Software Product Lines: Experiences from the Sixth DoD Software Product Line Workshop

John Bergey
Sholom Cohen
Lawrence Jones
Dennis Smith

March 2004

Product Line Practice Initiative

Technical Note
CMU/SEI-2004-TN-011

This work is sponsored by the U.S. Department of Defense.

The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

Contents

List of Figures

Abstract

The Carnegie Mellon® Software Engineering Institute held the Sixth Department of Defense (DoD) Product Line Practice Workshop in September 2003. The workshop was a hands-on meeting to share DoD product line practices, experiences, and issues and to discuss ways in which specific product line practices are accomplished within the DoD. Participants reported encouraging progress on DoD software product lines. Additionally, participants addressed some important implementation questions. This report synthesizes the workshop presentations and discussions.

1 Introduction

1.1 Product Line Practice

An increasing number of organizations are realizing that they can no longer afford to develop multiple software products one product at a time: they are pressured to introduce new products and add functionality to existing ones at a rapid pace. They have explicit needs to achieve large-scale productivity gains, improve time to market, maintain a market presence, compensate for an inability to hire, leverage existing resources, and achieve mass customization. Many organizations are finding that the practice of building sets of related systems together can yield remarkable quantitative improvements in productivity, time to market, product quality, and customer satisfaction. Those organizations are adopting a product line approach for their software systems.

A *software product line* is a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way [Clements 02a].

In January 1997, the Carnegie Mellon[®] Software Engineering Institute (SEI[SM]) launched the Product Line Practice Initiative to help facilitate and accelerate the transition to sound software engineering practices using a product line approach. The goal of this initiative is to provide organizations with an integrated business and technical approach to systematic reuse, so they can produce and maintain similar systems of predictable quality more efficiently and at a lower cost.

A key strategy for achieving this goal has been the creation of a conceptual framework for product line practice. The SEI *Framework for Software Product Line Practice*[SM] (henceforth referred to as the framework) describes the foundational product line concepts and identifies the essential activities and practices that an organization must master before it can expect to field a product line of software or software-intensive systems successfully. The framework organizes product line practices into practice areas that are categorized according to software engineering, technical management, and organizational management. (These categories represent disciplines rather than job titles.) The framework is a living document that is evolving as experience with product line practice grows. Version 4.0 is described in the book *Software Product Lines: Practices and Patterns* [Clements 02a], and Version 4.1 is available on the SEI's Web site [Clements 02b].

[®] Carnegie Mellon is registered in the U.S. Trademark and Patent Office.
[SM] SEI is a service mark of Carnegie Mellon University.
[SM] Framework for Software Product Line Practice is a service mark of Carnegie Mellon University.

The framework's contents were based on information-gathering workshops,[1] extensive work with collaboration partners, and continued research. The SEI has also incorporated practices reported at its two international Software Product Line Conferences (SPLC1 and SPLC2) [Donohoe 00, Chastek 02] and from the community.

In March 1998, the SEI hosted its first Department of Defense (DoD) product line practice workshop, *Product Lines: Bridging the Gap–Commercial Success to DoD Practice.* Product line practices, DoD barriers and mitigation strategies, as well as similarities and differences between DoD product line practice and commercial product line practices were discussed and documented [Bergey 98]. Subsequent workshops were held in successive years [Bergey 99a, Bergey 00a, Bergey 01], with the fifth workshop being held in conjunction with SPLC2 [Bergey 03a]. At all five DoD workshops, the SEI was encouraged to continue to hold other DoD workshop events and to continue to share best commercial and DoD practices through these forums.

One of the key outcomes of these workshops was the identification of product line practices that were particularly important to DoD acquisition organizations. This information supported development of a *companion* to the framework, titled *Software Product Line Acquisition: A Companion to A Framework for Software Product Line Practice* [Bergey 03b] (henceforth referred to as the companion). Similar to the strategy for the framework, the companion is a living document with the latest version available on the SEI's Web site.

1.2 About This Workshop

The goals of the Sixth DoD Product Line Practice Workshop in September 2003 were to

- share DoD product line practices, experience and issues, regarding both development and acquisition
- discuss ways in which the current gap between commercial best practice and DoD practice can be bridged
- explore ways to incentivize software product line practice in the DoD

All participants in this workshop were from the DoD acquisition and contractor community. They were invited based on our knowledge of their experience with and commitment to software product lines as either DoD system acquirers or DoD system contractors. Together, we discussed the issues that form the backbone of this report.

The workshop participants included

- John Bergey, Product Line Systems Program, SEI

[1] The results of these workshops are documented in SEI reports [Bass 97, Bass 98, Bass 99, Bass 00, Clements 01].

