DoD Product Line Practice Workshop Report

John Bergey
Paul Clements
Sholom Cohen
Pat Donohoe
Larry Jones
Bob Krut
Linda Northrop
Scott Tilley
Dennis Smith
Jim Withey

May 1998

TECHNICAL REPORT
CMU/SEI-98-TR-007
ESC-TR-98-007

CMU/SEI-98-TR-007
ESC-TR-98-007

DoD Product Line Practice Workshop Report

John Bergey
Paul Clements
Sholom Cohen
Pat Donohoe
Larry Jones
Bob Krut
Linda Northrop
Scott Tilley
Dennis Smith
Jim Withey

May 1998

Product Line Systems Program

Carnegie Mellon
Software Engineering Institute
Pittsburgh, PA
15213-3890

This report was prepared for the

SEI Joint Program Office
HQ ESC/DIB
5 Eglin Street
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER

(signature on file)

Mario Moya, Maj, USAF
SEI Joint Program Office

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

This document is available through Asset Source for Software Engineering Technology (ASSET): 1350 Earl L. Core Road; PO Box 3305; Morgantown, West Virginia 26505 / Phone: (304) 284-9000 or toll-free in the U.S. 1-800-547-8306 / FAX: (304) 284-9001 World Wide Web: http://www.asset.com / e-mail: sei@asset.com

Copies of this document are available through the National Technical Information Service (NTIS). For information on ordering, please contact NTIS directly: National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161. Phone: (703) 487-4600.

This document is also available through the Defense Technical Information Center (DTIC). DTIC provides access to and transfer of scientific and technical information for DoD personnel, DoD contractors and potential contractors, and other U.S. Government agency personnel and their contractors. To obtain a copy, please contact DTIC directly: Defense Technical Information Center / Attn: BRR / 8725 John J. Kingman Road / Suite 0944 / Ft. Belvoir, VA 22060-6218 / Phone: (703) 767-8274 or toll-free in the U.S.: 1-800 225-3842.

Table of Contents

List of Figures

List of Tables

Abstract

The Department of Defense (DoD) Product Line Practice workshop, *Product Lines: Bridging the Gap – Commercial Success to DoD Practice* was a hands-on meeting held in March 1998. Its purpose was to identify industry-wide best practices in software product lines, to share DoD product line experience, to explore the technical and non-technical issues involved, and to discuss ways in which the current gap between commercial best practice and DoD practice can be bridged. This report synthesizes the workshop presentations and discussions that described selected product line practices and identified barriers and enablers to achieving these practices within the DoD.

1. Introduction

1.1 Why Product Line Practice?

Historically, software engineers have designed software systems for functionality and performance. A single-system mentality prevailed. Little attention was paid to the consequences of a design in the production of multiple software-intensive products or their long-term sustainment. Large software development, acquisition, and reengineering efforts undertaken with this single-system mentality perpetuate a pattern of large investment, long product cycles, system integration problems, and a lack of predictable quality. Each product involves vast investments in requirements analysis, architecture and design, documentation, prototyping, process and method definition, tools, training, implementation, and testing with little carried forward to future products.

Many organizations have realized that they can no longer afford to develop or to acquire multiple software products one product at a time. They have instead adopted a product line approach that uses software assets to modify, assemble, instantiate, or generate multiple products referred to as a product line.

A product line is defined to be a group of products sharing a common, managed set of features that satisfy specific needs of a selected market or mission. A software architecture that capitalizes on commonalities in the implementation of the line of products provides the structural robustness, which makes the derivation of individual software products from software assets economically viable. A software architecture of a computing system is the structure or structures of the system that consist of software components, the externally visible properties of those components, and the relationships among them [Bass 97]. A software asset is a description of a partial solution (such as a component or design document) or knowledge (such as requirements database or test procedures) that engineers use to build or modify software products [Withey 96].

Some organizations have already experienced considerable savings by using a product line approach for software system production. Other organizations are attracted to the idea but are in varying stages of operationalizing product line practices.

In January 1997, the Software Engineering Institute (SEI) launched a technical initiative, 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 the multi-use of software assets so that these organizations can produce and maintain similar systems of predictable quality and at a lower cost. One of the strategies for reaching this goal involves direct interaction with and nurturing of the community interested in product line practice.

This transition strategy has been executed, in part, by a series of product line workshops organized by the SEI. Two of these workshops, in December 1996 and November 1997, brought together international groups of leading practitioners from industry to codify industry-wide best practices in product lines. The results of these workshops are documented in an SEI report entitled *Product Line Practice Workshop Report* [Bass 97].[1] The SEI has also refined the results of these previous workshops through work with collaboration partners, participation in other workshops, and continued research. In addition, the SEI is producing a framework for product line practice. The framework identifies the essential elements and practices that an organization should master for successful deployment of a product line. The framework categorizes product line practices according to software engineering, technical management, and enterprise management. These categories do not represent job titles, but rather disciplines. The framework is a living document that will grow and evolve.

1.2 About the Workshop

To share the industrial experience with the DoD product line practice community and to learn the factors and issues in current government approaches that both enable and inhibit software product lines, the SEI held a two day Product Line Practice Workshop, *Product Lines: Bridging the Gap - Commercial Success to DoD Practice*, in March 1998. All participants in this workshop were from the DoD acquisition and contractor community. They were invited based upon our knowledge of their experience with and commitment to software product lines and strategic software reuse as either DoD system acquirers or DoD system contractors. Together we elucidated and discussed the issues that form the backbone of this report.

The workshop participants included

- John Bergey, Product Line Systems Program, Software Engineering Institute
- Loring Berhnardt, Mitre/Integrated Tactical Warning Aid And Attack (ITWAA)
- Patrick Bidon, Joint National Test Facility
- David Bristow, ITT SSC/Integrated Tactical Warning Aid And Attack (ITT SSC/ITWAA)
- Brian Bulat, Joint National Test Facility/Lockheed Martin
- Paul Clements, Product Line Systems Program, Software Engineering Institute
- Sholom Cohen, Product Line Systems Program, Software Engineering Institute
- Peter Crump, TYBRIN Corporation
- Mark Dehlin, West Virginia High Technology Consortium (WVHTC) Foundation
- Pat Donohoe, Product Line Systems Program, Software Engineering Institute

[1] The technical report documenting the November 1997 workshop is currently in the external review process.

- LTC Eugene Glasser, U. S. Army Information Systems Software Center (USAISSC)
- Robert Harrision, Naval Surface Warfare Center (NSWC)
- Randall Heiling, United States Air Force (USAF)
- James E. Hooper, Sakonnet Technology Group
- Larry Jones, Government Sector, Software Engineering Institute
- Judy Kerner, Aerospace Corporation
- Bob Krut, Product Line Systems Program, Software Engineering Institute
- Bob Linza, Joint National Test Facility
- Reed Little, Product Line Systems Program, Software Engineering Institute
- Mike Lombardi, U. S. Army Communications Electronics Command (CECOM)
- Capt. John Marsh, Joint National Test Facility
- Chris Martin, Joint National Test Facility
- George Newberry, United States Air Force (USAF)
- Linda Northrop, Manager, Product Line Systems Program, Software Engineering Institute
- John Ohlinger, National Reconnaissance Office
- George Rumford, Office of the Secretary of Defense
- Robert Sanders, Joint National Test Facility
- Dennis Smith, Product Line Systems Program, Software Engineering Institute
- Scott Tilley, Product Line Systems Program, Software Engineering Institute
- Will Tracz, Lockheed Martin
- Joseph Vonusa, Naval Surface Warfare Center (NSWC)
- Roger Williams, Boeing
- James Withey, Product Line Systems Program, Software Engineering Institute

The workshop presentations and discussions focused on the structure of the SEI Product line practice framework, which identifies essential practices in the areas of software engineering, technical management, and enterprise management. To properly set the context, the workshop began with five presentations. The first three presentations were given by SEI technical leaders of the product line work. They characterized the current state of product line practice by describing the industry's best product line practices, the current contents of the framework, and product line acquisition issues prevalent in the DoD. The remaining two presentations described individual DoD product line experiences, each at rather different ends of the spectrum. These presentations were included to turn the focus toward the DoD and provide a taste of DoD product line approaches. Though there certainly are other examples of DoD product line experiences that have been described at other forums, the emphasis in this workshop was on interactive participation. Presentations were purposely limited to permit ample time for discussion and exploration of the relevant issues.

Following the presentations, the participants divided into four working groups compatible with the framework structure to further explore selected product line practices, barriers, and enablers within the DoD in the areas of software engineering, technical management, enterprise management for DoD acquisition organizations, and enterprise management for DoD contractor organizations. There were two working groups discussing DoD enterprise management practices because we wanted to explore these practices from the perspective of the contractor and the acquisition organization.

Each group was asked to select from among the practices identified in the framework for their area and to describe the following:

- the practice
- the delta for this practice for product lines versus single product development
- the barriers for this practice in working with or within the DoD
- the mitigation strategies to overcome the identified barriers

Each group was also asked to capture important general issues outside the focus of the working group.

The working groups then presented their results to the entire group. One of the participants, Will Tracz, provided a spontaneous workshop summary.

1.3 About This Report

This document summarizes the presentations and discussions at the workshop. As such, the report is written primarily for those in the DoD who are already familiar with product line concepts, most especially those who are already working or initiating product line practices in their own organizations. Acquisition managers and technical software managers should also benefit from the information in this report.

The report is organized into six main sections that parallel the workshop format:

1. Introduction
2. State of Product Line Practice: Digest of SEI Overview Presentations
3. DoD Product Line Experiences: Digest of DoD Presentations
4. Product Line Practices: Working Group Reports
5. Summary
6. Conclusion

The section following this introduction, *State of Product Line Practice: Digest of SEI Overview Presentations*, summarizes the three SEI presentations that set the context for the workshop. The next section, *DoD Product Line Experiences: Digest of DoD Presentations*, summarizes the product line experience of two of the workshop participants. Section 4 is composed of the four working group reports on selected practices, DoD barriers and enablers in software engineering, technical management, enterprise management in acquisition organizations, and enterprise management in contractor organizations, respectively. Each of the working group reports reflects the interests, experiences, and style of the individual group. The emphasis and completeness of the information varies by group and by practice. The practices discussed are important in their very selection. The summary in Section 5 recaps the major themes, and the conclusion in Section 6 provides a brief analysis and suggests future directions. Additionally, there is an Appendix providing a glossary of terms.

2. State of Product Line Practice: Digest of SEI Overview Presentations

2.1 Introduction

Three SEI technical leaders in the product line work gave presentations aimed at setting the context for the workshop by sharing both commercial product line practice and issues and the results of the SEI product line efforts to date, and by highlighting some of the perceived product line acquisition barriers within the DoD. Linda Northrop led the session with a talk that developed the primary themes for the workshop. She discussed the motivation for product lines, what a product line is, leverage offered by product lines, the state of commercial product line practice, relevance of product lines to DoD, the SEI Product Line Practice Initiative, the SEI Product line practice framework, skills and roles in product line practice, and the risks and challenges of software product lines.

Paul Clements then distilled the results of the SEI's Second Product Line Practice Workshop held in November 1997. He uncovered the issues and solutions shared by experts from seven commercial organizations with real-world experience in developing and fielding software product lines. Finally, James Withey provided an overview of the motivations for and the benefits accruing from product line practice in system acquisitions. His talk underscored several issues facing the DoD and discussed the pros and cons of two approaches for introducing product line practice into the current DoD organizational structure.

2.2 Essentials of Successful Product Line Practice, Linda M. Northrop - SEI

2.2.1 Motivation

Over the past 30 years, software engineering has emerged as the critical technology of the twentieth century. Every organization is in the software business, whether the product is phone service, engines, consumer goods, satellites, express package delivery, automobiles, weapons, elevators, or government services. Organizations that built their reputations on hard goods and electronics now find that their bottom line is controlled by their software quality and productivity. These organizations have business goals that include high quality, quick time to market, effective use of limited resources, product alignment, low cost production, and low cost maintenance. To achieve these goals these organizations have strategies for improved efficiency and productivity.

Unfortunately, the development of software is expensive and has often been unpredictable and unreliable. There are some fundamental reasons for this situation. Software engineering is relatively young and does not benefit from the legacy of discipline and codified standards and practices found in other engineering disciplines. Consequently, we have often developed software in a chaotic fashion, depending on the idiosyncratic technical skills of analysts and programmers, resulting in software that often overruns its schedule and budget, and that is difficult to evolve and maintain. At the same time, the systems we are building are vastly more complex. Instead of attacking this complexity by leveraging previous efforts, we have re-invented the same wheel hundreds of times. For example, many government organizations have developed their own payroll systems, inventory systems, and budgeting systems that essentially duplicate systems of other government agencies. There have been too few systematic efforts to leverage software investment across similar systems.

Given the gravity and pervasiveness of the problems with software, three classes of effort have emerged to enable the development of manageable, less expensive, and higher quality software. Technology innovations have been operationalized to great advantage. Improvements such as vastly increased memory, greater processing speed, distribution of computing resources, and more efficient languages, databases, and tools have solved many low-level problems and thereby have permitted greater attention to higher level computing issues. Ironically, the technology innovations have also paved the way for more complex applications that have exacerbated the overall software problem. The widespread movement toward software process improvement has yielded productivity and quality gains. The third class of efforts has focused on reuse.

Reuse has been promised to offer great potential. Reuse efforts with a focus on increasingly larger grain pieces—modules of the 70s, to objects of the 80s, to components in the 90s—have provided some opportunity for horizontal leverage, but have not produced the expected benefits. Despite early disappointing results, it is this last area, reuse, that appears ripe for exploitation. By refocusing the reuse target on strategic, large-grained reuse at the level of a product line, reuse can result in remarkable efficiency and productivity improvements and time economies. In combination with the known benefits of process improvement and technology innovation, systematic reuse through product lines offers great promise to software development and acquisition organizations.

2.2.2 What Is a Product Line?

A product line is a group of products sharing a common, managed set of features that satisfy specific needs of a selected market or mission. For example, a telecommunications company may offer a number of cellular phones that share a similar market strategy and an application domain, thus making up a product line. A domain is an area of knowledge or activity characterized by a set of concepts and terminology understood by practitioners in the area. The products in a software product line can best be leveraged when they share a common architecture that is used to structure components from which the products are built.

The architecture and components are central to the set of core assets[2] used to construct and evolve the products in the product line. In other words, a software product line can best be leveraged by managing it as a product family, which is a set of related systems built from a common set of assets. For example, if the product line of cellular phones is built from a common architecture and set of common components, it is managed as a product family. When we refer to a product line, we always mean a software product line built as a product family. This particular use of terminology is not nearly as important to us as the underlying concepts involved, namely, the use of a common asset base in the production of a set of related products.

Product line practice is therefore the systematic use of software assets to modify, assemble, instantiate, or generate the multiple products that constitute a product line. Product line practice involves strategic, large-grained reuse as a business enabler.

2.2.3 Leverage Offered by Product Lines

Developing, acquiring, and maintaining multiple software products one product at a time is no longer economically viable if a multi-project business case exists. Fred Brooks in his seminal article, *No Silver Bullet*, says that the most difficult part of building software is not the coding, but the decisions you make[Brooks 87]. It is these decisions, as captured in core assets, that are used multiple times in a product line approach. Reuse that occurs earlier in the life cycle than code accrues much more benefit the than the earlier idea of code reuse. Product lines amortize the investment in these and other core assets (such as requirements and requirements analysis, domain modeling, software architecture and design, performance engineering, documentation, test plans, test cases, and test data), people (their knowledge and skills), processes, methods, tools, budgets, schedules, work plans, and components.

A number of organizations have already gained order-of-magnitude improvements in efficiency, productivity, and quality through the strategic software reuse afforded by a product line approach. However, even more important than significant cost savings, product line practice enables an organization to get its products to market or field at the right time. Time has emerged as a critical success factor in a number of highly competitive product lines, such as cellular phones, pagers, and printers. If a product reaches the marketplace several months after its competitor, it may have lost its window of opportunity and has become a failure regardless of its features or cost.

The Swedish naval defense contractor, CelsiusTech, turned to a product line approach in the development of their on-board ship command and control systems in the mid 1980's [Brownsord 96]. Their efforts resulted in a product line they call Ship System 2000 that now spans 12 classes of ships, from surface vessels to submarines, and has fielded more than 50 ship systems from the same architecture and set of components. Among many other benefits

[2] Some organizations refer to the core asset base that is reused on systems in a product line as a *platform*.

that CelsiusTech has enjoyed with this product line is a reversal in the hardware-to-software cost ratio, 35:65 to 60:20, that now favors the software.

A number of other companies have shown similar success using a product line approach. Hewlett Packard, who like CelsiusTech has been using a product line approach for the past ten years, has collected substantial metrics showing two to seven times cycle time improvements with product lines. On one project they were able to ship five times the number of products, that were four times as complex, had three times the number of features, and with four times the number of products shipped per person.

Motorola used a product line approach for FLEXworks, a family of one-way pagers. They have shown a four times cycle time improvement with 80% reuse. Among other commercial domains that have shown equally dramatic results are air traffic control (Raytheon), commercial bank systems (Alltel), engines (Cummins), telecommunication systems (Ericsson, Nokis, Lucent, AT&T), college registration systems (Buzzeo). These organizations have not moved to product lines to break into the market. They have needed product line practice not only to improve time to market, but to continue their health in the market, to maintain market presence, to sustain unprecedented growth (especially poignant given today's employment market) to compensate for an inability to hire.

Product line practice is both a technical and a business decision. To move to product lines, an organization must alter its technical practices, its management practices, its organizational structure and personnel, and its business approach. Most importantly, it needs to move to an architecture-centric approach where the architecture is the foundation for the product line.

The architecture represents the key technical building block. A software architecture describes the structural properties of the software, typically the components and their relationships and guidelines about their use. It is the root of system qualities and ensures that variability across products can be accomplished by changes confined to one or a select set of components. The architecture of a system makes or breaks its ability to be secure, reliable, and meet its performance requirements. An architecture either explicitly or implicitly makes tradeoffs among each of these qualities. Once the basic structures of a system have been developed, tunings to the code will make only marginal differences.

A software architecture actually has multiple structures or views, each of which focuses on a particular set of issues important to one or more classes of stakeholders in the system.[3] A software architecture may have a number of constraints placed on it by an existing set of standards or technical architecture, such as the DoD's Joint Technical Architecture.

