates—Software Architecture Professional, ATAM Evaluator, and ATAM Lead Evaluator—are available to students who complete the required courses.
4. Participants shown for PEO Missiles and Space were part of the predecessor organizations PEO Tactical Missiles and PEO Air Space and Missile Defense.
5. Only a limited number of contractual engagements provided a suitable opportunity to apply the skills learned in collaboration with the development contractor without being intrusive (i.e., significantly disrupting cost and schedule).

REFERENCES

Anderson, W., Bergey, J., Fisher, M., Graettinger, C., Hanson, W. J., Obenza, R., et al. (2001, November). *Army workshop on lessons learned from software upgrade programs.* Special Report (CMU/SEI-2001-SR-021). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

Bass, L., Clements, P., & Kazman, R. (2003). *Software architecture in practice* (2nd ed.). Boston: Addison-Wesley.

Bass, L., Nord, R., & Wood, B. (2006, April). *Risk themes from ATAM data: Preliminary results.* Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

Bergey, J., Blanchette, S., Gagliardi, M., Klein, M., Little, R., Morrow, T., et al. (2007). *Army software architecture workshop: Reporting on Army experiences.* Special Report (CMU/SEI-2008-SR-003, Limited distribution). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

Blanchette, S. (2005, March). *U.S. Army acquisition – the Program Executive Officer perspective.* Special Report (CMU/SEI-2005-SR-002). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

Blanchette, S., & Bergey, J. (2007, June). *Progress toward an organic software architecture capability in the U.S. Army.* Special Report (CMU/SEI-2007-TR-010). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

Charette, R., McGarry, J., & Baldwin, K. (2003, January). *Tri-Service assessment initiative phase 2 systemic analysis results.* Tri-Service Assessment Initiative. Retrieved August 1, 2007, from http://www.splc.net/programs/acquisition-support/conf/2003-presentations/charette.pdf

Clements, P., Bergey, J., & Mason, D. (2005, September). *Using the SEI Architecture Tradeoff Analysis Method to evaluate WIN-T: A case study.* Special Report (CMU/SEI-2005-TN-027). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

Clements, P., & Bergey, J. (2005, September). *The U.S. Army's Common Avionics Architecture System (CAAS) product line: A case study.* Special Report (CMU/SEI-2005-TR-019). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

Defense Acquisition University. (2005, July). *Glossary of defense acquisition acronyms and terms* (12th ed.). Fort Belvoir, VA: DAU Press.

Defense Science Board Task Force. (1994, June). *Acquiring defense software commercially.* Retrieved August 1, 2007, from http://www.acq.osd.mil/dsb/reports/commrecialdefensesoftware.pdf

Defense Science Board Task Force. (2000, November). *Defense software.* Retrieved August 1, 2007, from http://www.acq.osd.mil/dsb/reports/defensesoftware.pdf

Erwin, S. I. (2007, April). Pentagon begins broad review of acquisition workforce skills. *National Defense, 91*(641), 18.

Government Accountability Office. (2007, June). *Defense acquisitions: Role of lead systems integrator on future combat systems program poses oversight challenges.* (GAO-07-380). Washington, DC: Author.

Government Accountability Office. (2008, January). *Defense acquisitions: DoD's increased reliance on service contractors exacerbates long-standing challenges.* Statement of David M. Walker, Comptroller General of the United States, in testimony before the Subcommittee on Defense, Committee on Appropriations, House of Representatives. (GAO-08-621T). Washington, DC: Author.

Kasunic, M. (2004, March). *Army Strategic Software Improvement Program (ASSIP) survey of Army acquisition managers.* Special Report (CMU/SEI-2004-TR-003). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

Keeler, K. L. (2005, October). *U.S. Army acquisition—The program office perspective.* Special Report (CMU/SEI-2005-SR-014). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.