- Sholom Cohen, Product Line Systems Program, SEI
- Edward Dunn, Naval Undersea Warfare Center (NUWC), U.S. Navy
- Matthew Fisher, Software Engineering Process Management Program, SEI
- Richard Flesner, Rockwell Collins
- Carl Higgenbotham, U.S. General Accounting Office
- Lawrence Jones, Product Line Systems Program, SEI
- Bob Lyons, Joint National Integration Center (JNIC)
- Linda Northrop, Director, Product Line Systems Program, SEI
- Juan Pastrana, Program Manager (PM), Force XXI Battle Command Brigade and Below (FBCB2), U.S. Army
- Frank Polster, U.S. Army Training Support Center (ATSC)
- Mary Rich, The Aerospace Corporation
- James Scott, U.S. Army Aviation and Missile Command (AMCOM)
- Dennis Smith, Product Line Systems Program, SEI
- Daniel Stroka, PM FBCB2, U.S. Army
- James Wills, Argon Engineering Associates

1.3 About This Report

This document summarizes the presentations and discussions at the workshop. This report is written primarily for those in the DoD who are already familiar with product line concepts, especially those working on or initiating product line practices in their own organizations. Acquisition managers and technical software managers should also benefit from this report. Those who desire further background information are referred to the following publications:

- *Basic Concepts of Product Line Practice for the DoD* [Bergey 00b]
- *A Framework for Software Product Line Practice, Version 4.1* [Clements 02b]
- *Software Product Line Acquisition: A Companion to A Framework for Software Product Line Practice, Version 2.0* [Bergey 03b]
- *Software Product Lines: Practices and Patterns* [Clements 02a]

The remainder of this report is organized into three main sections that parallel the workshop format:

1. Section 2, DoD Software Product Line Experiences: A Digest of Participant Presentations

2. Section 3, DoD Software Product Line Practices: Working Group Reports, which details two working group reports that addressed issues of concern for DoD software product line practices

3. Section 4, Summary, which recaps the major themes of this report and suggests future directions

2 DoD Software Product Line Experiences: A Digest of Participant Presentations

2.1 Introduction

Rick Flesner, Rockwell Collins, gave a keynote presentation. In addition, all participants were invited to give a short presentation about their organizations' product line interest and experience. A summary of these presentations follows.

2.2 The Common Avionics Architecture System (CAAS) - Rick Flesner, Rockwell Collins

Rick Flesner's keynote presentation reported on progress in creating a software product line for the Army's special operations forces. The product line, based on the CAAS, supports an open systems architecture and common avionics software for the MH-47, MH-60, and MH/AH-6 helicopters (Figure 1). Rockwell Collins' approach uses technology proven on its commercial software product line approach that supports the Boeing 767, KC-135, and Collins Pro Line 21 for business jets.

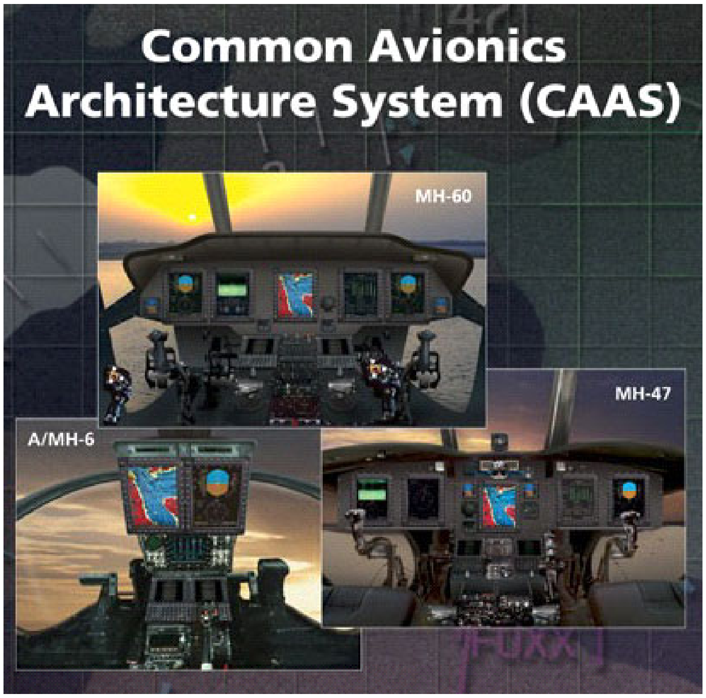

Figure 1: Helicopter Platforms Supported by the CAAS Product Line

The product line approach was motivated by the following problems the Army was experiencing with its special operations forces helicopters:

- inability of avionics to accommodate growth requirements
- high non-recurring engineering (NRE) costs to upgrade or add new functionality
- lack of preplanned avionics upgrades to support new functionality and/or component obsolescence
- unacceptable levels of aircraft availability
- high installation costs
- limited opportunity for third-party development
- diminishing manufacturing sources
- application of commercial off-the-shelf (COTS) software without adequate consideration of the military environment

To meet these challenges, the vision for the CAAS was twofold:

- to address obsolescence and modernization issues by creating a scalable system that would meet the needs of multiple helicopter cockpits
- to reduce the total cost of ownership by using a single, open, common avionics architecture system for all platforms

The CAAS embodies the following architectural precepts:

- an Open System Architecture (OSA) using published and controlled interface definitions, such that its hardware and software components can be replaced or upgraded with alternate components
- variability isolation to accommodate changes in the system over its life cycle, such that the impact of change is isolated to the smallest system component
- use of layers and partitions with widely accepted interfaces to isolate system components
- redundant software using master/slave protocol where every application is resident on every box and some applications are active on multiple boxes to support quality attributes such as availability and portability
- use of application templates across application, common software, and common reusable elements (CoRE) to support reusability, modifiability, repeatability, and affordability
- use of commercial standards including ARINC 661 (cockpit display system interface standards), POSIX (portable operating system interface), CORBA (Common Object Broker Request Architecture), IEEE P1386/P1386.1 (common mezzanine card families draft standards), OpenGL (graphical interfaces standards), and DO 178B (software considerations for airborne systems) to enhance portability, maintainability, and modifiability

- a virtual machine operating system (VMOS) based on the flight-ready POSIX operating system (OS) (with standard POSIX application program interface [API] and Ada 95 support) to encapsulate and manage any interaction with the computing platform and provide Level A design assurance

Other features being supported include a high-integrity, Ethernet local area network (LAN) with COTS general-purpose processors, video processors, and graphics engines.

The architectural impact on the total life-cycle cost is that it will reduce the NRE costs of upgrading or adding new functionality and allow the use of all available processing reserves when upgrading the system without drastically rerouting system data. A software application developer's toolkit will be available to third parties so they can integrate their software applications. This toolkit includes

- references to COTS APIs and COTS toolkits
- a set of Collins APIs for interfacing to Collins-developed applications
- a set of system integration tools

The software product line has these strengths:

- It supports the insertion of new technology.
- It enables multifunction displays and other avionic equipment units to be swapped out from one helicopter avionics system to another with automatic reconfiguration.
- It accommodates the integration of subsystems by third-party developers through well-defined APIs.

Although Rockwell Collins has given the Army exclusive "government-use rights" to the avionics software, its business model has proven very successful in that the company subsequently won three of four large avionics hardware equipment buys.[2] The Army is planning to adopt the same software product line approach to fill the needs of the entire fleet of Army helicopters.

2.3 RangeWare - Ed Dunn, Naval Undersea Warfare Center (NUWC)

Ed Dunn presented the status of product line activity within NUWC Newport. With support from the SEI Product Line Practice Initiative, NUWC has defined an initial measurement program for the software product line development of Navy range systems. SEI support came from a joint Business and Acquisition Guidelines and Acquisition Support Program pilot. The NUWC measurement program includes

[2] The U.S. Army's Technical Applications Program Office (TAPO) was the government acquirer. The SEI supported TAPO during its early product line exploration and later after its contract was let to Rockwell Collins.

- an overall goal statement
- subgoals
- key questions that the measurement program will answer
- data elements
- collection plan
- reporting process

The SEI and NUWC jointly developed the RangeWare product line concept, and NUWC has successfully implemented it. NUWC's implementation includes

- the asset base, a collection of over 20 large-grained product line components
- a range system architecture
- a production plan for developing range systems

Several projects using RangeWare have already been fielded. Four new projects sharing the RangeWare asset base will be collecting data that answer questions relating to the effort estimation in a product line environment. Data is already being collected from some of the projects. Future data collection will look at the degree and cost of using specific product line assets and at user satisfaction. The results of the pilot will be described by Cohen, Dunn, and Zubrow in a forthcoming technical note.[3]

2.4 Lighthouse - James Wills, Argon Engineering Associates

Argon Engineering is a rapidly growing systems engineering and development company providing full-service information solutions to a wide range of customers. Argon provides creative state-of-the-art technology solutions to difficult system problems. Current challenges include the design and development of communication systems that search, identify, and capture signals. Argon's work includes sensor development, data collection and decision support, and the analysis and design of information retrieval and visualization techniques [Argon 04].

Argon uses product line methods to develop and deploy many of its systems. It has seen the following benefits from the product line approach:

- shorter development schedules
- lower development and upgrade costs
- lower total ownership costs
- support for an incremental development model

[3] Cohen, S.; Dunne, E.; & Zubrow, D. *Case Study: A Measurement Program for Product Lines.* Pittsburgh, PA: Software Engineering Institute, to be published.

- shared technology costs
- best-in-class COTS/government off-the-shelf (GOTS) components
- continuous technology insertion

Argon's product line approach includes an architecture with defined modules for basic signal-processing capabilities and provides multiple components to provide flexible implementation for each module. Argon has fielded six programs using product line assets, and, over the five-year period of the product line, has expanded the scope of the product line covered by assets. As new programs develop capabilities, they become new assets for other programs. Argon metrics show that developing assets increases costs by approximately 50%. In addition, the metrics estimate a 25% reuse cost. In spite of these costs, Argon reports payback with the third use of an asset. These data confirm those reported by the SEI in other product line case studies.