---

[3] The most recent book in the SEI Addison Wesley Series, *Software Architecture in Practice*, provides a thorough treatment of software architecture [Bass 98].

2.2.4 The Relevance of Product Lines to DoD

There is a growing recognition within the DoD that new acquisition approaches leveraging best commercial practices need to be implemented. At the top DoD policy levels, acquisition reform from DoD Directive 5000.1 and DoD Regulation 5000.2-R have focused on using these best practices to reduce cost, schedule, and technical risks, and to advance architecture-based approaches to reuse that support open systems, interoperability, and COTS. Statements by present and former top-level DoD officials all express a need for the DoD to leverage the best commercial practices that have turned around American commercial industry over the last decade. It is important for the DoD to use innovative, commercially proven practices to reduce cycle time, improve quality, reduce cost, improve efficiency, and reduce technical risks. At an operational level, it is not exactly clear how this will happen. Support is needed to understand what the commercially proven practices are that cut cycle time and cost while improving quality and efficiency; what the viable architecture-based approaches to reuse are; and how systematic software reuse is adopted in a DoD organization.

There have been several reuse efforts within the DoD, and there are certainly examples where the systematic reuse and horizontal leverage characteristic of a product line approach have occurred and are occurring. Two such examples were described by the featured government speakers at this workshop (see Section 3). Moreover, there are many others within the DoD that are attracted to product line concepts. Yet we are not at the point where product lines are a truly viable, repeatable practice within the DoD: there is a gap between best commercial practice and routine DoD practice. Part of this gap is related to the standard acquisition approach of acquiring a single stove-pipe system at a time, and part is attributable to the fact that the commercially successful practices have remained proprietary. This workshop is a part of the planned activities of the SEI's Product Line Practice Initiative, which is attempting to bridge the gap, or at least to fill it.

2.2.5 The SEI Product Line Practice Initiative

The vision of the SEI Product Line Practice Initiative is that product line development will one day become a low-risk, high-return proposition and that techniques for finding and exploiting system commonalities and for controlling variability will be standard software engineering practice in the DoD, government, and industry. Our strategies to achieve these goals are to

1. develop an integrated business and technical approach to product line practice by selecting, refining, and codifying practices of demonstrated effectiveness for creating and acquiring software product lines in different domains and organizational contexts
2. build and nurture a community interested in and informed about product line practice in order to transition product line practices and enable their use in the DoD

To implement these strategies, the work of the initiative is focused on three individual technical maturation areas: architecture-based development, business and acquisition strategies that facilitate product line practice, and reengineering strategies for mining core assets. The initiative is also focused on two technical transition areas: product line integration and community outreach. The initiative leverages external experiences and initiatives as well as the output of other SEI technical initiatives. In addition, there is ongoing collaboration with selected DoD, government (non-DoD), and commercial organizations to mature and codify viable product line practices.

The central function of the product line integration work is to distill and document initiative results and knowledge in the SEI Product line practice framework,[4] to develop generic product line artifacts (such as a Product line concept of operations that organizations can tailor for their own purposes), and to document case studies of product line experiences. Widespread transition is accomplished via technical reports, publications, presentations, courses, check lists, quick guides, workshops, and the framework. All of our latest work is accessible via our Web site.

2.2.6 Product Line Practice Framework

We are capturing the essential elements of product line practice in an evolving framework. Organizations that have succeeded with product lines vary widely in the nature of their products, their market or mission, their organizational structure, their culture and policies, their software process maturity, and the extensiveness of their domain expertise and legacy artifacts. Nonetheless, there are universal essential elements and practices that emerge. The framework focuses on these universals while accommodating various organizational contexts and starting points.

---

[4] The framework, currently in draft form, is intended to be an evolving document. The first version is targeted to be on the Web in 1998.

As depicted in Figure 1, core asset development and acquisition are distinguished from product development and acquisition using these assets with the understanding that management orchestrates, tracks, and coordinates both sets of activities. The arrows signify the high degree of iteration involved.

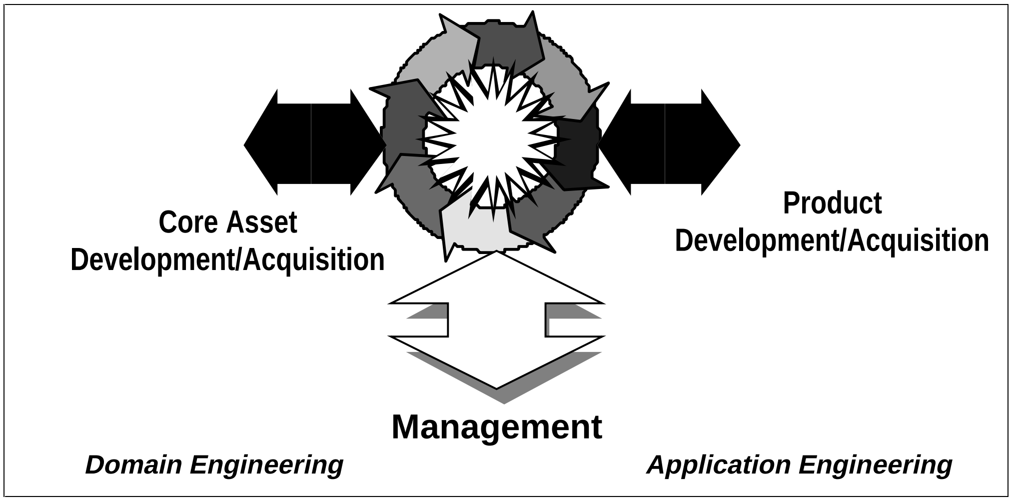

Figure 1: Product Line Practice Framework Organization

On the left side of the figure, the critical core assets involved are the architecture and components. Inputs to the development and acquisition of core assets are product constraints found by analyzing the similarities and differences of current and projected products, production constraints such as might be found in a technical architecture, a production strategy for the assets, and an inventory of pre-existing assets, styles, patterns, and architectural frameworks. The outputs are the core assets, a preliminary list of the products they will support, and a production plan for how the core assets will be used in the development or acquisition of products.

On the right side of the figure, individual products are developed or acquired from the core assets using the production plan that has been established. Product requirements are developed and refined with the existing core assets in mind, and products that systematically reuse the core assets are output.

There is a strong feedback loop between the core assets and products. Core assets are refreshed as new products are developed. In addition, the value of the core assets is realized through the products that are developed from them. As a result, the core assets are made more generic by considering potential new products on the horizon. There is a constant need for strong and visionary management to invest the resources in the development of the core assets, and to develop the cultural change to view new products through the filter of the core assets.

There are essential practices in a number of specific areas that are required to produce the core assets and products in a product line and to manage the process at multiple levels. The framework describes the essential practice areas for software engineering, technical management, and enterprise management, where these categories represent disciplines rather than job titles. For individual practice areas, the framework highlights the delta for the product line approach versus an approach for single-system development.

2.2.6.1 Software Engineering

The software engineering practice areas include

- requirements management
- domain analysis
- architecture exploration, development, and evaluation
- mining existing assets
- component development
- testing
- effective utilization of COTS products
- performance/reliability/security engineering
- software system integration
- asset evolution

With the exception of domain analysis, all of these practices are also important for traditional product development. However, there are differences in the application of these practices for product lines. For example, in product line practice, requirements management is constrained by the existing asset inventory, and uses the core assets as a point of departure. Mining existing assets focuses on finding and adapting components for use in a wide variety of potential products.

2.2.6.2 Technical Management

The technical management practice areas include

- process modeling and implementation
- planning
- metrics, data collection, and tracking
- program acquisition management
- make, buy, outsource analysis and execution
- risk management
- configuration management
- cost asset analysis
- technology refreshment

These practice areas at first glance seem to represent good technical management in general. However, for product lines they have particular focus. For example, because of the wide variety of potential variations of individual products, configuration management, with strong automated support, is particularly important for developing and maintaining product lines. A strong metrics and data collection program is crucial both to understanding whether the product line practice is making an impact, and also to providing ROI (return on investment) justification for top management. The process for developing core assets and turning them into a variety of products is sharply different from that of developing a single software application. Such a process needs to be developed, modeled, and enforced to enable the product line to succeed.

2.2.6.3 Enterprise Management

Enterprise management is the name we give to the management of the business issues that are visible at the enterprise level, as opposed to those at the project level. Enterprise management includes those practices necessary to position the enterprise to take fullest advantage of the product line capability. The essential enterprise management practices include

- ensuring sound business goals
- providing an appropriate funding model
- performing market analysis
- developing and implementing a product line concept of operations
- achieving the right organizational structure
- assuring proactive management
- building and maintaining appropriate skill levels
- managing the organization’s customer and supplier interfaces

- ensuring inter-group collaboration and communication
- risk management
- technology management

Successful product lines represent a new way of doing business. This way requires vision and explicit support at the organizational level. There must be an explicit funding model to support the development of core assets. Communication between the management team, the marketing team, the core asset group, and the product team is crucial. It is important for the organizational structure to support the product line concept.

2.2.7 Skills and Roles in Product Line Practice

There is no single recommended organizational structure for product line development or acquisition. Successful structures vary with organizational context and culture. Regardless of the selected structure, each group within the product line team has a different set of skill requirements. These skills need to be identified, recruited for, trained, and rewarded. Deep domain expertise is an overall requirement that must exist in each group.

For managers, the requirements are vision, an understanding of the basic technical issues, and the ability to be decisive and inspire confidence. Managers need to communicate this vision and direction to the rest of the team. It is important for managers to be resilient and focused on the vision, in spite of the inevitable uncertainties that will occur during the establishment and implementation of this vision. They also need to recognize that while the ROI for a product line is impressive - it is not immediate, and requires a period of investment without immediate return.

The core assets group develops and maintains the architecture, the components, and the environment. This team requires an individual or small group responsible for the architecture. It is essential for the team to be able to abstract, to mediate, and to understand the domain area and its potential permutations. The core assets team communicates the asset capabilities to the marketing team, the management team, and the product development team. The focus of this group is strategic.

The production teams have a tactical focus. They must be able to adapt to customer problems, and to engineer a product from the core assets according to the production plan rather than from scratch. They must be able to customize a set of features to the core assets.

The marketing team must fully understand the potential of the product line and the variability that can be accommodated by individual products. They must be able to articulate emerging customer requirements to the product line technical groups and to relate the product potentials back to the customers. There is often more negotiating with customers, since customers can sometimes save considerable money if they are willing to modify their initial requirements to account for the core asset base.

2.2.8 Conclusion: The Challenge of Product Lines

The benefits of product lines are many, and many organizations have succeeded in accruing these benefits. The key themes among successful product lines are

- long and deep domain expertise
- a legacy base from which to build
- architectural excellence
- management commitment

Yet, there are also costs and risks on any product line program. Product lines represent a new way of doing business and require substantial up-front investment. The culture and organizational structure of the organization may need to be changed; and substantial training is required. There is a major risk if the scope of the product line is not properly determined. Too broad a scope renders the core assets too complex to be effectively reused; too narrow a scope does not justify the cost of core-asset development and maintenance. Customers need to be re-educated to understand how they can benefit from adopting a different mind set to acquisition. Since the product line depends on a strong architecture and strong components, there is a major risk if these assets are of poor or marginal quality. Rapidly changing technology or domain instability may make the core assets obsolete. In addition, if the management team is inconsistent or focuses on immediate rewards, a product line program is doomed.

There are product line challenges for the entire software community:

- developing a strong architecture
- evolving the architecture and core assets
- developing and implementing product line migration strategies for legacy systems
- collecting relevant data and tracking business goals
- funding models to support strategic reuse decisions
- developing and using acquisition strategies that support systematic reuse through product lines
- having and employing repeatable, integrated technical, management, and enterprise practices

Nonetheless, if properly managed, the benefits of a product line far exceed the costs. Strategic software reuse through a well-managed product line approach achieves enterprise goals for efficiency, time to market, productivity, and quality. It is our vision that product line practices will pervade software engineering in the new millennium.

2.3 Summary of the Second Product Line Practice Workshop, Paul C. Clements - SEI

2.3.1 Introduction

In November 1997, the SEI conducted the second in its series of workshops on product lines. The goal of this workshop was to discover basic issues and solutions in product lines and to validate the SEI Product line practice framework by assembling experts with real-world experience in developing and fielding software product lines.

Representatives from the following organizations were present:

- Ericsson: switching systems and data networks
- ALLTEL: financial institution software
- Lucent: switching systems
- Motorola: digital pagers
- Bosch: automotive electronics
- Raytheon: air traffic control
- Hewlett Packard: laser printers

The workshop featured presentations by the participants, followed by working groups that focused on product line practices in the areas of software engineering, technical management, and enterprise management. This summary first synthesizes the major themes of the presentations in the following categories: contextual factors, software engineering practices, technical management practices, enterprise management practices, and “hard issues.” Next, the discussions of each of the working groups are summarized.

2.3.2 Contextual Factors

Contextual factors describe the environment in which the organization exists or existed when it launched the product line effort. In terms of motivation, one of the common themes expressed was employing a product line strategy as an approach to achieving large-scale productivity gains and time-to-market improvements in response to unprecedented growth. The viability of a business often depends on responding to tight market windows. There was an underlying sentiment that product lines were not just a good idea; they were essential to the organization's continued health in the market. Interestingly, several of the organizations moved to product lines as a response not to dwindling business, but to unprecedented growth. Product lines enabled these organizations to achieve four to six times productivity improvement goals, and goals of 80% to 90% reuse of core assets.

All participants started with some legacy assets, although there was considerable variation in the type of baseline. Their length of experience with product lines varied from those who were just starting out to organizations that had over 15 years of experience developing products using a number of product line characteristics. One constant factor cited by all participants is the need for rich domain experience.

2.3.3 Software Engineering Practices

Although all efforts developed a set of core assets, there was not usually an explicit first step of domain analysis because management is often unwilling to allow the up-front time that this requires. However, several organizations described abbreviated forms of domain analysis, such as commonality analysis used by Lucent. These methods focus on determining the scope of the product line by articulating key requirements and features, and analyzing common features of the core assets as well as variations in current and potential products.

The architecture forms the conceptual foundation of every product line identified by participants. A layered architectural style was used most often. Although there was often not an explicit first step in the creation of an architecture, everybody had one, and its creation was not cited as problematic, although discussion of architecture was somewhat limited due to its key to success and the proprietary nature of anything so critical. In addition to the architecture, large, pre-integrated chunks that can be used as components needed to be mined and developed. (A component is configurable, packageable, and distributable in a stand-alone fashion.)

Participants noted that because organizations do not really do “green-field” development of product lines, evolving a product line from existing software is the rule, rather than the exception. In fact, because of the importance of deep domain experience, “green-field” efforts suffered from a lack of initial feasibility proof. There is not a general approach for reengineering or mining of assets. Many ad hoc techniques have been used, depending on the current asset base.

2.3.4 Technical Management Practices

In traditional software development, it is common to talk about the importance of metrics. However, few software organizations actually have a systematic metrics program. For product lines, a metrics program is more important because such an approach requires radical changes in the enterprise, and there is a temptation for management to give up, especially in the early stages before clear ROI data are available.

In contrast to standard practice, our product line participants collected metrics in a fairly systematic way. The metrics included

- time to market
- percentage of reused software
- lines of code per programmer

- increased number of products shipped
- product volume shipped
- number of new features released per year
- product volume shipped per person

In all cases, the metrics reported were impressive, with improvements often stated in a range of three to seven times that of traditional software development methods.

Configuration management is a critical enabling technology for product lines. It is important to be able to rebuild any version of a product quickly. All organizations required sophisticated configuration management tool support. They noted the need to customize tools for organizational needs, and to carefully develop naming conventions and scripts to manage organization-specific components and architectural families.

2.3.5 Enterprise Management Practices

Achieving the right organizational structure is a critical success factor for the development of a viable product line. Although it is important to have separate groups for core assets and for product development, there are variations in how this is actually accomplished. At an earlier workshop, the predominant model consisted of separate organizational groups responsible for core assets and for product development. This structure prevents the pressures of product development from taking precedence over the need to continually evolve the core assets. At the November workshop, several participants described a model in which both functions are housed in the same group, but in which distinct roles for core assets and product development are defined. This approach combats a tendency for the core asset group to "produce beauty, not profit." Housing both functions in the same group seems to work best in smaller organizations.

A funding model for core asset development also needs to be developed because the core assets do not directly generate revenue. Some organizations place a tax on products, while others get the funding out of the research and development budget.

An effective product line approach enables a range of new business opportunities. Some organizations develop a separate unit or a technical steering group to oversee product line evolution. A recurring theme has been that the managed set of core assets provides leverage for unanticipated market opportunities and evolution to new types of products. For example, a command and control product line can provide the foundation for air traffic control, and air traffic control can evolve to marine vessel control.

2.3.6 "Hard Issues"

The development of a product line approach creates its own set of problems that do not have easy answers. Software engineering issues include answering the question of when to re-architect the system, when to re-engineer components, and how products should suggest changes to the core assets. In addition, there is no clear resolution to the problem of achieving reliability in the face of a test-case explosion for complex systems.

In the area of technical management, we have already noted that everybody collects metrics. However, there is no clear consensus on which metrics to collect or why to collect them. Configuration management presents its own problems of traceability through a complex set of derived products.

In the enterprise management area, a range of issues have emerged. No clear resolution of how to select an appropriate funding model has been developed, even though a funding model is critical to the viability of a product line. Organizations recognize that to maintain management support, visible results need to be demonstrated within a six-month time period. Otherwise, it is difficult to maintain constancy of management purpose and organizational direction. A set of issues related to managing long-term customer support for a constellation of products, on how to define and bid warranties, and long-term ownership must be addressed by organizations that develop product lines. In addition, new approaches for maintaining customer confidence in the maturity of the product line need to be explored, especially in safety-critical applications.

2.3.7 Working Group: Software Engineering

The Software Engineering working group focused on two issues: mining assets and domain analysis.

For mining core assets, four steps were identified:

1. deciding on commonalities among existing components
2. deciding that mining is the correct mechanism for achieving a new core asset
3. creating generic components
4. installing a generic component in the asset base for adoption by users of the core assets

For domain analysis, the group determined that there is a need to show a direct link to delivered systems to avoid disillusionment. Furthermore, the link from domain analysis to architecture development is not always clear or well understood. On the positive side, a useful byproduct of domain analysis is a body of knowledge that can be used as a training tool for the marketing group and developers.

This working group distinguished different types of domain analysis:

- Horizontal domain analysis is understanding the relationship among different features that provide different services.
- Vertical domain analysis is understanding the relationship among different layers that combine to form a usable collection of products.

2.3.8 Working Group: Technical Management

The technical management working group focused on three issues: metrics, testing, and configuration management.

Regarding metrics, the following types of metrics were identified:

- individual productivity in terms of lines of code per unit of time. (Even non-traditional approaches get justified in traditional terms.)
- productivity of the organization. An unresolved issue concerns how to weight organizational productivity by product complexity.
- time to market. Separate metrics can be developed for product time to market, core asset base time to market, and feature time to market. It is more important to deliver the product at the right time than to deliver it quickly. Too frequent releases may actually saturate the market.
- conformance to the reference architecture to measure the “success” of the product line within the organization.

Several hypotheses were developed about long-term impacts of product lines compared to single product development. These included the following:

- Overall quality should go up.
- The cost to fix any one defect is probably about the same.
- Cost per affected system should go down, leading to a potential “fix effectiveness” measure.
- Platform defect probability goes down as the platform is reused.

In the area of testing, experience suggests that the quality of the core assets improves over time. The testing effort for a product line is greater than the effort for a single system because fixes need to work for all products. However, the cost of the increased testing can be amortized over all of the products in the product line. Although testing per component increases, the total amount of testing for a specific product decreases because of the reuse of previously tested code. The working group formulated the hypothesis that there should be more product bugs than platform bugs as the product line matures. The implication is that the ratio of the number of core asset base bugs to product bugs found during system testing may be a metric to capture core asset base quality.

The technical management working group also discussed configuration management. Configuration management is different for product lines because of the complexity of the complete development history. Context switching occurs often, and core asset artifacts must be partitioned from product artifacts. Because of the importance of configuration management, the group made the assertion that an organization's practices in this area must be at Capability Maturity Model℠ (CMM®) level 2 by the time the first product in a product line is shipped.

The required configuration management services for a product line include version management and branching, labeling and control of labeling, storage and control of historical information to enable the easy re-creation of previous versions, mapping of component versions to product versions, and easy access to any version.

2.3.9 Working Group: Enterprise Management

This working group discussed two topics:

- architecture and organizational level issues
- product line production strategies

The responsibility for core assets varies by organization, and can be found at the corporate level, the business unit level, or the product line level. Two different sets of responsibilities can be distinguished in core asset development and management:

- Architecture development and evolution of the architecture is led by a senior architect.
- Asset development focuses on components, tools, and methods to help product development groups and releases periodic updates to the core assets.

Strategies for developing assets are based on a number of factors including competition, anticipated future demand, current capacity, technology maturity, and pricing strategies. Two types of production strategies were identified. The first strategy is a core asset production strategy in which there are standard core assets with little customization. In this case, product variability is controlled by individual product projects. The advantage of this strategy is that the core assets are simpler to manage and maintain. However, change is slow, and the strategy may not be flexible enough for the market. The second type of strategy is a customizable component strategy in which the variability of the product line is codified in the architecture and components. The scope of the product line is increased, and up-front costs

---

℠ Capability Maturity Model is a service mark of Carnegie Mellon University.
® CMM is registered in the U.S. Patent and Trademark Office.

are greater. Product development becomes component development and acquisition, and product building becomes a relatively simple integration task.

Two open issues were identified. The first issue concerned whether the core assets and products should be managed in two groups or one. The two-group approach may be most effective in green-field efforts because the product developers would not have the necessary deep product knowledge, and they may be able to leverage some of this knowledge from the core assets. The second issue concerned the different expectations of senior managers and product managers on the appropriate scope covered by the core assets. Senior managers want more products to be supported by the core assets, while product managers want more product features to be provided. While there is not an easy resolution to this built-in creative tension, the two perspectives are both important and need to be factored into strategic decisions on product line evolution.

2.4 Overview of Product Line Practice in DoD Acquisitions, James V. Withey - SEI

2.4.1 Introduction

Jim Withey of the SEI gave an overview of the motivations for and the benefits accruing from product line practice in system acquisitions. His talk highlighted several issues facing the DoD and discussed the pros and cons of two approaches for introducing product line practice into the current DoD organizational structure. Some ideas to stimulate the work of the working groups were also offered. The presentation concluded with a summary of SEI activities in the acquisition area.

2.4.2 Motivation and Benefits

Several factors provide the motivation for the DoD move to product lines. Chief among these factors is the high cost of software-intensive systems, especially in light of the department's shrinking budget. Another is the long lead time of systems, typically two years for contract award and three years from start of development to fielding. A third is that the inflexibility of complex software systems precludes rapid adaptation to changing mission requirements (e.g., Desert Storm). Product line practice can mitigate these deficiencies by creating a modular enterprise based on an architecture and assets, as shown in Figure 2.

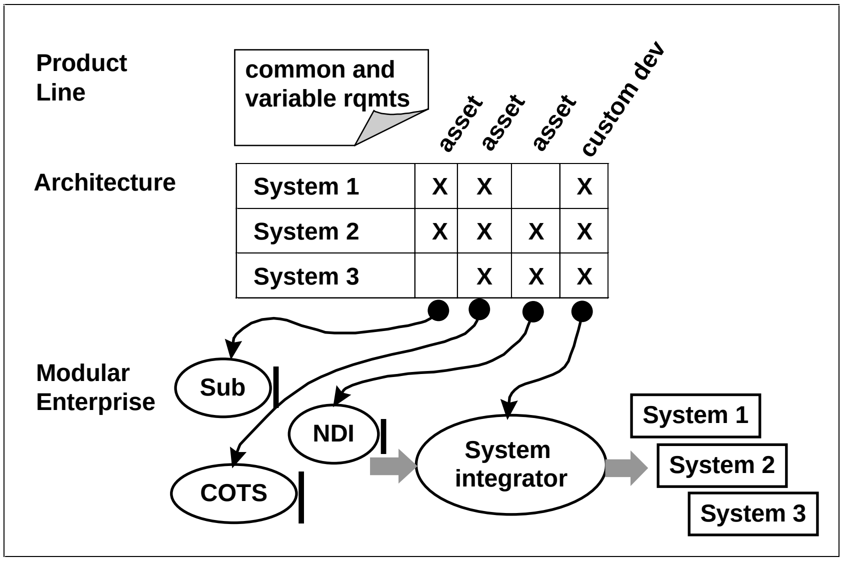

Figure 2: Product Line Practice in Acquisition

Common and variable requirements are defined for the product line and allocated to assets, which in turn are allocated to systems in the product line. The product line architecture becomes the basis for the work breakdown structure for the product line organization. This architecture facilitates decisions about which assets are to be built by subcontractors, which assets are to be commercial off-the-shelf (COTS) or non-developmental items (NDI), and which assets must be custom built.

The benefits of this approach are higher quality, flexible products that have a shorter cycle time. The product line approach yields economies of scope—a greater variety of products from a common set of assets, with less effort—because of its emphasis on reuse of product line assets. Since asset costs are shared across multiple customers (including many which are non-DoD), the result is a lower total cost of ownership and maintenance for the DoD. Less effort is required for each acquisition, and because the work breakdown structure (WBS) is based on an architecture, program managers have better overall program insight into the development process.

2.4.3 General Issues

Several issues confront DoD organizations when moving to product line practice: roadblocks, planning, contract interface, organizational structure, and management oversight.

Roadblocks. The STARS (Software Technology for Adaptable Reliable Systems) program identified several roadblocks[5] along an organization's path to institutionalizing product line practice [Foreman 96]. These include

- the existing policies, regulations, and laws governing acquisitions (there is a lack of capital investment policies that would fund generic needs in advance of specific requirements)
- the effort required to change the current culture (single-system focus, custom developments, institutional inertia, military rotation, etc.)
- the issue of product line ownership (for example, one contractor builds the architecture, another builds a system to that architecture, and a future failure may result in a battle over liability)

Planning. Planning for the definition, development, evolution, and support of product lines creates new milestones and dependencies for the definition and development of a product line, and its subsequent evolution and support. The key idea is that the architecture and development process must be defined before creating the WBS and allocating resources, and that the WBS is created before source selection and system development. Since development of the first system involves validation of the architecture and development process, extra development time must be allotted. New teaming relationships may be created; for example, an integrated product team may be established to define the architecture and WBS, and contractor teams may be created to build system increments from assets.

Planning must also address the following issues:

- Tradeoffs in scope: Widening the scope of a product line has the potential for greater cost avoidance but the corresponding increase in complexity may eventually wipe out this advantage.
- Careful management of the lineage of the products in the product line: Planned evolution of features and architecture must account for the relationship of each product to its predecessor and successor.
- Assignment of priorities: Priorities must be assigned to architecture "hotspots" (areas of change), such that there is agreement on what remains invariant in the architecture and what may be varied (for example, by customization).

[5] We agree that these are significant issues, but prefer to call them challenges instead of roadblocks since we know they have been overcome in specific government product line successes.

Contractor interface. Product line practice will alter the interface between a DoD acquisition organization and a contractor, as shown in Table 1.

Typical practice	**Product line practice**
Hierarchy of subcontractors	Architecture-based network of suppliers
Responsibility for design is centralized: the prime controls detailed design; subcontractors concentrate on coding.	Responsibility for detailed design is distributed: different organizations design or supply components that conform to functional and interface specifications
Purchaser has full property rights	Licensing is a common practice
No commitments beyond current contract; often adversarial relationship	Long-term relationships that involve sharing of product plans and early involvement in architecture design
Emphasis on lowest development cost	Emphasis on design to cost

Table 1: Contractor Interface Table

Organizational structure. Within the organizational structure of the DoD there are many sources for architecture and other assets to support product lines. Sources include research and development (R&D) centers within the material command, and program executive offices and program management offices within the acquisition executive. The essential questions are, “Who is responsible for the acquisition, support, and evolution of a product line architecture and assets, and who decides?”

The role of a product line manager is missing from the DoD organizational structure. In industry, a product line manager is typically responsible for the long-term business performance of a line of products, and has the flexibility to allocate resources to architecture, assets, or products. The DoD is constrained by an organizational structure that hampers the coordination and staffing of a product line organization and sustains a “stovepipe” approach.

Management oversight. Management oversight of a product line must be established and integrated in current program reviews. The purpose is to motivate program executive officers (PEO) and program managers (PM) to develop artifacts and processes that will help other program managers reduce total life-cycle cost. Without constant management attention at program milestone reviews, motivation to capitalize on synergies across system acquisitions will wane. Additionally, DoD policy fails to address the ownership issues of limited data rights and licensing associated with the asset-based approach of product line practice.

The next two sections describe possible approaches to product line practice within the organizational structure of the DoD.

2.4.4 Product Line Practice in a Program Executive Office

In this scenario, the Program Executive Office (PEO) is the product line organization. The PEO is responsible for the architecture and other assets while the individual program managers are responsible for single-system acquisitions. The benefits of this approach are the synergy possible across multiple programs because of the asset-based approach and the existence of a responsibility center for total costs.

The current acquisition environment raises some issues about the feasibility of this scenario. Typically, the power of the PEO is limited because funding for acquisitions is still done on a single-system basis, with no provision for multi-system assets. The organizational structure of the DoD also complicates things: often R&D expertise is concentrated in the materiel commands and is not in the acquisition executive chain. Apart from the "color of money" problem that this creates, it is also unclear what the role of R&D would be within the acquisition executive reporting chain. The PEO also has the challenge of obtaining the commitment of the PMs to product line practice without adding to the constraints under which PMs already operate.

The current funding structure is also a problem for the program manager who wants to do the right thing but is forced to live within the present limitations. A program manager is rewarded for cost and schedule performance; product line practice jeopardizes this by introducing new dependencies into the critical path. Additionally, any cost avoidance gains from product line practice will be used to reduce the PM's budget.

2.4.5 Product Line Practice in a Program

An alternative scenario places product line practice within a program rather than a program executive office. Here the PM is responsible for the architecture and assets, and multiple deliveries of different systems. Product line practice is implemented as a pre-planned product improvement (P3I) program that allows the PM to reset the program baseline based on, for example, increased understanding of problems and solutions, and the introduction of new technology.

The principal benefit of this scenario is that it is feasible within the existing culture and funding mechanisms. However, like the previous scenario, this scenario raises some issues, chief among them continuity and synergy. The long-term continuity of this approach is placed in jeopardy because of its vulnerability to funding cuts and the management rotation typical in programs. This calls into question the post-production development and support of systems; the PM's role typically ends when the first system is delivered. There is also little, if any, synergy with other programs because a multi-program perspective is simply not part of this scenario.

The planning horizon of the PM may be too small to be effective for product line planning. Also, the PM's tolerance of risk may make him or her balk at the significant risks associated with the adoption of a product line approach.

2.4.6 SEI Activities in the Acquisition Arena

The SEI is collaborating with several DoD and government organizations that are adopting a product line approach to system acquisitions. The benefit to the SEI from these collaborations is the opportunity to observe first-hand the practices that enable these organizations to be successful and to facilitate the maturation of practices where needed. These practices can then be incorporated into the SEI Product line practice framework for wider dissemination within the DoD.

An important activity for the SEI in its role as a technology-transition organization is the development of community awareness of product line practices. To that end, the SEI holds workshops such as this one and participates on the Product Line Issues Action Team (PLIAT). The PLIAT hosts quarterly meetings on specific issues related to the government's and contractor community's adoption of product line practice. Results from each meeting are posted on the PLIAT Web site, [http://columbia.ivv.nasa.gov:6600/pliat]. PLIAT is a sub-group of the ACM SigAda Reuse Working Group.

Guidelines to support acquisition organizations are being developed. In addition to the product line practice framework, members of the Product Line Systems Program produce technical reports, white papers, and educational materials. With funding and requirements from CECOM (Communications-Electronic Command), the program is currently creating a software architecture course targeted to DoD acquisition practitioners.

2.4.7 Some Final Remarks

To stimulate the discussions of the working groups, the following questions were posed:

- Are there other approaches for implementing product line practice in the DoD?
- What are some risk management checkpoints to include in an acquisition plan?
- At a minimum, what should be written in a contract for
  - a system that is developed from an architecture and assets?
  - an architecture?
  - a component or subsystem?

3. DoD Product Line Experiences: Digest of DoD Presentations

3.1 Introduction

Two speakers were selected to present DoD product line experiences. They were chosen in part because they provided two very different examples and two different levels of product line maturity. Robert Harrison described a legacy of successful product line-like approaches at the Naval Surface Warfare Center (NSWC). He showed that product line approaches are neither new to the DoD nor impossible. John Ohlinger described an ongoing product line development effort that was initiated in 1997 at the National Reconnaissance Office (NRO) for the satellite ground-based command and control domain.

3.2 Are Industries' Product Line Practices Sufficient to Make DoD's Acquisition Needs Affordable? Robert Harrison - NSWC

Robert Harrison of the Naval Surface Warfare Center (NSWC) addressed the mandate of acquisition reform as articulated by the Honorable Jacques Gansler with specific focus on what Gansler referred to as his "two critical questions:" "what we buy" and "how we pay for it."

The premise was that the NSWC, in particular, has developed computer programs as engineered products for over 30 years (at least in selected areas). Examples of these areas in NSWC practice and the corresponding years of corporate experience include Submarine-Launched Ballistic Missile (SLBM, 35 years), Naval Tactical Data System (NTDS), Combat Direction System (CDS), and Advanced CDS (ACDS, 30+ years), Missile & Gun Fire Control (30+ years), AEGIS (20 years), and Tomahawk (18 years). These systems provide a starting point for answering the questions raised by Dr. Gansler.

Navy experience with these systems suggests that successful projects share the characteristic of viewing software as an engineered product. Software must be considered as a major element of the entire system from the beginning, not developed and delivered as an independent entity. These Navy experiences also offer a rich set of lessons learned for other DoD systems regarding the essential importance of defining architectures early in the life of a system and the need for flexibility of the architecture to accommodate inevitable changes. Product line concepts, such as a common architecture and the use of open systems, present additional opportunities that should be a key to the long-term success of any system. The common engineering and management threads from these systems include

- a strong system engineering approach in which requirements are defined, implemented, and validated
- planning and resource management in which people, facilities, and development activities are scheduled, funded, and tracked using metrics
- a well-defined process in which management and technical processes are documented and followed
- stable and qualified teams that exist at all program levels

War-fighting systems have a number of common requirements, including long-range search, horizon search, target track, illumination, and mid-course guidance. Volatile data senescence, aperiodic event deadlines, and hard real-time periodics characterize the domain. Meeting performance requirements is a critical success factor. The negative consequences of getting "the right answer late" are far greater for the DoD, especially in the area of weapon systems. For all of these needs, automation is a key to success. Automation implies, in warfighting systems, more pervasive use of computing than in previous implementations of such systems.

Lessons learned at NSWC have made substantial contributions to insight in two distinct perspectives. The first perspective addresses developing a disciplined approach to system engineering. This disciplined system-development methodology needs to recognize a different set of needs at each phase of a system’s life cycle. Initially, the requirements are ill formed and minimal documentation is available. These requirements need to be evolved and become well defined to enable robust validation of the architecture, eventually leading to development testing and a stable, reliable engineered product. This type of disciplined approach evolved at NSWC during the 1970s.

The second perspective involves the capability to leverage the commonality in systems through systematic reuse of features at various levels of aggregation. Reuse, as a capability and technology, has developed more slowly, starting with low levels of aggregation, such as reuse at the subroutine and module level, and evolving, only gradually, toward more coarse-grained levels. In addition to common work products, some commonality also began to emerge by following a common process of specifying requirements and developing a contracting process.

This focus on discovering commonality across product families contrasted with an earlier practice in which each product was treated separately in terms of design, development, testing, and acceptance. In the earlier focus on individual systems, each system developer selected their own networks, computers, support services, and system composition services. “Stovepipe” systems resulted, in which complex interfaces had to be established to exchange information, no resource-sharing capability was available, and the cost of integration was high. Examples of such systems included AWS (Air Weather Service), JMCIS/C41 (Joint Maritime Command Information system/Command, Control, Communicate, Computers and Intelligence), and ATWCS/NWCS (Advanced Tomahawk Weapon Control system).