2.5 Avionics Architecture Description Language (AADL) - James Scott, U.S. Army Aviation and Missile Command (AMCOM)

Jim Scott of AMCOM presented the results of his investigation into a design language for aviation systems—the AADL. The SEI and AMCOM collaborated in the development of the AADL. While primarily a design language, the AADL incorporates many concepts that are critical to successful product line deployment. The AADL has been proposed as an international standard to the Society of Automotive Engineers (SAE).

Developers of avionics systems use the AADL to support the model-driven architectural design and specification of performance-critical computer hardware/software systems and components. These systems are characterized as real-time, high-reliability, fault-tolerant, and safety-critical applications. In addition, the language supports product line design for a range of applications and analyses in a generic, flexible, and extensible fashion. The AADL directly addresses the problems of the impact of change on component-based systems, systems of systems, plug and play, system evolution, and overall life-cycle reusability. Using the AADL in a reengineering effort results in a 50% cost reduction over traditional development approaches.

Although the primary focus of the AADL is to support embedded real-time application domains, the language's generic architectural specification can be useful over a wide range of performance-critical product line mission areas including integrated/interoperable command and control (C2) systems. The AADL offers the following advantages over traditional design approaches:

- AADL designs can be analyzed by automated tools to assess impacts, analyze multiple issues, and validate requirements.

- The designs support “life-cycle reusability” as a key element of the product line approach.

2.6 Mary Rich, The Aerospace Corporation

The Aerospace Corporation sees great potential for product lines in its ground-based satellite control systems and plans to have a special software product line session at its annual Ground Systems Architecture Workshop (GSAW) this year. The corporation is still in the early stages of promoting software product line adoption and is exploring acquisition strategies to promote the practice among its contractor community. The corporation is also interested in addressing cultural issues, eliminating stovepipe approaches, and creating incentives that will offset the perception of revenue loss in software maintenance contracts. In terms of next steps, Aerospace is interested in conducting a software product line technical probe to get a better handle on its strengths and weaknesses with a view towards how best to launch a product line acquisition.

2.7 Frank Polster, Army Training Support Center (ATSC)

The ATSC reported that significant organizational improvements were made following the use of the SEI Product Line Technical Probe[SM] (PLTP[SM]), the subsequent planning sessions to address the PLTP findings, and the use of the Architecture Tradeoff Analysis Method[SM] (ATAM[SM]). The ATSC sees product line technology as the way to embed training systems in Army combat systems.

2.8 Dan Stroka, Force XXI Battle Command Brigade and Below (FBCB2)

With the help of the SEI, FBCB2 is developing a software product line architecture to support the rapid configuration of its system that supports C2 and situational awareness. After-action reports from Operation Iraqi Freedom provided glowing testimonials to the usefulness of FBCB2. The product line approach promises to enable FBCB2 to provide greater support more efficiently and cost effectively over a broader range of platforms.

2.9 Bob Lyons, Joint National Integration Center (JNIC)

The JNIC has recently expanded its mission from missile defense wargaming and exercise support to add the analysis, test, and integration of actual missile defense systems. The JNIC is exploring a software product line approach to addressing the complexity of this expanded mission.

[SM] Product Line Technical Probe, PLTP, Architecture Tradeoff Analysis Method, and ATAM are service marks of Carnegie Mellon University.

3 DoD Software Product Line Practices: Working Group Reports

Following the plenary presentations, workshop participants were divided into two working groups. Each group selected discussion topics from a proposed list of questions important to DoD product line practice. The groups then reconvened and presented the results of their discussions. The following sections summarize the work of the two groups.

3.1 Group A

Group A discussed four questions. Each one is shown below along with a summary of its subsequent discussion.

3.1.1 Question 1

In light of the concept of maturing technology before giving that technology over to a program office for acquisition

- Should program offices buy from a product line or own it?
- If the product line is government owned, in what organization should product line work be managed or performed? research and development (R&D) organizations or program offices?

Discussion

The group felt there was no single answer to the question of product line ownership: it depends on the circumstances. Buying products from an existing product line creates incentives for suppliers and puts the supplier in charge of sustainment.[4] If sufficient demand exists, the government can drive (or influence) the market without having to be in the business of product line ownership.

Supplier ownership would be preferable if there were an affirmative answer to the question "Is the existing market sufficient to support mission requirements?" The following factors would contribute to an affirmative answer:

- a well-established, mature, dependable market

[4] One supplier reported successfully getting the government to provide funds for core asset sustainment once a relationship of trust was established.

- a mature domain with more likelihood of successful product lines
- a history of supplier continuity

However, if the risk to the mission were too great, government ownership would be preferred. In this case, the group felt that ownership by an R&D organization was inappropriate for at least two reasons. First, if the domain is mature enough for a product line, it shouldn’t be an R&D domain. Second, core competencies of R&D organizations do not generally include fielding and supporting operational systems.

Once the government owns a product line, it must be concerned with supporting a broader range of stakeholders than in a typical acquisition. The acquisition organization must have mechanisms to consider requirements that may span many organizations.

Whether the decision is to buy from an existing product line or to create and own one, the rationale should be documented. Circumstances may change, and thus the decision may have to be reconsidered.