However, as war-fighting technology has evolved, naval missions have evolved causing a convergence of previously disjointed war-fighting domains. These domains can be viewed in terms of the following types of tasks and timeframes:

1. Planning, analysis, and training involve deliberate planning, rehearsal, and training. Planning is typically measured in hours, and the end of the planning phase typically precedes the start of the operation.
2. Battle management involves information acquisition and assessment of the situation. Battle management is typically measured in minutes, and it typically precedes actual engagement with a target.
3. Sensor to shooter involves targeting, weapons control, and striking the target. This phase typically takes place in seconds, and meeting hard performance requirements can mean the difference between mission success or failure.

The recognition of common feature requirements across a broad range of war-fighting systems prompted NSWC to define a common computing architecture. This architecture included software partitioning features, as well as software composition technologies. These two aspects coupled with open application approaches allow for distribution of processing demands across a broader base of computing assets while still maintaining the inherent coherency characteristics of tactical function solutions. This architecture takes advantage of both the concepts and the practices of the commercial computing industry as they have matured over the past 15 years. This architecture further enables a high-performance implementation by having the software architecture reflect the current state and emerging trends of computing hardware, interconnect, and middleware technologies rather than the ‘60s vintage equivalents.

Thus, strategic reuse is potentially raised to a new level of aggregation that is much higher than previously attained, namely, the computing architecture level. Validation of this computing architecture has been ongoing since 1994. Yearly experiments of increasing complexity and functionality have been used to examine the feasibility, performance, and characteristics of this new approach to reuse. A particularly visible DoD product line, the Baseline 7 Aegis Combat System, is the initial target for this computing architecture if the validation is successful in mitigating the necessary risks. The Aegis platform (cruiser and destroyer classes) will complete production in 2002, representing approximately a $100B investment in 70+ ships. This combat system included interfaces for components such as the electronic-sensing system, the sonar system, the fire-control system, and the vertical launching system. The Aegis Weapon System on each ship is based on a common computing architecture built from military components. This new computing architecture enables the use of commercial computing COTS (commercial off-the-shelf) products. This clearly constitutes a product line opportunity for fleet warfighting software if the computing architecture can support many, if not all, of the warfighting systems on a surface ship. A critical question concerns the implications of this approach for the rest of DoD.

The first step could be the definition of a notional architecture that captures the computing infrastructure needs of the entire ship as a single entity. This step would be conceptually similar to the approach used in providing cooling and electrical systems as part of the basic infrastructure that supports the entire ship. The second step would be to take a multi-shipclass perspective based on this architecture that could span CG-47, DDG-51, LPD-17, NSSN, CVN 77, and SC-21 ship classes. The variability in these multiple classes of ships typifies the increased system engineering complexity in the coming decades. Meanwhile the operational scope of warfighting is expanding from ship and force to encompass theatre-level engagements. The related computing challenges of this expanding scope are for mechanisms that address new levels of complexity management and scalable open systems that work together in coordinated harmony.

Lessons learned in validation efforts since 1994 have yielded the following insight. Through advances in interconnect technology, the performance of dispersed applications are well within weapon-system requirement timelines. The distributed computing infrastructure would need to have technical characteristics such as COTS-based open application designs, scaleability (in capacity and functional aspects), portability across vendor classes (this includes hardware, languages, operating systems, interconnects and middleware), fault tolerance, instrumented, and testable. For example, network technologies have enabled the advance from shared memory designs to point-to-point communication-oriented designs and networks. This advancement enabled evolving message passing to client/server and now to current distributed object technology such as CORBA (Common Object Request Broker Architecture). The future is seen as the powerful but cognitively complex “network component computing.” The article by Robert Freeman from the June 1997 issue of Distributed Object Computing was cited as a reference of this type of infrastructure [Freeman 97].

Complexity management is already a challenge for a product line. The use of one vendor's architecture and its components across multiple products is currently the state-of-the-art. At a higher level of aggregation, an *open* architecture with the required abilities needs to be defined. The question of how such an architecture would affect affordability was raised.

Currently, an “open system” is usually open only to the system’s vendor. This situation will have to change for future systems. The DARPA/Aegis High Performance Distributed Computing Program (HiPer-D or http://www.nswc.navy.mil/hiperd) was used as an example of a large-scale open system that was successfully engineered from both DARPA and commercial computing components and validated in the context of an AEGIS Weapon System performance timeline. The positive experience of the NSWC in this effort showed that commercial computing mainstream COTS products can meet most DoD requirements, given the proper software architecture is chosen. It was emphasized that the niche market process control or real-time COTS products were *not* used. Web applications and multimedia requirements have moved the commercial computing mainstream, in the Unix domain, into the real-time domain. Unique solutions have been marginalized to an ever-decreasing scope at the sensor interface realm.

The SEI has a unique opportunity to affect the future of shipboard computing. Cost and interoperability requirements suggest a common computing infrastructure could potentially address both of these concerns. A common computing infrastructure would be somewhat similar to what CelsiusTech did, but on a larger scale[Brownsword 96]. The SEI could promote the adoption of product line practices for computing architecture definition and development for surface ships. A business case could be developed in which this new approach could potentially provide leverage across classes of ships rather than replication as we have it today. Such an approach would address “what we buy” —a new product line called computing plants— and also address “how we pay for it”—a product line approach with the supporting business case to show either savings or cost avoidance.

3.3 Control Channel Toolkit: A Product Line Initiative in the NRO, John F. Ohlinger – National Reconnaissance Office (NRO)

The NRO has initiated a product line approach for ground satellite system software. As part of the planning for three new satellite systems, an initial feasibility analysis suggested a high degree of commonality within the domain and provided the incentive for developing a vision for managing the program as a product line. Because of the complexity of the systems, their high cost, and the long time horizon for fielding any specific system, top management has been closely involved in the decision making and has been willing to make the organizational changes needed to create an effective product line. A technical program office has taken a strong lead in developing the vision for the project, in communicating with top management, and in enlisting top management support as needed. The NRO provides an example of how to develop a product line from a top-down perspective, with careful planning and a series of incremental steps. The current status of the program is that of work in progress, so that lessons learned from this program will be of value to other complex systems with a long time horizon.

The Control Channel Toolkit (CCT) Program, begun in 1997, provides a common architecture and set of components from which individual satellite systems will evolve. The vision for the program is for reduced maintenance costs through the use of common code across multiple programs. To support this overall vision, CCT is being specifically designed to support a family of systems. CCT is based on the following principles: an open standards-based architecture, easy integration of contractor-specific and COTS products, flexible implementation options, and increased interoperability across programs. In addition, CCT is a focal point for enhancement and evolution. The goal is that CCT will become stable and robust due to use across multiple programs and that it will be available for future use on command and control programs. Thus, CCT is seen as initially forming a set of core assets for a family of ground satellite systems, and then evolving to become a more generalized platform for other command and control systems. The effort focuses on a domain that has a scope within the span of control of the program, but it leaves open the possibility of migrating later to a broader scope.

The program is structured into six increments. The first increment, consisting of the domain specification, domain definition, software architecture, infrastructure, and application program interfaces (APIs), has been completed. The program is well into increment two; increment three design review is scheduled for July 98. Increments are scheduled at approximately six-month intervals.

A domain analysis was performed to identify the objects, operations, and relationships that domain experts judge to be important for the ground satellite domain. This work developed a generalized specification, a domain definition, and a domain specification. The domain analysis provided initial confidence that sufficient generality existed for a set of core assets to form the basis for managing the systems as a product line. A shared paradigm of the problem domain was defined by consensus.

In contrast to examples from some other domains, there was not pressure to get immediate results from the domain analysis. The time table was long enough and the mission is critical enough that top management supported the long-range objectives that a domain analysis implies, with the understanding that this initial step would form the basis for longer term ROI. Cost models have been developed for the program. Expenses for development and sustainment are budgeted separately. It is expected that development of the core assets will increase by 0.3% ($100,000) over current development costs for 5 years. However, over a nine-year period, the program anticipates saving 27.8% ($15.9 million) in sustainment costs and 18.2 % ($15.8 million) in total costs (development and sustainment).

The CCT was developed as a set of core components for three products: DCCS (Distributed Command and Control System), SSCS (Standard Satellite Control Segment), and MALTA. A detailed analysis was performed of commonality among the three systems. This analysis found that commonality in satellite command and control and infrastructure components ranged from 49% to 89% among these three systems. Some of the product-specific services that were required by the individual systems included mission-specific altitude determination, scheduling, payload management, and hardware interfaces.

A common architecture was developed to define the system context. The system architecture became the key for analyzing components. A set of infrastructure services was specified based on the taxonomy of CORBA services and facilities. The CCT infrastructure was structured as an open-reference architecture to enable plugging in mission-specific components, such as status, control, and orbit. This pre-planned flexibility permits the substitution of components, the use of multiple contractors, varying COTS products, and technology evolution.

Three general issues have been raised in developing this product line. These concern the management of the baseline, ownership of assets, and specification of performance. The baseline management issue is being addressed by developing a single baseline across the multiple programs. Product variability is controlled by the individual programs. The advantage of this strategy is that the core assets are simpler to manage and maintain. This strategy is effective because of the stability of the systems after deployment. It is not important to accommodate to a rapidly changing environment. However, the strategy of a single baseline does have the drawback that it can sometimes be difficult to attain consensus on changes. A formal change control process has been established. Representatives of each program sit on the change control board. It is still, however, sometimes difficult to achieve consensus since the needs of each program vary and the board makes technical and financial decisions.

With regard to the ownership of assets, it was decided that the government would maintain ownership of the architecture and components. These assets are available to contractors for government use to encourage a richer set of compatible components that can be used to perform specific services. Additionally, the object and infrastructure definitions are being put into the public domain, hopefully encouraging other contractors to create their own domain objects that these contractors would then be free to market.

The issue concerning specification of performance has not been entirely resolved, which is not surprising because this is largely an open issue within the broader software engineering community. For satellite systems, performance is a critical quality attribute. Although the architecture can be designed for performance, the use of components, particularly COTS components, implies that a certain amount of control over performance and similar attributes is given up, since these components are essentially black boxes from the perspective of the system. The reusable components created as part of CCT are, however, being “characterized“ for performance allowing reuser design teams to be able to estimate system performance.

4. Working Group Reports

The following sections contain reports from the working groups. These working groups covered software engineering practices, technical management practices, enterprise management practices for acquisition organizations, and enterprise management practices for contractors.

4.1 Software Engineering Practices

The SEI product line practice framework identifies several software engineering practice areas important in the development and acquisition of product lines. Members of the group voted for which practice area to explore in more detail from the following list: requirements management; domain analysis; architecture exploration, development, and evaluation; mining existing assets; component development; testing; effective utilization of COTS; performance, reliability, and security engineering; software system integration; and asset evolution. The three areas chosen were (1) domain analysis; (2) architecture exploration, development, and evaluation; and (3) asset evolution.

The following sections summarize the results of each of the working groups in their investigation of these three areas. A synopsis is also provided.

4.1.1 Domain Analysis

Domain analysis was discussed first. The initial discussion centered on basic issues. For example, what exactly is domain analysis in the context of product lines?

4.1.1.1 The Practice Area

One proposal was to define domain engineering as an amalgam of domain analysis, domain design, and domain implementation. Some group members felt that architecture could be used as input to domain analysis. Another proposal was to characterize domain analysis as being similar to requirements analysis, but with emphasis on variability analysis and in a much larger scope.

Since the benefits of using domain analysis were also discussed (see below), a preliminary discussion also explored the products of a domain analysis. Representative work products are a domain dictionary or lexicon, domain-scoping rules (used to determine what is “in” the domain and what is “out” of the domain), and class diagrams with use cases/scenarios.

4.1.1.2 Differences for Product Lines

Domain analysis in the case of product lines may not be very different from "regular" domain analysis if there is such a thing. By definition, domain analysis takes a broader view than a single product. What may be different is the analysis of the variability of a single product as it evolves within the product line over time.

4.1.1.3 Barriers for the DoD

It is unclear if the fundamental barrier to domain analysis for product lines within the DoD is any different than for non-DoD organizations. For example, the lack of expertise within particular domains is a common problematic phenomenon. Within the DoD, the accelerated change of personnel in particular positions means there is a continual loss of so-called “corporate memory,” further exacerbating the situation. There is a non-convergence on any single domain analysis process. The lack of domain analysis and its products forced many organizations to continue to rely on “gurus” for domain expertise. For an acquisition organization, the reluctance on the part of contractors to divulge their domain expertise can be problematic; contractors may view such information as part of their competitive advantage.

There was recognition of the need to understand the benefits and ROI of domain analysis. However, this is sometimes hampered through confusion over what exactly the work products of a domain analysis (as discussed above) are. Historically, some of the work product outputs have been ill defined and contextual. There is also the question of who owns the work outputs: the acquirer or the contractor.

There is also the possibility that non-DoD organizations have constraints that the DoD does not. For example, funding for DoD projects is not subjected to the same type of market-driven competitive risk as commercial projects. This is not to say that DoD projects operate without funding constraints; rather, the constraints and risks are somewhat different than in the commercial marketplace.

4.1.1.4 Mitigation Strategies

The group discussed several mitigation strategies to overcome some of the barriers to domain analysis within the context of product lines. One solution was to glibly characterize all barriers as enterprise management acquisition issues, rather than software engineering practice issues, so the whole issue could be dismissed by the group. For example, altering the funding strategy of DoD projects might address some of the barriers.

More practically, it was felt that it might be more prudent to adopt the recycling theme of “think globally, act locally.” The NRO's CCT effort was cited as a successful example of this approach. They seem to have limited the scope of their domain analysis to those aspects of the problem that are under their control. Proper scoping is a critical success factor in both domain analysis and product lines in general. Scoping heuristics could be a help here.

Interestingly, this approach may give rise to "stove-piped product lines" that may need to be merged into a meta-product line in the future, but at the current time the scoping makes the domain analysis practical and the expected leverage significant.

Another strategy suggested was to employ the approach used in the WarSim2000 project. In that instance, the request for proposal (RFP) specified mitigation strategies directly: the DoD, as acquirer, assumed ownership of all work products. This included a two-phased acquisition process where the second phase involved the execution of the "best of breed" domain analysis submitted by the contractors during the first phase. This was viewed as a win for the DoD because it created consensus among the competitors in subsequent work.

4.1.1.5 Issues

Several general issues concerning domain analysis in the context of product lines were discussed. Can domain analysis be done for just a single system, or is it inherently a multi-product activity and hence well suited to product lines? If domain scoping is performed too narrowly, the economies of scope inherent in product lines may not be realized. Performing "good enough" domain analysis (e.g., Lucent's "commonality analysis") may be appropriate in certain circumstances. The phrase "reference architecture" in connection with both domain analysis and an architecture for a product line means many things to many people. Does domain analysis more properly concern the problem space or the solution space? Similarly, are non-functional requirements in the problem space or the solution space?

4.1.2 Architecture Exploration, Development, and Evaluation

The second practice area discussed concerned aspects of software (and system) architecture. The distinction between system and software architecture is not always appreciated. It was felt there was a need to inform the system engineering community about the advances in software architecture in general and product lines in particular.

4.1.2.1 The Practice Area

The architecture practice area is very broad. There was some discussion on where system architecture ends and where software architecture begins. The exploration aspect was characterized as analyzing architectural styles with respect to selected quality attributes. Existing architectures can be explored for potentially reusable assets (see 4.1.3) during domain analysis (see 4.1.1). The SEI’s Architecture Tradeoff Analysis Initiative was used as an example of active work in this area. The SEI's software architecture analysis method (SAAM) for architecture evaluation was briefly discussed as a relevant practice [Bass 98].

4.1.2.2 Differences for Product Lines

The exact differences between software architecture for a product line versus a single product was not clear to the working group participants. The notion of a “reference architecture” was proposed, being more abstract or generic than a “regular” architecture.[6] The confusion on this issue is due in part to evolving terminology.

A product line architecture was viewed as more complex, due to the need to represent variations in the product line. A robust and repeatable configuration process for variation-point management is needed; there seemed to be little consensus on how to do this. The increased variability may also increase the tradeoff contention between potentially conflicting quality attributes and the need to carefully analyze architectural decisions.

4.1.2.3 Barriers for the DoD

Many of the barriers to software architecture for the DoD are shared by non-DoD organizations. Much of the problem is rooted in the relative immaturity of the field. For example, existing evaluation strategies are still being developed; some are subjective and manual (e.g., SAAM), and hence not repeatable in an automated manner.

The ill-defined nature of what is a “view” of software architecture contributes to the problem. There is general agreement that architectural views are important, but the proliferation of views is a barrier to their adoption. There is no clear guidance regarding which standards to use within the DoD, the services, and the external software engineering community. Deciding which views are important (and in what context), what they actually represent, and how they should be represented are all open issues.

Even the more fundamental question, what is software architecture, is still not resolved for most people. Many definitions exist, but architecture means different things to different people. For the DoD, there is a definite need to understand how DII COE (Defense Information Infrastructure Common Operating Environment) relates to the notion of software architecture elsewhere.

4.1.2.4 Mitigation Strategies

To overcome the field's relative immaturity, it was suggested that the DoD work closely with established external authorities. For example, the IEEE (Institute of Electrical and Electronics Engineers) has a working group on architecture descriptions. The IEEE is a respected organization with sufficient influence to mature some of these areas to the benefit of the DoD. Part of this exercise involves getting the systems engineering community more closely involved in the creation of architecture guidelines and standards.

A more hands-off mitigation strategy that was proposed was for the DoD to simply wait for the state of the art to mature. Once it becomes an established state of the practice elsewhere, the DoD could revisit the architecture topic.

---

[6] A discussion of this architecture can be found in *Software Architecture in Practice* [Bass 98]

4.1.2.5 Issues

No general issues on this topic were recorded.

4.1.3 Asset Evolution

The final practice area discussed was asset evolution. This is an important topic because it is related both to salvaging assets from existing legacy systems and to evolving individual assets in a product line over its lifetime. One of the first issues raised was what actually constituted an asset.

4.1.3.1 The Practice Area

The notion of exactly what an “asset” is was discussed. It was decided that there are two tiers of artifacts than can be considered assets from a product line perspective. The first tier consists of the architecture, components, and a repeatable process for variation point (configuration) management. The second tier consists of processes, test cases, documentation, and the domain model.

4.1.3.2 Differences for Product Lines

Asset evolution for product lines was considered more complex than for the single product instance. This analysis is based in part on the increased complexity of managing multiple versions of the same asset in different variants of the product line over time. Propagation of changes in core assets to multiple deployed products in the product line requires more configuration management maturity.

When core assets are changed, testing their new functionality and ensuring thorough regression testing also becomes more complex. Core assets are typically more generic than product-specific assets, making exercising all their functionality more problematic. The example of one organization refusing to certify extremely parametizable Ada generics was mentioned as an example of this phenomenon.

There was also a discussion of when to roll back product-specific features into the product line’s core assets. This rollback is in fact a similar exercise to mining a legacy system for startup core assets. This comparison is related to the distinction between application engineering (product features) and domain engineering (product line assets).

The question of acquiring new assets that must be merged into an existing product line was also raised. The example of Microsoft's purchase of Vermeer's *FrontPage* product for Web page development and Web site management was used. How did Microsoft manage to incrementally merge the Vermeer product into the *Microsoft Office* family of products? This was thought to be an example of how planning for the insertion of new technology into an evolving product line can be challenging.

4.1.3.3 Barriers for the DoD

The DoD shares at least one common barrier with non-DoD organizations regarding asset evolution in a product line context: the “clone and own” practice. This is typified by software engineers copying selected code fragments (or even larger-grained components), modifying the code in a relatively minor way, and then inserting the new code into an existing program. When maintenance is needed on the original source fragment, it is very likely it will also be needed on all of its copied and modified versions as well. This is one of the worst cases of “reuse” that is in fact a detriment to asset evolution.

The opposite case of this type of software misuse is the “not invented here” syndrome. It manifests itself as a reluctance on the part of contractors to accept the core assets of a product line and evolving them. The desire to start a green-field effort, rather than rely on the work of others (which may be difficult to understand, and hence difficult to modernize), may be too great. Economic factors also play a role in this issue: current contracting practices may in fact discourage acceptance and evolution of assets.

For an acquisition organization, the issue of “who owns what” was raised. Asset ownership reflects on many central issues during product line evolution. When one of the core assets of a product line requires modification, who is allowed to carry out the modification? Who is qualified to carry out the modification? Who is responsible for carrying out the modification?

4.1.3.4 Mitigation Strategies

One of the mitigation strategies discussed was the use of an architecture review board. Members of the board are charged with accepting or rejecting changes to the product line's architecture and to its core assets. In this way, a centralized group maintains control of asset evolution.

Another mitigation strategy was better requirements management. This was an underlying theme for several of the practice areas and emerged as a common theme for the entire workshop. If users can be convinced to accept an 80% solution for less than 50% of the cost of a 100% solution, more core assets could be used essentially “as is.”

For better management of asset variability, it was felt there was a need for better tools, perhaps from research, to be made more available and to be widely adopted. The Graphical Layout User Environment (GLUE) application from the DSSA-ADAGE (Domain Specific Software Architecture – Avionics Domain Application Generation Environment) project was cited as a good example of powerful configuration management tools well suited to product lines.

To address the “not invented here” syndrome, the notion of “cooperating competition” was suggested. This is an emerging industrial practice where different companies that compete in the marketplace cooperate on selected aspects of the core technologies. Research consortium members use this approach to create the common infrastructure for a new product area, then compete on top of this infrastructure.

4.1.3.5 Issues

The only issue discussed was how to refer to a subset of related product instantiations in a product line. Some participants felt that a new term might be required for this subset.

4.1.4 Synopsis

It was not surprising that most software engineering barriers are non-technical in nature. Organizational and managerial issues can have far more influence on a project than technical issues. This observation is also reflected in the workshop’s structure: three of the four working groups focused on non-technical software engineering issues.

The value of domain analysis to the DoD needs to be better quantified. A more practical approach may be to adopt “good enough domain analysis.” This analysis might be carried out at the same time as preliminary architecture exploration. This combination would help scope the domain analysis by limiting its budget and its timeframe for completion. Long periods of perceived inactivity before any tangible benefit is seen from domain analysis may be counter-productive.

Few organizations have significant experience with the extraction of architectural assets. The methods for evaluating software architecture are immature but promising. There is a potential for long-term gain through the use of generation technologies that are connected to system descriptions based on some form of architecture description language. It may be that the best thing the DoD can do in this regard is to take a “wait and see” approach until the area matures.

Requirements management was seen as a critical area for product line practice. Negotiating for 80% functionality for 50% cost may be a difficult notion for some to support, especially given the unique requirements of some DoD systems. The inability to compromise on requirements is an impediment to product lines. A related issue was the somewhat surprising view that large-grained reuse may in fact contribute to the "not invented here" syndrome, further eroding the success of product lines. As the granularity of the asset increases, it may be more domain specific, but it may not cut across multiple products in the product line if not properly designed. This increase of granularity, therefore, affects economies of scope.

4.2 Technical Management Practices

Of the practice areas described in the product line practice framework that belong to technical management, this working group elected to turn their attention to three:

- program acquisition management
- core asset analysis (which evolved into a discussion about legacy system migration management)
- metrics, data collection, and tracking

4.2.1 Program Acquisition Management

This group interpreted program acquisition management as referring to the product-line-level practices necessary for an organization to subscribe to a product line effort. This definition is intentionally general: the organization mentioned may be the one launching the product line, or it may be one that oversees a project desiring to become a part of an already-existing product line effort.

4.2.1.1 The Practice Area

Acquisition management for a product line includes the following five activities:

1. *preparing a business case* for why the organization is pursuing the product line strategy. Typically, it was reported, the product line approach is adopted because of direction from a higher authority, or internally motivated because of a desire primarily for decreased life-cycle costs.
2. *building a strong understanding of the scope* of the product line: what products will (or, in the future, might) be in the product line, and what products will be excluded. The scoping, which is a subset of domain analysis, identifies the variabilities across the family members as well as the commonalities that all members share. The variabilities will either be specific, in which case particular products are identified, or they may be general, to allow for future growth of the family. In other words, the contracting organization can either procure specific variants, or procure the capability for future variance. The scope of the product line will have a strong impact on the architecture that the family members will share, for the architecture is the mechanism by which the variability will be achieved.
3. *writing an RFP* to do the work necessary to build (or join) the product line.

4. *building or commissioning an architecture* that satisfies the requirements for the individual members of the product line, as well as containing room for future expansion of the line. In this area, “ownership” of the architecture is a critical issue. If the acquisition organization dictates the architecture to the contractors who produce the products, then it must

   – take all responsibility for the adequacy of that architecture
   – make sure that it has the technical expertise at hand to craft that architecture
   – assume at least part of the liability for products in the product line that adhere to the architecture

   If, however, the acquisition organization commissions the contractor to provide the architecture, then it must make sure it has the technical expertise at hand to evaluate the architecture for fitness of purpose and suitability

5. *finding/using legacy assets* to help populate the core asset base of the product line.

4.2.1.2 Barriers for the DoD

Barriers to effective program acquisition management include many conditions that are endemic to DoD organizations, such as

- three-year billet rotations in positions of authority (Launching a product line requires a strong vision and an effective managerial hand over the long term.)
- lack of funding for long-term life-cycle costs, as opposed to funding for up-front meeting of new requirements
- the fact that some projects are funded through multiple sources, each of which demands accountability and may not wish their funds to be used to save someone else’s money (by building generic core assets, for instance)
- a shortage of domain experts within the DoD
- the fact that changing contractors in mid-stream is exceptionally difficult within the DoD, as opposed to industry where contracts may be ended and new suppliers chosen with little effort
- the strongly hierarchical management chain within the DoD, which makes it difficult for projects to cooperate with each other to exploit commonality (Joint collaboration can only occur by enlisting the support of the lowest node of the hierarchy that is a common superior of both projects.)
- lack of flexibility to handle teaming issues (The government usually lacks the authority to form the best teams of contractors, and there are usually difficulties about handling intellectual property rights [and sharing of responsibility and liability] among the team members.)

4.2.1.3 Mitigation Strategies

Mitigation strategies include the business case and domain scoping steps mentioned above, before the RFP is released. Also, a staged procurement approach may work well for a product line. First, the domain scoping is completed. Then, an architecture is solicited. Then, either a core asset base is built (acquired), or individual systems in the product line are built, and so forth. These separate steps provide for go/no-go decision points along the way. Further, collected metrics allow the opportunity for strong successes to be documented for increased visibility. Finally, starting each new step will ensure constancy of vision and purpose by bringing the overall effort squarely back on the product line trajectory. Finally, it was suggested that best-value (rather than lowest-bid) evaluation criteria be used to award product line contracts.

4.2.2 Core Asset Analysis and Legacy System Migration

4.2.2.1 The Practice Area

Core asset analysis and legacy system migration are activities by which organizations examine existing systems within the boundaries of the pre-defined product line and establish plans to use them to develop the product line assets and systems. This effort is generally driven by the extremely high maintenance cost of legacy systems; merging two or more systems that exhibit similar funtionality or that are built from common assets should lead to reduced maintenance costs.

A prerequisite is a unified business plan for product line development (i.e., a common set of product line goals, a mission statement, etc., over a hierarchical management chain within the DoD). The business plan sets up the core asset analysis and legacy system migration by

- *providing a strategy* for the reduction of system development and maintenance costs. The business plan helps an organization understand the short-term vs. long-term benefits and costs. In a DoD environment, this cost savings may be more apparent in the long-term maintenance phase than in the system development efforts.
- *helping the organization focus on objectives* for adopting a product line approach, common architecture, and common infrastructure. Without the business plan, an organization would have no justification for embarking on the core asset analysis and legacy system migration.
- *providing a powerful incentive* and aid for moving in the product line direction.
- *conveying personal authority*. This is extremely important when attempting to get cooperation among the different stakeholder units.

With the motivation, focus, and authority provided by the business plan, how should a given a set of projects, programs, and systems be merged into a product line, and how is that made to happen? Core asset analysis helps to define current assets; and legacy system migration outlines how to get to the desired end state. The core asset analysis must

- *identify candidate systems within the scope of the product line*. Will the domain scoping exercise provide enough input to the effort? Potential systems for inclusion into the product line may not reside within the chain of command for this effort. Effective program acquisition management and metrics are necessary in this situation. Acquisition management will provide the leverage to include the candidate system, and metrics will provide incentive for inclusion.
- *identify the amount of commonality across the systems within the scope of the product line*. In a DoD environment, the systems may have been developed by different organizations. Therefore, these systems may first need to be described in a common language before commonality can be recognized. Some of this information may be provided by a quick domain analysis.
- *identify any uniqueness that might disqualify a system's participation in this effort*. Product line managers should not try to force a bad fit. But inclusion in a product line is not an all-or-nothing proposition; perhaps an outlying system can join at a later time, or perhaps it can contribute a common element or use a common component. Disqualifying uniqueness may come from using an esoteric operating system or special security requirements. A different programming language should not, however, disqualify a system from joining a product line.
- *identify the underlying architecture from each of the legacy systems*. What is the impact on the possible predefined architecture? Will the architecture have to be compromised to get changes made? The scope of the product line will have a strong impact on the architecture that family members will share.

Legacy system migration must establish a plan to evolve legacy systems into a product line. The legacy system migration plan must take into account the following issues:

- Do you have management control over all the projects? If so, what authority do you have? For instance, can you require the different projects to use the same computing platforms? If you do not have management control over some of the projects, have you marketed to other managers/customers to acquire some control over their projects? Do you have cooperation among the different stakeholder units?
- Do you have customer buy in? Are you able to get adequate requirements from them? Do they mind getting a new version of a system they are using? Do they care how you implement their requirements (e.g., in a "stovepipe" or product line development)?
- Is this effort developing its own product line or is it the result of an RFP that has been written to use someone else's product line? Is there enough documentation (or knowledge) to know how to use the other product line (e.g., practices to allow a project to subscribe to a PL effort)? How many COTS products, GOTS (government off-the-shelf) products, etc., were included in the RFP, and is it propriety?

- Do you have a product line group that owns the core assets? How will the assets be maintained? Are the core asset developers also the system developers? Will a contractor evolve the current legacy systems into a product line? These questions lead to the same “ownership” issues as those discussed in the previous section for architectures, (e.g., asset responsibility, liability, and technical expertise).
- Who owns the architecture? Is the owner of the architecture different than the owner of the assets? Initial architectures need to evolve; because of this evolution, who is responsible for this evolution and growth? Who will modify the assets (components) to fit into this evolved architecture? Who paid for the architecture? Who paid for the assets?
- How hard is it to implement new ways of doing things within your organization (i.e., change the culture)?
- What type of funding model is in place? Are you planning a phased implementation plan of small successful steps, each of which can be used to gain additional support and funding to proceed to the next step? Or are you planning to tax each one of the common projects to build the common assets?

4.2.2.1 Barriers for the DoD

Barriers to core asset analysis and legacy system migration are mostly technical in nature. Being able to mine the assets from the systems within the scope of the product line and evolve these assets with the product line architecture into a family of systems is extremely difficult. However, from the technical management perspective, the barriers center around the ownership of the legacy systems and the product line architecture and assets. Will contractors and acquisition organizations work among themselves and overcome the responsibility and liability issues? These problems are especially thorny in the context of multi-contractor acquisition efforts.

4.2.3 Metrics, Data Collection, and Tracking

4.2.3.1 The Practice Area

The most pertinent question to ask concerning metrics and data collection is "Why?" That is, what is the goal of the effort? What information is sought and for what purpose? The working group asserted the following as the primary goals of data collection:

- to help make the decision whether to launch the product line or not, by projecting whether the projected savings would be worth the cost
- as supporting evidence to sell or market the product line approach to any number of stakeholders who will need to be convinced that the sponsoring organization is on the right track
- to help the acquisition agency monitor the development effort, and apply mid-course corrections throughout the life of the effort where necessary (the most usual reason to collect data in non-product-line efforts)
- to help make technical decisions, such as whether or not it makes sense to expand the scope of the product line to encompass another legacy system that is perhaps not a part of the original product line vision

A major goal of metrics collection for the DoD is to move DoD organizations towards the product line way of doing business. There are many stakeholders who need supporting evidence that the product line approach is in their best interests. On the contracting side, these stakeholders include

- the *sponsor* of the product line effort. The sponsor needs to be sold on the basis of lower life-cycle costs across the family members, as well as short-term benefits that will accrue. The sponsor can also be sold on the basis that future deliveries of new releases are much more likely to be predictably on schedule. Finally, it can be demonstrated that a product line will increase his or her competitive advantage in the application domain by enabling the organization to take on new requirements faster and cheaper than other organizations competing for the same budget dollars.
- the *customers* of the product line (and the individual products within it). Customers need to know that the product line approach will result in higher quality and reliability of systems; that those systems will be delivered faster and cheaper over the long run and with less risk; and that the long-term costs of configuration control, testing, and maintenance will decrease by spreading the cost over the family.
- the *users* of the products within the family. Users may need to be sold on the approach because a product line approach may very well change the look and feel, functionality, and requirements of systems they are already using. They can be sold on the basis that less training will be required, and the training that is required will in fact help them become fluent in many systems instead of just one. It is also likely that user requirements will be met using systems that work together and that new requirements will be accommodated faster and more reliably. It may be the case that if legacy systems are merged during the product line creation, that fewer people will be required to use those systems; hence, there may be attrition in the user community. This has to be carefully managed and offset against the advantages for the user community at large.

On the contractor side, vested stakeholders include

- *project managers*. Although they may be wary about the prospect of building products whose cost is less, project managers can be incentivized by higher award fees for quality deliveries. Also, since contracts are not usually priced as a function of how many people work on the project, the prospect of smaller staff requirements to produce these products should be attractive. An enticing incentive is the notion that the conractor organization will be building large, generic components that are by definition useful across many different systems in a domain, and will have access to (if not the responsibility for creating) the architecture in which these components are used. Both of these incentives represent highly prized capital assets that can be used with little or no change to launch (or at least feed) other business areas of the company.
- *developers*. System-builders often yearn for the opportunity to build high-quality systems using sound engineering methods. A true product line effort offers this opportunity. Job satisfaction is likely to rise. In addition, the developers will by definition be working on a family of systems at once; hence, their skills become more widely applicable and general, and they therefore become more marketable. This needs to be balanced against the possibility that fewer developers will in fact be needed.

A marketer of the product line may need evidence to support all of the claims, and more, depending upon the audience. He or she will also want to make a strong claim about the organizational maturity of the groups acquiring and developing the systems in a product line.

Once the goals for metrics have been established—that is, what the product line owner wishes to be able to show has been articulated—then the individual metrics can be identified and plans made for their collection. This will require a metrics plan. The product line sponsor/owner should be responsible for producing such a plan. The metrics plan should contain the following:

- a clear articulation of the organization's business goals for engaging in the product line effort (Without knowing what the organization is trying to achieve, it will be impossible to determine whether or not progress is being made.)
- a description of what metrics to collect and their frequency of collection, their traced-back relationship to the organization's business goals, and a convincing argument as to why those metrics will in fact shed light as to how well the organization is progressing towards its aims
- information to assure the validity of the collected metrics (How can project management be sure that the collected data mirrors reality?)
- a statement as to the intended usage of the collected information. (This may include attribution issues, as well as who will be allowed to see the data and for what purpose. There may be data that contractors may agree to provide only under conditions of non-attribution, or selectivity of purpose. But in general, this section dictates the "data flow" of collected information in terms of the reports that will be generated as a result and to whom those reports will be circulated. This defined data flow is intended to avoid the case where data are collected simply to satisfy a contractual requirement, but never examined.)
- a projection of how much it will cost to collect the required data—this is never free—and how this cost will be borne
- a statement of how the data collection activity will be monitored, to assure continuity and validity, as well as contract compliance
- predictive models for how the metrics are expected to behave over time, so that progress against the goals can be tracked (Without these predictive models, it will be impossible to use the data for project management, because there will be no standard of "good" or "bad" trends.)
- a plan for modifying this plan, as changes and accommodations are inevitable as the information is synthesized

4.2.3.2 Barriers for the DoD

Barriers to sensible metrics and data collection are primarily inertial in nature: There are few (if any) examples of product line projects that have collected product-line related data and used it to their advantage; there is not even a commonly agreed to set of product line metrics that should be collected. Data collection is one of many aspects of the product line paradigm shift, and paradigm shifts are never immediate.

4.2.3.3 Mitigation Strategies

Mitigation strategies include establishing a workable set of metrics for product lines (through workshops such as this one), using those metrics wherever possible or advantageous in product line efforts, and then getting the word out as to how those metrics contribute to the success of the product line endeavor.