3.1.2 Question 2

What can the DoD do to get commercial suppliers to build tools that would support product line development needs?

Discussion

The “simple” answer to the question is “Create a big enough market.” The DoD has the same need for tools as industry. One unique contribution the government could make is to sponsor tool building. The government could also help identify areas where existing tools may be tailored to support product line needs.

The rest of the discussion identified areas where existing tools fell short for product line support:

- Requirement-tracking tools are insufficient for handling product line needs.
- Configuration management complexity for product lines is not handled directly. Variations in time are handled, but not variations in space.
- Integrated environments provide some capability, but you must take the whole environment regardless of whether you need it all.
- There is a need for improved interfacing between development tools and planning tools.

3.1.3 Question 3

How do we organize and execute a measurement program for product lines?

Discussion

During this practical discussion, Ed Dunn discussed NUWC’s efforts to establish a “goal-driven” measurement program[5] for its software product line. Discussion points included the following:

- It is important to obtain management buy-in for your measurement program. You need to have someone who is responsible for total life-cycle ownership costs across the product line see results; he or she will supply muscle to support the effort.
- The information you need may be inconveniently scattered across various places in the organization.
- Initially, NUWC’s stakeholders had a greater need for reliable estimates (e.g., cost and schedule) than for showing cost savings to justify the product line.
- Cost tracking was critical for NUWC’s contractors.

When group members were asked whether there was a standard set of useful product line measures, the answer was no. To collect useful data, you should set your product line goals and then let them drive the measures you need.

3.1.4 Question 4

Where X is one of the following, how do I know that my supplier is doing a good job of product line practice X?

- requirements engineering
- product line scoping
- software architecture definition and evaluation
- component development
- software system integration and testing

Discussion

A general remark was that some of the activities might be outsourced, but it might not make sense in all cases. For example, there was sentiment that if the government owned the product line, it might not make sense to outsource requirements engineering and product line scoping.

[5] A goal-driven measurement program identifies and defines software measures to support an organization’s business goals [Park 96].

For those activities that are outsourced, a good metrics program can help define what a “good job” is. You want to remove as much subjectivity as possible.

The following points were made regarding performance evaluation for other practices:

Architecture Definition and Evaluation

- You should count on architecture definition being iterative and requiring refinement. This should be built into the supplier’s plan.
- Determine whether the supplier has architecture-centric processes and design practices.
- Determine whether the architecture and interfaces are well defined and extensible where product line variation is required.
- Determine whether the supplier conducts architecture reviews as part of its internal design process. Participate in the reviews as an observer. How well are the necessary quality attributes defined and analyzed? Do they include qualities important to product line support?
- If a software architect is identified, talk with him or her. Good software architects recognize a peer and often welcome external reviews.

Component Development

- Determine whether developers follow the architecture. Do they follow the interface control documents?
- Test to determine whether the component works. Does it work correctly? Does it work the same way every time?
- Examine the effectiveness and institutionalization of the configuration management process.
- Determine how well requirements traceability is performed.
- Determine whether the tool support is adequate.
- Does the supplier have a “hero-based” culture? (If so, that is a bad indicator.)

Integration and Test

- Consider the ideas given above under “Component Development.”
- Determine how well problems are tracked.
- Determine how well regression testing is performed.

In conclusion, as you focus on the product line practices, don’t lose sight of the quality of the end product.

3.2 Group B

Group B discussed five questions. Each one is shown below along with a summary of its subsequent discussion.

3.2.1 Question 1

This question consisted of three sub-questions regarding the adoption of a software product line approach by a DoD organization:

1. What's in it for the supplier?
2. What is needed to incentivize suppliers?
3. What incentives can the government provide to encourage suppliers to propose a product line approach?

Discussion

The major benefits the group saw for suppliers centered on the multiple business opportunities that would open up once the first system was delivered and operationally deployed. After moving to a product line approach, a supplier would become more efficient and be in a better position to realize a big payoff downstream. The supplier might have to give up some proprietary rights, but that shouldn't infringe on its commercial rights to the software. Key factors would be a supplier's ability to leverage existing software assets and whether it sees a product line as an opportunity to increase its competitive leverage in the marketplace.

The government can incentivize and encourage suppliers in many ways, but the best approach would be for it to develop a compelling business case that offers a substantial "carrot" beyond the delivery of the initial system. In developing the business case, the government should explore potential synergy with commercial products and develop a realistic budget and schedule that include the additional up-front resources a product line approach requires.

Additional incentives suggested by the group were

- use a cost-plus contract for developing core assets
- make a product line approach a hard requirement in the request for proposal (RFP) or contract
- include product line experience as a sub-factor in the technical evaluation criteria
- include software life-cycle sustainment in the scope of the RFP/contract
- include both development and production competition in the scope of the RFP or contract

- include options in the contract that would allow other programs to use the product line architecture and other core assets

One overriding consideration is adopting a realistic budget, especially when the application domain is not closely aligned with product offerings in the commercial market.

3.2.2 Question 2

What are the government's cultural, organizational, and regulatory barriers? How can they be addressed?

Discussion

Although group members agreed there were some significant barriers to adopting a product line approach, they felt those barriers could be overcome. The main obstacle is that a program manager (PM) gets "gold stars" for on-time, on-cost delivery. As a result, PMs typically don't want to introduce any changes or unknowns that might impact how they conduct their acquisitions, especially those likely to increase the cost of delivering the first system. Another major barrier is that the budget process is clearly a "waterfall" process geared to developing stovepipe systems.

The group suggested two ways such barriers could be addressed:

1. Develop new guidelines to support product line acquisitions.
2. Adopt a lead system integrator (LSI) contracting strategy.

The new guidelines would be based on creating a business case for a product line that encompasses the entire life cycle. Since the total cost of ownership would be reduced by a product line approach, the group was convinced that if a PM were held accountable for life-cycle costs, a product line business case would be most reasonable. The new guidelines would also have to address changing the budgeting process to accommodate iterative system development and allow for more flexibility in allocating funds to overcome restrictions such as those imposed by different "colors" of money.

The group also felt that an LSI contracting strategy, which is gaining in popularity in the DoD, would favor a product line approach because an LSI is in an ideal position to

- easily acquire core assets from multiple suppliers
- scope the product line
- serve as the chief architect
- define the variability that the product line will need to accommodate
- develop an overarching architecture and enforce compliance
- develop and enforce guidelines for software components and other core assets

- assume the role of product developer and integrator

A major advantage LSIs have is that they have more flexibility in contracting with potential suppliers, establishing teaming relationships, allocating funding, and providing technical direction to participating suppliers.

3.2.3 Question 3

This question consisted of two sub-questions:

1. To achieve product line success, what do I (as a supplier) want in a program office?
2. What do I (as an acquisition organization) want in a supplier?

Discussion

The group felt strongly that a common element to the success of a product line from both the perspective of the supplier and program office is to focus on developing a good partnering agreement that

- is non-adversarial
- represents a “win-win” strategy
- builds mutual confidence and trust
- enables suppliers to make a profit
- promotes good communication
- aligns with and leverages a supplier’s own R&D initiatives
- involves suppliers up front so they can contribute to the product line concept and scope

Things to avoid include creating a loose and ambiguous RFP or contract, insisting on a COTS-only solution, and mandating a business process.

The biggest impact, though, to a DoD organization is that product line adoption requires changing acquisition practices, such as

- adopting a spiral/evolutionary acquisition approach
- planning and budgeting for engineering change proposals (ECPs)
- ensuring that the first spiral is realistic and produces tangible results
- strongly emphasizing requirements management throughout the life cycle
- acquiring good documentation for core assets

These changes may require the program office to obtain additional technical management and software engineering support from external sources such as systems engineering technical assistance (SETA) contractors.

3.2.4 Question 4

How do we address the following acquisition strategy risks?

- limited management visibility
- organizational roles and responsibilities
- ownership and data rights
- architecture compliance mandates
- core asset sustainment
- product development and support

Discussion

The group thought that the visibility needed to manage and maintain the technical oversight of a product line acquisition included

- obtaining insight into cost and schedule
- having more deliverables included in the contract
- creating a robust metrics plan
- implementing a risk management strategy appropriate to a product line approach
- establishing a long-term view of managing costs, a schedule, and a solution (i.e., products)
- having a plan for getting all stakeholders involved from concept definition to deployment

Adopting an iterative approach could provide increased visibility and reduce risk. Developing an operational requirements document (ORD) and an initial set of “mission threads” to evaluate the architecture proposed by the contractor would be an appropriate starting point.

Since the acquisition organization may not have the management and technical skills to properly oversee a product line, a promising alternative is to delegate those responsibilities by contracting with an LSI. The LSI would also have integration responsibility—something the government is not particularly good at, but something with which system prime contractors have in-depth experience. This contractual arrangement should make it easier for the government to take advantage of a product line approach instead of kludging systems together.

From a data rights perspective, the best compromise is for the contractor to own the software and the acquisition organization to have exclusive “government-use” data rights. This avoids many of the problems typically associated with using proprietary items such as a contractor’s proprietary software architecture.

Mandating architecture compliance can be tricky from a liability standpoint. A practical solution is to have the contractor develop a plan for architecture compliance and verification; to include architecture evaluations and quality assurance (QA) reviews; and to develop appropriate architecture documentation.

Establishing a common software development environment was thought to be a critical infrastructure piece for core asset sustainment. Risks associated with core asset sustainment could be reduced by

- stressing systematic configuration management
- adopting open standards for software interfaces
- developing a plan up front for preplanned product improvements (P^3I)
- delegating responsibility for the management of each core asset
- addressing non-code assets as part of core asset sustainment

Product development risk is very dependent on good configuration management, the proper certification of assets, the careful management of product-unique components outside the core asset base, and an acquisition strategy that is compatible with the business case and total cost of ownership.

3.2.5 Question 5

"If you were king," what would you do to promote product line practice in the DoD?