4.3 Enterprise Management Practices for Acquisition Organizations

This working group focused on enterprise management for acquisition organizations. Its members began by considering the baseline set of 11 enterprise management practice areas identified in the SEI product line practice framework:

- ensuring sound business goals
- providing an appropriate funding model
- performing market analysis
- developing and implementing a product line concept of operations
- achieving the right organizational structure
- assuring proactive management
- building and maintaining appropriate skill levels
- managing the organization's customer and supplier interfaces
- ensuring inter-group collaboration and communication
- risk management
- technology management

The goals of the group were to select three or four key enterprise management practice areas from this list and describe them from an acquisition perspective to promote the adoption of product lines by DoD organizations. The group chose to explore the following areas:

- providing an appropriate funding model
- developing and implementing a product line concept of operations

- achieving the right organizational structure[7]
- building and maintaining appropriate skill levels[7]

In the course of discussing these practices, the group identified two additional practice areas they consider essential to enterprise management of product lines:

- building and communicating a business case
- developing and implementing an acquisition strategy[8]

All of these practices relate to enterprise management because their orchestration occurs primarily at the corporate level, transcending individual organizational units and projects. In the working group discussions, we attempted to reach a common understanding of the key activities that characterize these practices and some of the associated barriers. In the process of discussing these practices, the group tended to focus on identifying the gap that exists between commercial practices and DoD practices, and potential mitigation strategies for overcoming these enterprise management barriers.

The following sections include a summary of preliminary discussions, each of the selected enterprise management practice areas, and a list of open issues for further investigation.

4.3.1 Preliminary Discussion

There was a lively discussion about what is meant by the term "enterprise." Clearly, there are many levels of enterprise within the DoD: an organization within a particular command, an R&D activity, a group of activities organized around a particular domain (e.g., air/ship/ground-based/space systems), one of the services, or all of the armed services. However, for the purposes of our group, we recognized that the enterprise is what is meaningful to each organization in terms of their charter and authority, span of control and influence, and the funding (under their purview) that empowers them to effect and manage change. The consensus was that the enterprise be defined within the constraints of funding; this emphasizes the importance of a funding model.

Beyond the scope of the working group, participants felt that there were felt to be many strategic impediments that need to be addressed at the very highest enterprise levels for a product line approach to be fully effective in the DoD. These include high-level DoD policies, acquisition regulations, and built-in barriers to sharing funds and resources within and across services and project boundaries (e.g., stovepipe funding and organizational structures that are orthogonal to a product line approach).

---

[7] These practices were not discussed as thoroughly as others due to time constraints.
[8] It was discovered during the working group reporting session that the technical management working group also discussed these practice areas as described in Section 4.2.

The working group concluded that a useful post-workshop action would be to

- succinctly identify the product line and acquisition barriers at the higher enterprise levels
- determine how these service-wide issues can best be presented to the Pentagon decision makers to influence OSD and effect meaningful change at the DoD corporate level

4.3.2 Building and Communicating a Business Case

4.3.2.1 The Practice Area

Building a business case will play a strategic role in deciding how to change the organization's current mode of operation to one that is supportive of a product line approach. This approach will also substantiate that product lines are appropriate for the business area and that it should be communicated across the enterprise to obtain initial buy in. Before developing a business case, sound business goals should already be established. These goals provide a common understanding of what the enterprise hopes to achieve by adopting a product line approach, and articulate the basis for determining whether the effort is successful.

Key to the business case is identifying the scope of the product line and the potential savings (over the life cycle) and contrasting it with the current way of doing business.

Several participants volunteered that in the current DoD downsizing environment, which is characterized by a scarcity of funds and limited in-house expertise and resources, a business case for a product line approach is easier to create. DoD participants believe it represents the only strategy they can foresee (in the current environment) that has the potential to cope with the escalating demands for developing systems “faster, better, cheaper.”

Building a business case based on hard evidence from pilot projects was a common theme of the working group. Although outside experience may be sufficient to obtain support for an initial pilot effort, hard evidence from internal efforts is considered a must for transitioning the concept within the organization.

Prerequisites the group identified for building a business case include the following:

- Managers need to be selective about where and when to apply a product line approach; multiple mission areas may need to investigate different approaches.
- Solid justification is needed; anticipated savings/pay-back on potential programs must be clearly identified and include the candidate systems that will be part of the product line.
- Market surveys may play an important role in corroborating the soundness of the concept, but the business case needs relevant examples of product line success or examples of demonstrated savings on pilot efforts within the organization.
- Incentives for achieving efficiency must be addressed in the business case.

4.3.2.2 Barriers for the DoD

Some of the barriers that were foreseen relate to organizational structure and skill levels which correspond to the enterprise management practices described in Sections 4.3.4 and 4.3.7, respectively.

4.3.2.3 Mitigation Strategies

A recommendation for mitigation is to include a rough draft of the product line concept of operations with the business case to provide insight into how the product line concept will work within the organization. The business case must substantiate that the considerations that are leading an organization towards adopting a product line approach are, in fact, valid for the enterprise.

4.3.3 Providing an Appropriate Funding Model

4.3.3.1 The Practice Area

Once an organization has established its business goals and built its business case, a funding model (i.e., funding strategy) is needed. The funding model identifies the funding sources that can realistically be used to initiate and sustain support for the product line. The funding model must address both the development of the core assets (i.e., domain engineering) and the development of derivative products using these core assets (i.e., application engineering). Developing a suitable funding model involves clearly laying out a product line approach over multiple systems and identifying the life-cycle cost savings and benefits to senior level management to obtain their buy in. The funding model must be compatible with the product line concept of operations and indicate how the projected funds for these systems can be pooled and aligned to sustain the product line over its life cycle.

One of the participants stated that “seed money” is essential to overcoming objections, and without it there may be no practical way to get started and demonstrate savings. Although there was general agreement that the product line startup risk should ideally be addressed through R&D, the current funding structure often works against this.

Suggestions for creating a funding model for a product line approach include the following:

- getting a clear picture of what you are trying to do, learning the bounds, and working them to your advantage within your sphere of influence and control
- obtaining grass roots support and convincing sponsors and projects of the superiority of the product line solution rather than management directing a “technical best” solution
- reallocating a portion of the funds from programs that will benefit from the product line approach and using those moneys to fund the product line
- aligning funding to support the long-term plan and justifying seed money from other areas (including using R&D funds for pilot projects)
- creating a horizontal funding line as a firm part of the budget based on product line feasibility and return on investment

4.3.3.2 Barriers for the DoD

A major barrier that was cited is that the organizational unit responsible for developing the concept of operations is not usually in charge of the funding model. One participant suggested that the DoD should institute a policy change requiring sponsors to direct funds for a product line approach and ensure the type of funding is commensurate with product line maturity. These points reemphasize the need for a product line funding mechanism that can align sponsorship with horizontal areas that cut across projects.

Other barriers that were discussed include funding instability, parochial views of organizations opposed to the pooling of funds, restrictions on the use of funds (e.g., color of the money), and a lack of incentives for an enterprise approach to systems development that transcends organizational units and commands.

4.3.3.3 Mitigation Strategies

One of the individuals in the group pointed out that a recent change in Army policy may also apply to the other services. Under AR-70-1, the charter for software centers now allow PMs (program managers) to go directly to contractors. Consequently, Army centers are in direct competition with industry for program work. This may be viewed as a formidable barrier or as an opportunity to justify investing in a product line approach, which presents a PM with a viable option for cutting costs and reducing development time. It also affords a common basis for achieving interoperability with related systems (that other PMs are responsible for) that fall under the cognizance of the parent PEO (program executive officer).

A novel idea for obtaining funding centered on identifying what parts of product line interests intersect with operations and maintenance (O&M) and charging them as an O&M service. An example is using O&M funds to reengineer legacy assets to obtain core assets or product-specific components.

4.3.4 Achieving the Right Organizational Structure

4.3.4.1 The Practice Area

The group agreed with the commercial data suggesting that one of the greatest challenges in implementing a product line approach is achieving the right organizational structure. Implementing a product line approach is dependent on managing horizontally (i.e., in a matrix mode) across projects to produce products that are part of a family built around a common architecture and core set of assets, as well as managing vertically to create individual products. This presents a real challenge for DoD organizations that are traditionally highly “stovepiped” with regard to their sponsorship, project structure, funding, resources, contracting, and reward system. As one participant so aptly stated, "we [in the DoD] are horizontally challenged."

A primary consideration in organizing a product line approach is structuring the organizational units responsible for developing and sustaining the core assets versus those responsible for developing derivative products using the core assets. These organizational considerations raise many questions including the following:

- Who has final authority over the architecture?
- How are changes to the core assets controlled and funded?
- Who ensures the architecture will be responsive to project-specific requirements?
- Is some form of centralized acquisition support available via an umbrella contract or does each project have to fend for itself?

Organizational changes need to be carefully orchestrated with appropriate training and with building a technical and process infrastructure. The wrong organizational structure can defeat solid product line technology and processes. On the flip side, however, an abrupt organizational change without the essential underpinnings in software engineering and technical management practices to execute a product line approach is also a recipe for failure. Achieving the right organizational structure involves both determining the appropriate structure and a strategy to implement it. It is also the case that the definition of the right organizational structure may change as the product line matures.

4.3.4.2 Barriers for the DoD

The challenge in creating such a suitable organizational structure is to avoid making wholesale changes that can be unduly disruptive to the culture of the work place, while at the same time trying to align the organization with product line goals that cut across project efforts.

Other barriers that the group identified include resistance to change, lack of incentives, incompatibilities with the existing reward structure, and the lack of champions in the sponsor and project arena who are willing to commit to a product line approach.

4.3.4.3 Mitigation Strategies

While the group acknowledged that there may be many barriers to achieving the right organizational structure, the organization has to stay focused and concentrate on identifying how it can mitigate these barriers within their purview of authority and influence. For example, one approach that was suggested for mitigating organizational barriers is to create a virtual organization to implement a product line approach. A virtual organization can strategically position itself to take full advantage of the synergy afforded by capitalizing on a common set of assets. Another suggestion was to start small. Choose a well-scoped product line with modestly scoped organizational change rather than attempt a risky enterprise overhaul.

4.3.5 Developing and Implementing a Product Line Concept of Operations

4.3.5.1 The Practice Area

Once management gives the go ahead for embarking on a product line approach, the development of a product line concept of operations (CONOPS)[9] represents a major milestone. The CONOPS is a key work product that defines the participating organizations and processes for implementing a product line approach. It identifies product line stakeholders and clearly describes their roles and responsibilities. Typically included in the CONOPS are appropriate mechanisms for sustaining the product line over its life cycle, folding back improvements, interfacing with customers, and other support functions that are essential for achieving long-term product line success (and for avoiding regression back to the development of a set of unique systems). The CONOPS addresses the operation of both the acquisition and development groups as well as the role of the product line architecture. It is sometimes a useful vehicle to obtain the early buy in of the domain experts who may question the practicality of the approach.

4.3.5.2 Barriers for the DoD

Several participants commented on the DoD propensity for starting out at too grandiose a level. Instead, the need for incremental evolution of product line organizations was stressed. The preferred mode of operation is to have one product line beget another as opposed to starting product lines across the enterprise. It was also noted that it may be more advantageous to get current users to sell the product line approach to potential users rather than the organization selling it.

4.3.5.3 Mitigation Strategies

Since a CONOPS describes how the product line concept will work in a particular environment, the group viewed it as a practical way to demonstrate how the organization will mitigate barriers and overcome resistance. In light of this, the CONOPS developers should strive to

- promote buy in of stakeholders that will scale up from the individual to the enterprise level
- clearly show the relationship of the product line organization to the existing enterprise organization
- compensate for lack of horizontal infrastructure support—DoD is still entrenched in a stovepiped project, funding, and acquisition model
- address the risks of identifying a new mode of operation where significant cultural changes and new job descriptions may be required

---

[9] The CONOPS applies to the development and evolution of the product line and should not be confused with the traditional DoD concept of operations that describes the operational system.

- incorporate domain and systems engineering infrastructure support for core asset creation and evolution
- cover life-cycle aspects of sustaining the product line, but clearly identify how the organization can incrementally transition to new modus operandi

The group encouraged the SEI to continue developing example work products such as a generic concept of operations. It was viewed that guidance documents, like a generic CONOPS that will describe the roles of the sponsor, acquirer, user, and developers, are needed to provide the kind of insight that is considered critical to DoD adoption of product line practices.

4.3.6 Developing and Implementing an Acquisition Strategy

4.3.6.1 The Practice Area

All of the participants indicated that developing and implementing a suitable acquisition strategy is critical to achieving a product line approach in DoD. One of the key perceived differences in implementing a product line approach in the DoD environment, as opposed to commercial ventures, is the predominant role that acquisition plays. The acquisition approach defines how to deal with product lines within the contracting process and still be responsive to unique project requirements. One participant suggested that the contracting process provides a lot of freedom; the challenge is to find the appropriate contractual vehicle and recognize that the early buy in and endorsement of the contracting officer and contract negotiator play a very pivotal role in the acquisition approach. An effective acquisition strategy for product lines must address

- development of a product line architecture and other common core assets
- procurement of COTS components for core assets
- reengineering or mining of legacy system assets to obtain a startup set of core assets
- sustainment and evolution of core assets
- development of products using the product line architecture and common core assets
- procurement of COTS components to be incorporated into products to meet project-specific requirements
- reengineering or mining of legacy system assets to obtain components to be incorporated into products to meet project-specific requirements
- sustainment and evolution of derivative products built from core assets

4.3.6.2 Barriers for the DoD

The two key issues for the DoD participants in developing a product line acquisition approach are

- how to acquire architecture-centric core assets so DoD activities can be responsive to multiple program needs
- how to competitively acquire derivative products without endangering contractor interests or the government's ability to maintain control over the core assets

A common concern of the group is that proven acquisition approaches (i.e., ones that are repeatable and responsive to life-cycle requirements) constitute a major unknown, and will need to be gradually developed, refined, validated in actual practice, and disseminated. Guidance is especially needed on how to include architecture issues in an RFP. Other areas where it was indicated that acquisition guidance is needed to support a product line approach include

- developing an acquisition plan and selecting a suitable contract vehicle(s) that is compatible with the product line concept and takes full advantage of acquisition reform measures
- preparing a solicitation package and specifying an appropriate technical evaluation criteria
- including precautionary measures to minimize the risk of a protest before or after contract award
- incorporating contract incentives to sustain contractor motivation (after contract award), and to encourage cooperation and efficiency commensurate with their role as a product line team player

All of these measures are aimed at overcoming the traditional mindset of a single-system acquisition program strategy and accommodating multiple project efforts.

4.3.6.3 Mitigation Strategies

In general, the group thought that having real examples of acquisition work products (e.g., acquisition plan, RFP, statement of work [SOW]) would provide them with much needed insight. These work products could also potentially serve as models that could be adapted (or tailored) to meet their own enterprise-specific needs.

4.3.7 Building and Maintaining Appropriate Skill Levels

4.3.7.1 The Practice Area

All the participants cited the importance of building and maintaining appropriate skill levels. These skills must cover the entire spectrum beginning with building the business case through development of a product line architecture, acquiring and developing derivative products, and sustaining and evolving the core assets and derivative products throughout their life cycle.

The group identified several skill areas they believe are essential to a product line approach and are of significant concern to enterprise management:

- business skills for making, communicating, and selling the business case at all levels of the enterprise (and billets for people having these skills)
- product line management
- domain engineering (including architecture development, architecture evaluation, and systematic reuse)
- acquisition including experience writing RFPs, SOWs, and technical evaluation criteria

4.3.7.2 Barriers for the DoD

Participants felt there was a decided lack of skills development in the areas listed above. Few have a solid understanding of architecture. Too few are adequately equipped to construct and communicate a solid business case suitable for the DoD.

4.3.7.3 Mitigation Strategies

To overcome these skill barriers, the group stressed the critical need for training courses and instructional materials. To illustrate the importance of training to DoD, one of the participants stated that attending the workshop was worthwhile if for no other reason than he learned about the course the SEI is developing on software architectures for DoD acquisition practitioners.

A common theme the working group participants again expressed, related to improving skill levels, is their desire to obtain exemplary work products—ones that can serve as models for their consideration in adopting a product line approach for their enterprise.

4.3.8 Issues for Further Investigation

A number of enterprise management issues for further investigation were captured during the discussions. They include how to

- change the DoD corporate mindset and stovepipe-driven infrastructure to accommodate business across organizational boundaries and allow sharing of funds and resources
- obtain buy in and funding for a product line approach without a strategic infrastructure in place to support activities across the DoD in adopting such practices
- build a solid business case that a contracting officer can support and champion through the acquisition approval chain
- transition from multiple, competing contractors to a consolidated acquisition approach that is fully supportive of product line objectives and encourages cooperation between contractors
- formulate incentives that can sustain the contractor and promote cooperation with the acquiring activity as well as other contractors developing core assets or derivative products

While these issues clearly reflect the DoD acquisition-based environment that the participants work in, they are supplements to and variations to the open enterprise management issues (and needs) that their commercial counterparts have identified in our previous workshops.

4.4 Enterprise Management Practices for Contractors

4.4.1 Introduction

This working group examined the enterprise-level practices of contractors developing products for DoD product line organizations. Like the other working groups, this group was chartered to describe product line practices listed in the product line practice framework and point out how they differed from traditional practices, identifying barriers for the DoD, and mitigation strategies. However, in reviewing the list, members of the group observed that at the enterprise level, contractor product line practices seemed to be tightly coupled to the acquisition approach of the DoD customer. At least for traditional, single-system acquisitions, the business and funding models, the organizational structure and operations, the resource development and allocation processes and other senior management practices seemed to be based on the customary acquisition practices of the DoD. Therefore, the working group elected to treat the enterprise practices as a whole, in the context of an acquisition, rather than describing the differences, issues, and mitigation strategies practice by practice.

The group described the enterprise practices for the contractor in a traditional, single-system acquisition, and then defined a generic acquisition approach for a product line. With this as a background, the group was able to describe a general organizational structure for a contractor. Enterprise issues and barriers that were identified centered on the business motivations of the acquisition organization and contractors, particularly regarding the development and maintenance of a product line architecture. In developing mitigation strategies for these issues, the group discovered that the enterprise practices of a contractor depended more on the acquisition environment and goals of its main customers than it originally thought. Contractors would implement different enterprise practices depending on the product line and market power of the acquisition organization. Consequently, the group described practices for three scenarios that are believed to be among the major cases for product line acquisitions.

In the description of the contractor's enterprise in the context of single-system and product line acquisitions, the working group touched on the following enterprise elements. (The related practice areas are given in parentheses.)