Discussion

The group session was brought to a close by having each individual identify the one thing he or she would do to promote product line practice if he or she were in control. The members' responses included

- Ensure a sufficient and stable budget for those willing to adopt a product line approach.
- Have the government procure software-intensive subsystems and products as a product line, rather than as an extension of an aircraft (platform) buy, if the subsystem/product is suitable for deployment on multiple aircraft (platforms).
- Educate PMs and make sure they have sufficient experience before taking on major acquisitions.
- Sponsor the development of a set of DoD best acquisition practices that are mandated with variants for a product line.
- Get senior DoD leadership and decision makers to understand the importance of software technology, such as product lines, and make an investment to promote such practices.

4 Summary

The SEI's Sixth DoD Product Line Practice Workshop explored the product line practices of organizations in the DoD community in light of best commercial practices and government experience in software product lines. The presentations and discussions again validated the pivotal pieces of the framework and provided valuable feedback on the companion. Challenges and solutions within the DoD community were discussed.

The working groups addressed questions important to DoD product line practice. As in previous workshops, the empirical and anecdotal evidence that the workshop participants brought to the discussion significantly enhanced our current understanding of the practices and issues as they apply to the DoD. Traditional DoD acquisition strategies are not naturally conducive to software product lines. However, product line practice is possible within the DoD, and more and more DoD organizations are taking a product line approach.

Within the DoD, there needs to be increased awareness about DoD product line activities that might be relevant. It is critical for the DoD to think more strategically and to share information and outcomes among its different areas. These outcomes could help to prevent duplication and redundant development.

In an effort to expand both the information base and the DoD community interested in software product lines, the SEI was encouraged by the participants to continue to hold similar workshops.

The results of this workshop have been incorporated into the companion, which will continue to be refined and revised as the technology matures and as we continue to receive feedback and to work with the growing community of software engineers championing a product line approach. If you have any comments on this report or are using a product line approach in the development or acquisition of software-intensive systems for the DoD and would like to participate in a future workshop, please send email to Linda Northrop at lmn@sei.cmu.edu.

References

URLs are valid as of the publication date of this document.

[Argon 04] Argon Engineering. <http://www.argoneng.com/intro.html> (2004).

[Bass 97] Bass, L.; Clements, P.; Cohen, S.; Northrop, L.; & Withey, J. *Product Line Practice Workshop Report* (CMU/SEI-97-TR-003, ADA327610). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1997. <http://www.sei.cmu.edu /publications/documents/97.reports/97tr003/97tr003abstract.html>.

[Bass 98] Bass, L.; Chastek, G.; Clements, P.; Northrop, L.; Smith, D.; & Withey, J. *Second Product Line Practice Workshop Report* (CMU/SEI-98-TR-015, ADA354691). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1998. <http://www.sei.cmu.edu/publications/documents/98.reports /98tr015/98tr015abstract.html>.

[Bass 99] Bass, L.; Campbell, G.; Clements, P.; Northrop, L.; & Smith, D. *Third Product Line Practice Workshop Report* (CMU/SEI-99-TR-003, ADA361391). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1999. <http://www.sei.cmu.edu /publications/documents/99.reports/99tr003/99tr003abstract.html>.

[Bass 00] Bass, L.; Clements, P.; Donohoe, P.; McGregor, J.; & Northrop, L. *Fourth Product Line Practice Workshop Report* (CMU/SEI-2000-TR-002, ADA375843). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000. <http://www.sei.cmu.edu/publications/documents/00.reports /00tr002.html>.

[Bass 03] Bass, L.; Clements, P.; & Kazman, R. *Software Architecture in Practice*, 2nd ed. Reading, MA: Addison-Wesley, 2003.

[Bergey 98] Bergey, J.; Clements, P.; Cohen, S.; Donohoe, P.; Jones, L.; Krut, R.; Northrop, L.; Tilley, S.; Smith, D.; & Withey, J. *DoD Product Line Practice Workshop Report* (CMU/SEI-98-TR-007, ADA346252). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1998. <http://www.sei.cmu.edu /publications/documents/98.reports/98tr007/98tr007abstract.html>.

[Bergey 99a] Bergey, J.; Campbell, G.; Clements, P.; Cohen, S.; Jones, L.; Krut, R.; Northrop, L.; & Smith, D. *Second DoD Product Line Practice Workshop Report* (CMU/SEI-99-TR-015, ADA375845). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1999. <http://www.sei.cmu.edu/publications/documents/99.reports /99tr015/99tr015abstract.html>.

[Bergey 99b] Bergey, J.; Fisher, M.; Jones, L.; & Kazman, R. *Software Architecture Evaluation with ATAM in the DoD System Acquisition Context* (CMU/SEI-99-TN-012, ADA377450). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1999. <http://www.sei.cmu.edu/publications/documents/99.reports /99tn012/99tn012abstract.html>.

[Bergey 00a] Bergey, J.; Campbell, G.; Cohen, S.; Fisher, M.; Jones, L.; Krut, R.; Northrop, L.; O'Brien, W.; Smith, D.; & Soule, A. *Third DoD Product Line Practice Workshop Report* (CMU/SEI-2000-TR-024, ADA387259). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000. <http://www.sei.cmu.edu /publications/documents/00.reports/00tr024.html>.

[Bergey 00b] Bergey, J.; Fisher, M.; Gallagher, B.; Jones, L.; & Northrop, L. *Basic Concepts of Product Line Practice for the DoD* (CMU/SEI-2000-TN-001, ADA375859). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000. <http://www.sei.cmu.edu/publications/documents/00.reports /00tn001.html>.

[Bergey 01] Bergey, J. et al. *Fourth DoD Product Line Practice Workshop Report* (CMU/SEI-2001-TR-017, ADA399205). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2001. <http://www.sei.cmu.edu/publications/documents/01.reports /01tr017.html>.

[Bergey 03a] Bergey, J. et al. *Fifth DoD Product Line Practice Workshop Report* (CMU/SEI-2003-TR-007). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003. <http://www.sei.cmu.edu/publications/documents/03.reports/03tr007.html>.

[Bergey 03b] Bergey, J.; Campbell, G.; Cohen, S.; Fisher, M.; Gallagher, B.; Jones, L.; Northrop, L.; & Soule, A. *Software Product Line Acquisition: A Companion to A Framework for Software Product Line Practice, Version 2.0.* <http://www.sei.cmu.edu/plp/companion.html> (2003).

[Brownsword 96] Brownsword, L. & Clements, P. *A Case Study in Successful Product Line Development* (CMU/SEI-96-TR-016, ADA315802). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1996. <http://www.sei.cmu.edu/publications/documents/96.reports/96.tr.016.html>.

[Chastek 02] Chastek, G., ed. *Proceedings of the Second Software Product Lines Conference (SPLC2)*. San Diego, CA, August 19-22, 2002. New York, NY: Springer-Verlag, 2002.

[Clements 01] Clements, P. et al. *Fifth Product Line Practice Workshop Report* (CMU/SEI-2001-TR-027, ADA272355). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2001. <http://www.sei.cmu.edu/publications/documents/01.reports/01tr027.html>.

[Clements 02a] Clements, P. & Northrop, L. *Software Product Lines: Practices and Patterns.* Reading, MA: Addison-Wesley, 2002.

[Clements 02b] Clements, P. & Northrop, L. *A Framework for Software Product Line Practice, Version 4.1*. <http://www.sei.cmu.edu/plp/framework.html> (2002).

[Cohen 02] Cohen, S.; Dunn, E.; & Soule, A. *Successful Product Line Development and Sustainment: A DoD Case Study* (CMU/SEI-2002-TN-018, ADA407788). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002. <http://www.sei.cmu.edu/publications/documents/02.reports/02tn018.html>.

[Donohoe 00] Donohoe, P., ed. *Software Product Lines: Experience and Research Directions*, Proceedings of the First Software Product Line Conference (SPLC1). Denver, CO, Aug. 28-31, 2000. Boston, MA: Kluwer Academic Publishers, 2000.

[Park 96] Park, Robert E.; Goethert, Wolfhart B.; & Florac, William A. *Goal-Driven Software Measurement—A Guidebook* (CMU/SEI-96-HB-002, ADA313946). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1996. <http://www.sei.cmu.edu/publications/documents/96.reports/96.hb.002.html>.

[SPC 93] Software Productivity Consortium. *Reuse-Driven Software Process Guidebook, Version 02.00.03* (SPC-92019-CMC). Herndon, VA: Software Productivity Consortium Services Corporation, 1993.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY
(Leave Blank)

2. REPORT DATE
March 2004

3. REPORT TYPE AND DATES COVERED
Final

4. TITLE AND SUBTITLE
Software Product Lines: Experiences from the Sixth DoD Software Product Line Workshop

5. FUNDING NUMBERS
F19628-00-C-0003

6. AUTHOR(S)
John Bergey, Sholom Cohen, Lawrence Jones, Dennis Smith

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

8. PERFORMING ORGANIZATION REPORT NUMBER
CMU/SEI-2004-TN-011

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
HQ ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2116

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified/Unlimited, DTIC, NTIS

12B DISTRIBUTION CODE

13. ABSTRACT (MAXIMUM 200 WORDS)

The Carnegie Mellon® Software Engineering Institute held the Sixth Department of Defense (DoD) Product Line Practice Workshop in September 2003. The workshop was a hands-on meeting to share DoD product line practices, experiences, and issues and to discuss ways in which specific product line practices are accomplished within the DoD. Participants reported encouraging progress on DoD software product lines. Additionally, participants addressed some important implementation questions. This report synthesizes the workshop presentations and discussions.

14. SUBJECT TERMS
DoD product line practice, Product Line Practice Framework, product line workshop, software architecture, software product lines, DoD software acquisition, Acquisition Companion to A Framework for Software Product Line Practice, acquisition companion

15. NUMBER OF PAGES
32

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
UL