- *organization structure* - the functional entities in the organization and their reporting structure (achieving the right organizational structure)
- *organization operations* - the inputs, activities and outputs for the entities in the organization (developing and implementing a product line concept of operations)
- *contract interface*, specifically the funding mechanisms and deliverables placed under contract (managing the customer and supplier interfaces)
- *business model* - the strategy for gaining profits and sustaining competitive advantage; the engagements, transactions, and business motivations of the organization described (ensuring sound business goals and strategies)
- *technology management*, specifically the role of R&D in the organization (technology management)

This section of the report is organized in the following manner. First, the enterprise practices for contractors are briefly summarized in the context of a single-system acquisition. Then, a general acquisition process for a product line is presented and a generic organization structure that supports this process is introduced. The other enterprise practices are discussed in the context of three acquisition scenarios.

4.4.2 Traditional Single-System Acquisition

In typical single-system acquisitions involving new development or a major system upgrade, the contractor usually signs a cost-plus contract to design and code a system to specific program requirements. Profit is controlled to motivate contractor performance: it may be tied to milestone or to cost performance. Engineering change packages are written for any enhancements or upgrades not originally put in the contract.

The business model is straightforward. The contractor either functions as a prime systems integrator or as an engineering subcontractor to the prime. Costs that cannot be directly charged to a contract are viewed as overhead and are kept to a minimum. Improving the software process and building generic components are usually not directly chargeable to the contract. In a single-system acquisition scenario, the contractor is typically organized as shown in Figure 3.

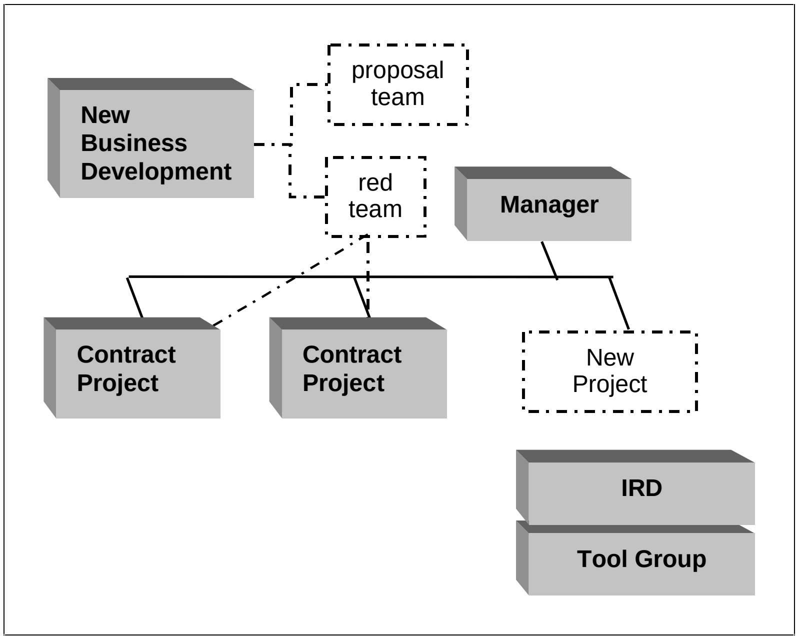

Figure 3: Traditional Contractor Structure and Operations

Members of the group noted that the contractor's organization mirrors the organization of its customer. Each contract project is a "stovepipe effort," characterized by a high degree of autonomy and focused on a specific point solution. Because the funding for each project is tied to a contract, the manager over the projects administers company overhead.

Whenever a new request for proposal is announced, new business development launches a proposal team. The proposal is reviewed by a red team composed of engineers from the different contract projects. If the proposal is awarded, then a new project is created.

Though this acquisition process and organization structure work for single projects, they inhibit synergy among projects. Contractors typically might address the inhibitors by adopting one or more mitigation strategies such as

- moving key engineers from project to project
- lifting code from project to project

Only when the contractor has a sufficient number of projects, does it have sufficient overhead to fund small internal research and development (IRD) projects to capitalize on synergies in process and products. Many contractors have a separate tool group responsible for defining the development environment for each project. Depending on the importance of software process improvement, this group may also function as a software engineering process group.

If the contractor is also awarded the follow-on operations and maintenance contract, then it has more motivation to create and capitalize on commonalties in the product and process. More time may be spent up-front to make essential subsystems more modifiable or reusable. A test bed and other development facilities may be planned. The contractor will invest more in training and skill development.

4.4.3 Product Line Acquisition Context

Before elaborating on the enterprise-level product-line practices of a contractor, the group felt it necessary to describe a generic product line acquisition approach. Drawing on experience, the working group reasoned that a contractor would orient its enterprise to accommodate the different acquisition stakeholders and their priorities. By defining a generic approach, it hoped to identify some of the essential stakeholders and their outputs, and with that insight, describe a general enterprise structure for a contractor. Accordingly, the focus shifts in this section from the contractor to the acquisition organization.

A product line acquisition strategy at a minimum involves the creation of a product line manager in the government organization and the identification of one or more contractors who are responsible for developing an architecture, core assets, and multiple systems[10] (see Figure 4).

[10] Product line acquisition strategy was also discussed in two other working groups, as described in Sections 4.2 and 4.3.

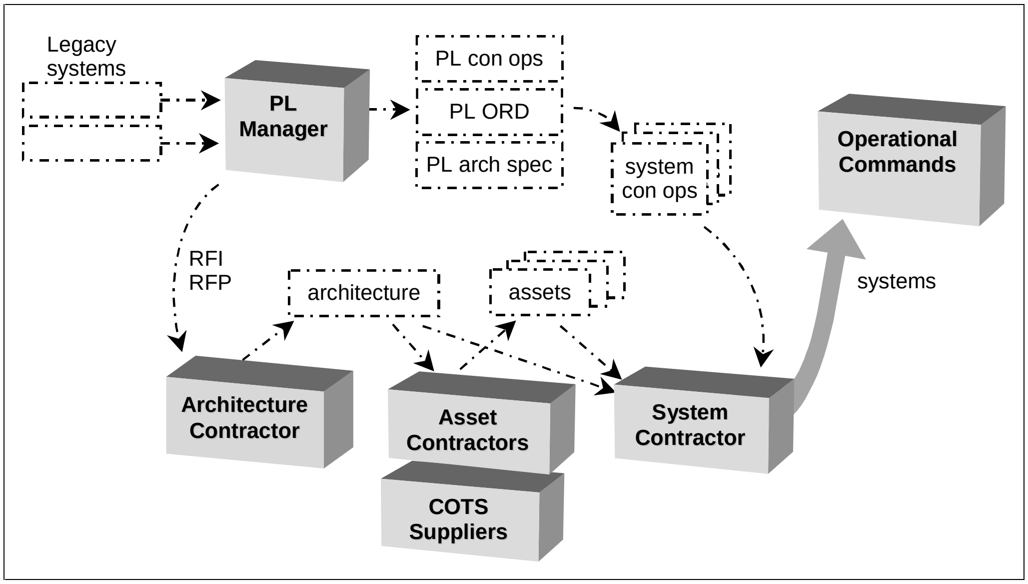

Figure 4: An Example Product Line Acquisition

A product line acquisition process is divided into at least two separate phases. The first phase entails developing or acquiring the architecture. The second phase involves developing or acquiring the assets and developing systems using those assets and the architecture. Figure 4 also shows some of the products produced and used in the acquisition process.

In the first phase, the DoD product line organization develops a product line concept of operations (PL CONOPS) and operational requirements documents for the product line (PL ORD). The product line organization then issues a request for information (RFI) from contractors regarding development of a product line architecture. Contractors respond with white papers discussing architecture specifications that support the concept of operations and operational requirements for the product line and that are compatible with legacy systems. Based on the knowledge gained, the product line manager may issue a request for proposal (RFP) and specification for developing an architecture. A contract is awarded, and a product line architecture is developed.

In the second phase, the architecture is used to define the work break-down structure for asset development and system integration. Some components may be licensed from COTS suppliers. The development of assets and the first few systems may be concurrent and may involve one or more contractors. A program office will develop operational requirements and a concept of operations for an individual system, which is then given to a system contractor. The system contractor develops software for the custom requirements, integrates the other assets, and delivers the system to the operational commands. Because multiple systems are to be developed, work may be issued under indefinite delivery, indefinite quantity (IDIQ) contracts.

The working group did not define how the architecture and assets are kept technically and functionally up-to-date.

4.4.4 Contractor Product Line Structure and Operations

Acquisition strategy, the contractor's structure, and operations change, to some extent mirror the DoD product line organization (Figure 5).

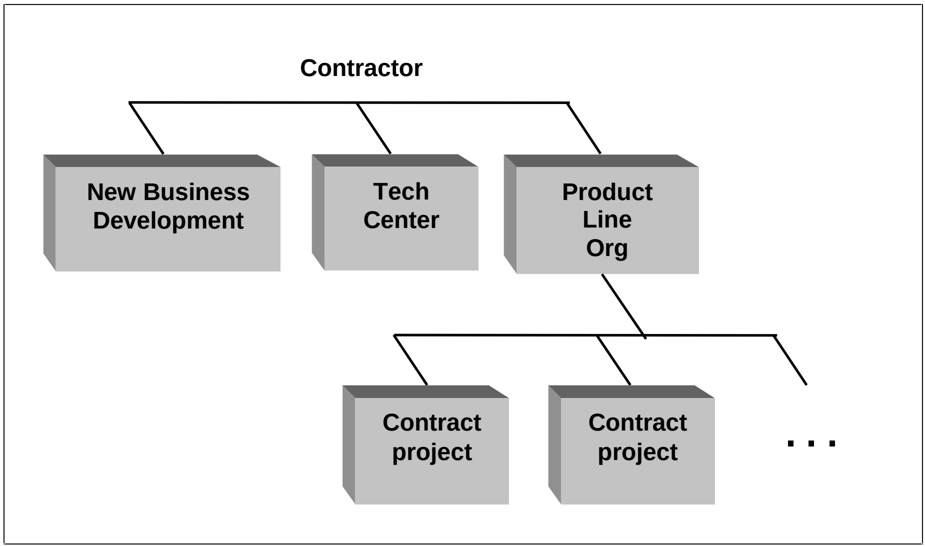

Figure 5: Product Lines in a Contractor Organization

The contractor creates a product line organization that is responsible for all contract projects in the product line. These projects may develop assets or integrate systems. If a project is new, then the product line architecture will be used in development. If the project already exists, then analysis of the feasibility of migrating to a product line approach is performed. If the project is nearing completion, then the effort to adopt the product line architecture and assets may be too costly. If the project is part of a maintenance contract, then a reengineering plan may be developed to migrate the system to the product line architecture and available assets. Depending on the acquisition scenario, the architecture may not be developed in the product line organization.

Tasks for a new business development group also change. This group now works with the DoD product line organization to define upgrade packages that exploit the capabilities of the architecture and assets, and also negotiates proposals that minimize the customization of the architecture.

Internal research and development becomes more explicit. If the contractor is developing multiple assets and/or systems for a DoD product line, then there is a critical mass of funds to devote to product improvement. Multiple, overlapping contracts create a stable source of funds and an opportunity to exploit synergies across products in the product line. The contractor will establish a technology center to develop new business opportunities through the research and development of assets and technologies. The center also plays a primary role in developing, transitioning, and sustaining the product line architecture.

4.4.5 Product Line Enterprise Issues/Barriers

Comparing the traditional enterprise to the product line enterprise, a few issues concerning the implementation of enterprise practices for a product line come to the forefront. Most issues center on the role of the architecture in product line acquisitions.

The first issue concerns the contractor's business model. The traditional business model is no longer applicable. Contractors now have multiple and different business opportunities. They can focus their business on one or more of three services:

- lead contractor for architecture
- subsystem/asset developer
- systems developer/integrator

The presence of choices raises several questions:

- What are the criteria that would lead a contractor to choose one business opportunity over another?
- Would not most contractors opt to lead architecture development for the contract security and competitive advantage it provides over asset developers and system integrators?
- If a contractor develops the product line architecture, would it be prohibited from developing assets or systems?
- What is to keep a contractor from developing an architecture that is not decomposable, whose components cannot be partitioned over contracts?

Although not explicitly discussed, some of these questions are addressed in the next section.

The second issue concerns shared commitment. For a product line approach to be successful, the working group believed that contractors and the acquisition organization must share responsibility and commitment to cost avoidance through systematic reuse. How is this achieved?

The third issue concerns contractor buy in of a product line architecture. Systems integrators will not be motivated to use a mandated product line architecture that does not reflect their design practices. System development risks and costs will be greater, particularly if the contractor has no experience and assurance that the architecture is valid. The architecture will be “dead on arrival.” How is this scenario avoided?

A fourth issue concerns the acquisition context. The choice of contractor structure, operations, and contract interface depends on the contractors' role in developing a product line architecture. The generic product line functions in the enterprise were described earlier, but there are significant differences in skills, staffing levels, development activities, and responsibilities when the contractor is involved in architecture development and evolution.

Having all interested contractors collaborate on developing a product line architecture may resolve the above issues, but this may not be feasible in all cases. For example, the architecture may be an open systems standard, or only one contractor may have the needed expertise. In addition, are there cases when the performance and schedule risk of an architecture by consensus is too great?

4.4.6 Mitigation Strategies

The set of issues and barriers seems daunting. Recalling that the contractor's enterprise depends on the acquisition context, the group chose to explore strategies that mitigate these issues by delineating different scenarios for developing a product line architecture. Three scenarios are discussed below. The first defines when a contractor-proprietary architecture may be appropriate and describes the enterprise activities for an organization developing such an architecture. The second scenario defines program context and goals for a collaborative approach to architecture development, and describes some of the enterprise activities of contractors involved in collaboration. The last scenario defines program context and goals for an architecture based on commercial-off-the-shelf (COTS) components and describes the related enterprise activities.

4.4.6.1 Scenario 1: Proprietary Product Line Software Architecture

In this scenario, the architecture is developed internally by a team composed of engineers from the technology center and contract projects and technical managers from the product line (dotted lines in Figure 6). New business development puts together a product line proposal team similar to a team created in a single-system acquisition (see Figure 6).

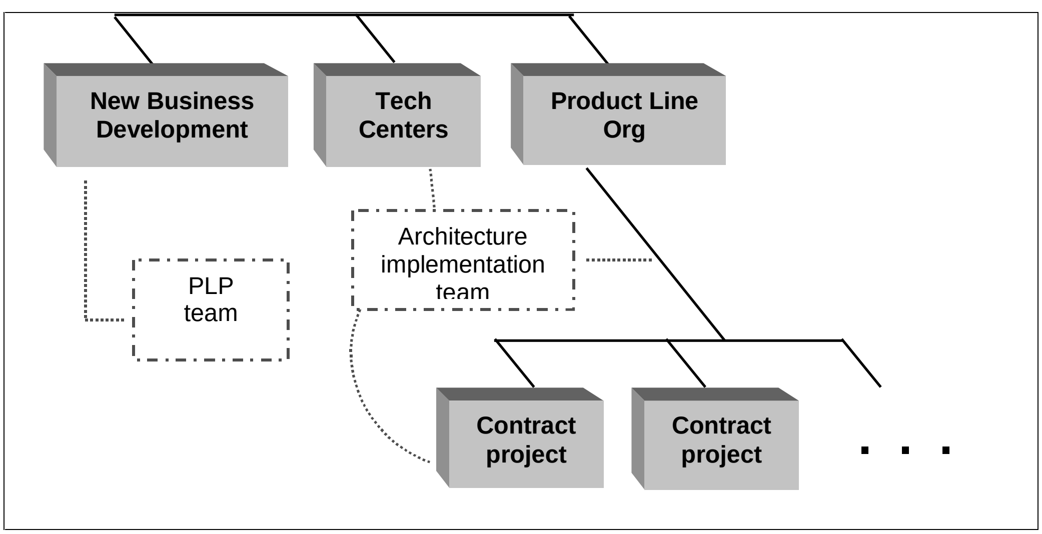

Figure 6: Contractor Enterprise Developing a Proprietary Architecture

An acquisition program may adopt this scenario regardless of the risk of contractor lock-in for the following reasons:

- The programs are top secret.
- Few systems are developed.
- Working with a single contractor is less risky.
- The program manager is primarily seeking to
  - leverage common functionality
  - obtain greater flexibility in functionality
  - and gain better program insight

A couple of issues with this approach concern the lack of interoperability and a drift away from COTS products.

4.4.6.2 Scenario 2: Product Line Software Architecture Based on Consensus and Collaboration.

In this scenario, the architecture is developed by a team composed of system and software architects from different contractors (see Figure 7).

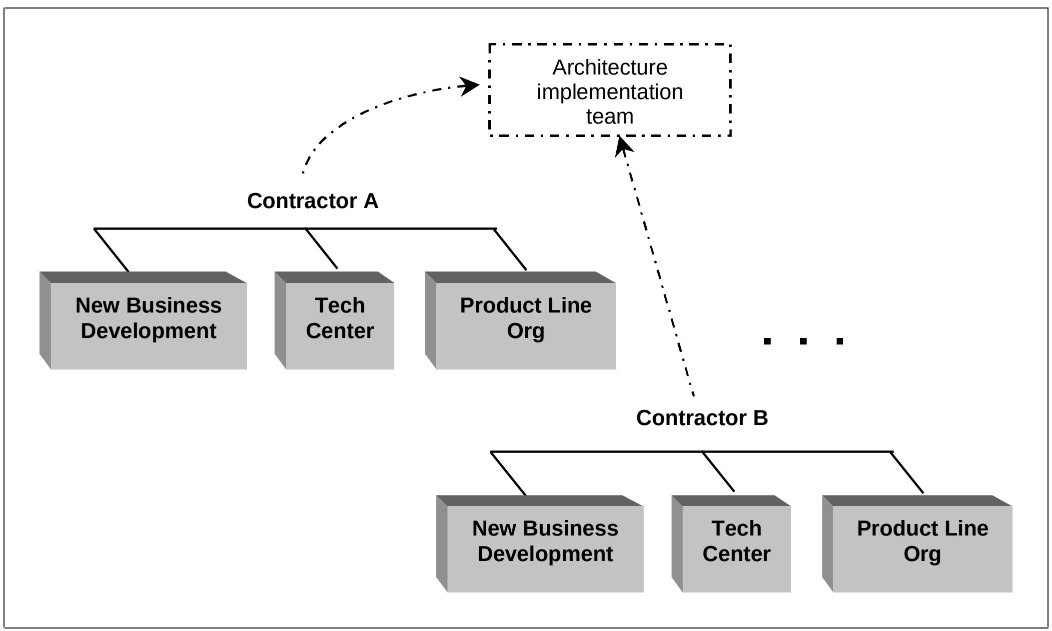

Figure 7: Contractor Enterprise Developing a Collaborative Architecture

An acquisition program may adopt this scenario for the following reasons:

- There are requirements for open competition.
- Many systems will be developed.
- Multiple contractors have developed similar systems in the past.
- The program manager is primarily seeking to
  - lower cost of ownership by distributing costs across multiple programs and suppliers
  - create redundant sourcing

This approach requires broad participation to ensure innovation and openness, yet expert control to prevent contractors from inserting uncommon capabilities in the architecture. This approach also requires a long lead-time before systems development.

4.4.6.3 Scenario 3

The product line software architecture is based on open system concepts and standards and on COTS products.

In this scenario, the architecture is developed in three phases (see Figure 8). DARPA (Defense Acquisition Research Projects Agency) or department-level research and development proves an architecture concept based on commercial technology. A research and development center in a service materiel command applies the technology to the weapon domain, developing a “virtual” design and providing transition management to the architecture implementation team in a contractor organization. Applying technology to an application domain is often referred to as small “r” and big “D.” The research and development center also may be responsible for refreshing the architecture with new technology.

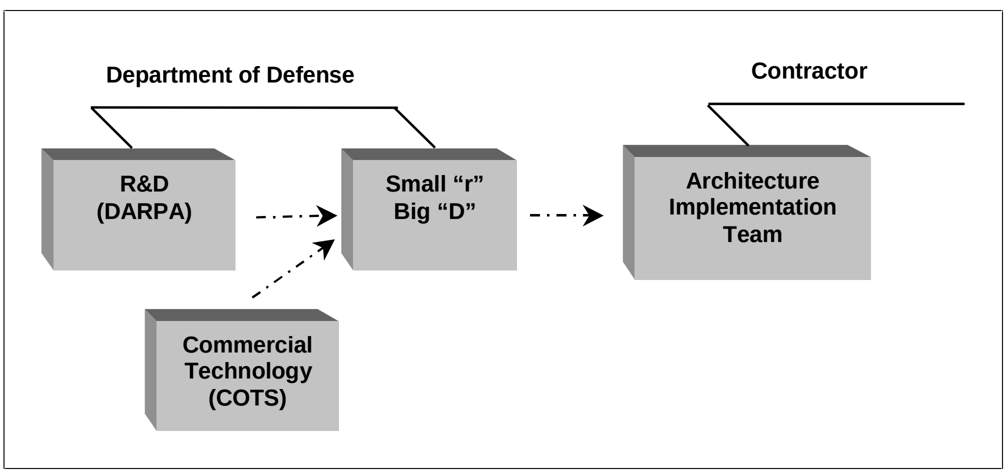

Figure 8: Three Phase Architecture Development

An acquisition program may adopt this scenario for the following reasons:

- It is mandated to use COTS products.
- It is necessary to re-architect the system to get high performance (incremental migration of COTS products fails to achieve required performance) .
- The program manager wishes to lower the cost of ownership.

Key issues with this approach include little program control over future capabilities and updating the architecture as COTS products quickly evolve.

4.4.7 Discussion

The group discussed at length the applicability of these scenarios: when would these different scenarios come into play? Are the scenarios the "essential few" cases? To these questions, the working group explored the possible kinds of product lines in the DoD, then discussed the market forces that would lead an acquisition organization to adopt one scenario over another.

An acquisition organization has a choice regarding where to capitalize on product synergies. As shown in Figure 9, an acquisition organization can create product lines at different system implementation levels. There may be a product line for

- a family of systems delivered to end users such as command and control centers
- a domain-specific component such as an orbital telemetry subsystem
- a network transport layer and underlying operating system that would be included in the computing infrastructure

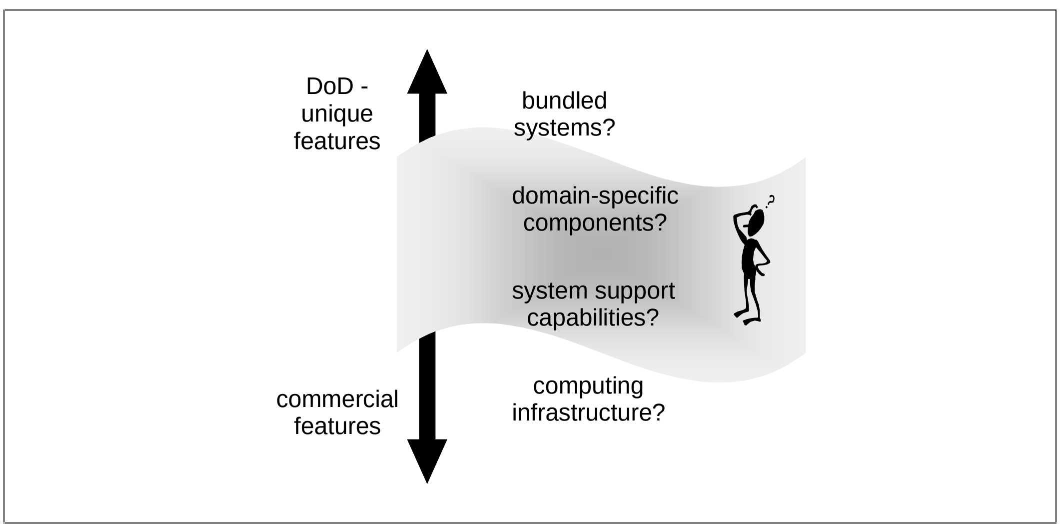

Figure 9: What Should Become a Product Line

As a general case, the more application-specific the product line, the more DoD mission-specific features will be embedded in the products. The opposite holds true for product lines of computer system-level products. These products have features that are universal and commercially available.

The choice of product line affects the options of scenarios that are available. For example, to the extent that product features are DoD-specific, market power becomes more concentrated among a fewer number of purchasers and suppliers. Although the purchaser has greater control over product features, this comes with a higher cost of ownership; market forces are not sufficient to sustain product innovation and to distribute costs over multiple purchasers. On the other hand, the selection of a product line for which there are many commercial implementations may narrow choices to those acquisition and enterprise scenarios that involve many suppliers.

The relationship of market dynamics to the acquisition and enterprise scenarios is illustrated in Figure 10.

Purchasers	many	Cell 2 Proprietary architecture that is nearly a de-facto standard	Cell 1 Scenario 3 open systems architecture
	few	Cell 3 Scenario 1 Competing, proprietary architectures	Cell 4 Scenario 2 Public architecture formed through collaboration and consensus
		few Suppliers	many

Figure 10: Relation of Market Dynamics to Acquisition and Enterprise Scenarios

Because a COTS market is sustained by many purchasers and suppliers in Cell 1, Scenario 3 is an appropriate strategy. Products have standard sets of features, although compatibility with other products is not a primary concern of suppliers. Competition forces rapid product innovation and maturity, so open architectures are important.

In Cell 2, where there are many purchasers and few suppliers, one of the few suppliers usually ends up controlling the market through an architecture. The other suppliers provide products that are compatible with this architecture, so in fact the architecture serves as a de-facto standard. Often the architecture supplier is a trade group such as OMG (object management group). Purchasers may specify specific product requirements, but risk losing the cost leverage of a standard architecture and components.

By specifying unique product features, the purchaser is essentially moving to Cell 3, a niche market with few purchasers and few suppliers. Scenario 1 belongs here. With few suppliers, developing a product line architecture that appeals to a wide supplier base is not needed. Purchasers have the choice of one or two contractors, and these contractors will compete for the few contracts by offering architectures that provide special capabilities. However, because purchasers need to keep the few suppliers in business, contracts are distributed more or less evenly. The competition settles down and long-term relationships are developed. Suppliers are sustained by ongoing modifications and upgrades. More than likely, the same supplier will develop the architecture, the non-commercial assets, and the first few systems in the product line.

Cell 4 is characterized by a few purchasers, many contracts involving considerable sums of money, and many suppliers. Contractors will use a software architecture to block the competition and lock in the customer. However, because purchasers will want to take advantage of the available supplier base, a single contractor will not have control of the architecture. To maximize compatibility with the technologies and capabilities of many suppliers, and to minimize liability of the architecture, the acquisition organization will have the architecture developed as a collaborative effort.

4.4.8 Further Work

The working group did not touch on all the contractor enterprise practices. Topics that may be investigated in this area include the following:

- addressing liability and intellectual property practices and issues in general and for each scenario
- elaborating other aspects of the contract interface such as performance criteria and types of deliverables
- describing how architecture knowledge would be transferred across contractors
- describing the process for evolving and sustaining a software architecture and related skills

5. Summary

In summary, the following guidelines emerged as a net impact of the initial presentations, the participant experience base, and the working group discussions:

- Be certain to carefully scope the product line.
- Provide education and skills development in the area of software architecture.
- Take a realistic approach to domain analysis: beauty is not necessary.
- Be sure to construct and articulate a solid business case for the product line.
- To use legacy assets, create a migration plan.
- Make architecture analysis an explicit and structured step in the process.
- Metrics are critical to long-term success, but ensure that metrics reflect the business goals.
- Adopt a metrics collection plan *before* the project begins.
- Develop and implement a sound acquisition strategy.
- Ensure an appropriate and enduring funding model.
- Develop a product line concept of operations that articulates the supplier and customer interfaces.
- Choose an appropriate and realistic organizational structure that evolves with the maturity of the product line.

6. Conclusion

Reflecting on the practices and issues discussed during the workshop, it became apparent that DoD product lines and industry product lines are more alike than different. In particular, the essentials recapped in the summary (Section 5) for the most part echo what we have heard in the commercial workshops. A similar observation can be made regarding the issues of importance, namely

- *Funding the development of the architecture and other core assets*: Are the funds from taxes on projects that use the assets or are they from corporate investments in research and development? Are the funds part of a strategic initiative or the result of a tactical maneuver by a product line manager?
- *Getting and keeping management's attention on asset development*: What are the business goals? Are there easy measures than can show progress toward these goals? Is product line practice part of a management oversight process? Is there a point person (a manager) responsible for successful implementation?
- *Reorganizing to leverage synergies and reduce complexity in coordination and communication*: Are domain experts in projects available for architecture and asset development? What are the responsibilities of different groups for developing and modifying an architecture, assets, and products? How can the different development groups remain in continuous contact with each other? Are immediate technical and product managers able to reallocate resources and resolve conflicts?
- *Defining which varying features should be supported by an architecture*: To what extent is it effective for an architecture to support unique product requirements? What is the scope of the architecture? How many product variations should be targeted? How is this determined? How is architecture kept current as product features change?

There is *therefore,* a strong reason to continue studying commercial practice and understanding how successful efforts solve these problems. Nonetheless, there are some unique issues for the DoD. The business context, with its emphasis on system acquisition over system development, raises some unique and thorny difficulties. Competitive contracting forces stakeholder relationships and design decisions to become legal. The different business goals of contractors and DoD are often at odds and make leveraging synergies across products more difficult. Additional attention on practices that address the following are necessary for the DoD:

- ownership and liability of non-development items such as the architecture and other assets
- sharing and/or transferring domain and architecture knowledge across contractors

- investing in assets whose benefits are not realized under the current contract

The acquisition context also changes the priority of many issues that are shared with industry. Because architecture and assets are typically developed in-house in commercial industry, companies can proceed with a product line approach without a concrete definition of measures for quality attributes for the architecture and other assets, and the typical costs to develop the assets. However, these are first priority for the DoD. It is difficult to proceed with an acquisition contract without a clear means of estimating costs and assessing quality.

As Dr. Will Tracz in his workshop summary concluded, “The workshop might not have exactly bridged the gap (between commercial best practice and DoD practice), but went a long way to begin to fill the gap.” The SEI was encouraged to continue this gap filling process and hold forums like this workshop. Although the product line practice framework is a work in progress, this DoD workshop (and the previous workshops) reinforces the notion that the basic elements of the framework are sound. Feedback from the workshop is already being incorporated in the framework, and the pointers to more successful product line efforts within the DoD are being studied so that they too can be included in the framework and reported to the community.

References

[Bass 97] Bass, Len; Clements, Paul; Cohen, Sholom; Northrop, Linda; & Withey, James. *Product Line Practice Workshop Report* (CMU/SEI-97-TR-003, ESC-TR-97-003). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1997.

[Bass 98] Bass, Len; Clements, Paul; & Kazman, Rick. *Software Architecture in Practice*. Reading, Massachusetts.: Addison-Wesley Longman, Inc., 1998: 23.

[Bohm 66] Bohm, Charles. "Flow Diagrams, Turing Machines, and Languages With Only Two Formation Rules." Communications of the ACM 8, 5 (May 1966): 366-371

[Brooks 87] Brooks, Frederick, P. “Essence and Accidents of Software Engineering.” Computer 20, 4 (April 1987): 20-29.

[Brownsword 96] Brownsword, Lisa & Clements, Paul. *A Case Study in Successful Product Line Development* (CMU/SEI-96-TR-016, ESC-TR-96-016). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1996.

[Foreman 96] Foreman, J. “Product Line Based Software Development - Significant Results, Future Challenges,” *Proceedings of the Software Technology Conference*. Salt Lake City, Utah, April 23, 1996. Hill Air Force Base, Utah: Software Technology Support Center, 1996.

[Freeman 97] Freeman, Robert. "NCC: Network Component Computing." Distributed Object Computing 1, 5 (June 1977): 41-47.

[Kant 81] Kant, Elaine. "A Semester Course in Software Engineering." *ACM Software Engineering Notes* 6, 4 (January 1981): 24-36.

[Jandourek 96] Jandourek, Emil. “A Model for Platform Development,” *Hewlett-Packard Journal* 47, 4 (August 1996): 56-71.

[Meekel 97] Meekel, J.; Horton, T.; France, R.; Mellone, C.; & Dalvi, S. "From Domain Models to Architecture Framework." *Software Engineering Notes (ACM)* 22, 3 (May 1997): 75-80.

[Withey 96] Withey, James. *Investment Analysis of Software Assets for Product Lines* (CMU/SEI-96-TR-010, ESC-TR-96-010). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1996.

Glossary

application engineering — an engineering process that develops software products from partial solutions or knowledge embodied in software assets

business model — a framework that relates the different forms of a product line approach to an organization's business context and strategy

domain — an area of knowledge or activity characterized by a set of concepts and terminology understood by practitioners in that area

domain analysis — process for capturing and representing information about applications in a domain, specifically common characteristics and reasons for variability

economies of scale — the condition where fewer inputs such as effort and time are needed to produce greater quantities of a single output

economies of scope — the condition where fewer inputs such as effort and time are needed to produce a greater variety of outputs.

Greater business value is achieved by jointly producing different outputs. Producing each output independently fails to leverage commonalities that affect costs. Economies of scope occur when it is less costly to combine two or more products in one production system than to produce them separately.

investment analysis — a process of estimating the value of an investment proposal to an organization.

Investment analysis involves quantifying the costs and benefits of the investment, analyzing the uncertainties, and constructing a spending strategy. This analysis links the strategic and technical merits of an investment to its financial results.

platform — core software asset base that is reused across systems in the product line

product family	a group of systems built from a common set of assets
product line	a group of products sharing a common, managed set of features that satisfy specific needs of a selected market or mission area
product line approach	a system of software production that uses software assets to modify, assemble, instantiate, or generate a line of software products
product line architecture	description of the structural properties for building a group of related systems (i.e. product line), typically the components and their interrelationships. The guidelines about the use of components must capture the means for handling variability discovered in the domain analysis or known to experts. (Also called a reference architecture)
product line system	a member of a product line
production system	a system of people, functions, and assets organized to produce, distribute, and improve a family of products. Two functions included in the system are domain engineering and application engineering.
software architecture	structure or structures of the system, which consists of software components, the externally visible properties of those components, and the relationships among them [Bass 98]
system architectures	software architecture plus execution and development environments
software asset	a description of a partial solution (such as a component or design document) or knowledge (such as a requirements database or test procedures) that engineers use to build or modify software products [Withey 96]

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (LEAVE BLANK)	2. REPORT DATE **May 1998**	3. REPORT TYPE AND DATES COVERED Final
4. TITLE AND SUBTITLE **DoD Product Line Practice Workshop Report**		5. FUNDING NUMBERS C — F19628-95-C-0003
6. AUTHOR(S) **John Bergey, Paul Clements, Sholom Cohen, Pat Donohoe, Larry Jones, Bob Krut, Linda Northrop, Scott Tilley, Dennis Smith, Jim Withey**		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213		8. PERFORMING ORGANIZATION REPORT NUMBER **CMU/SEI-98-TR-007**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) HQ ESC/DIB 5 Eglin Street Hanscom AFB, MA 01731-2116		10. SPONSORING/MONITORING AGENCY REPORT NUMBER **ESC-TR-98-007**
11. SUPPLEMENTARY NOTES		
12.A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS		12.B DISTRIBUTION CODE

13. ABSTRACT (MAXIMUM 200 WORDS)

THE DEPARTMENT OF DEFENSE (DOD) PRODUCT LINE PRACTICE WORKSHOP, *PRODUCT LINES: BRIDGING THE GAP – COMMERCIAL SUCCESS TO DOD PRACTICE* WAS A HANDS-ON MEETING HELD IN MARCH 1998. ITS PURPOSE WAS TO IDENTIFY INDUSTRY-WIDE BEST PRACTICES IN SOFTWARE PRODUCT LINES, TO SHARE DOD PRODUCT LINE EXPERIENCE, TO EXPLORE THE TECHNICAL AND NON-TECHNICAL ISSUES INVOLVED, AND TO DISCUSS WAYS IN WHICH THE CURRENT GAP BETWEEN COMMERCIAL BEST PRACTICE AND DOD PRACTICE CAN BE BRIDGED. THIS REPORT SYNTHESIZES THE WORKSHOP PRESENTATIONS AND DISCUSSIONS THAT DESCRIBED SELECTED PRODUCT LINE PRACTICES AND IDENTIFIED BARRIERS AND ENABLERS TO ACHIEVING THESE PRACTICES WITHIN THE DOD.

14. SUBJECT TERMS **best practices, commercial product line practice, DoD product line practice, product line practice framework, product line workshop, software architecture, software assets, software product lines**			15. NUMBER OF PAGES **79**
			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