

DEVELOPING AN ACQUISITION STRATEGY

The Context: Software Product Line Practice

A product line is a set of products that together address a particular market segment or fulfill a particular mission. Product lines are, of course, nothing new in manufacturing. Airbus builds one, and so do Ford, Dell, and even McDonald's. Each of these companies exploits commonality in different ways. Boeing, for example, developed the 757 and 767 transports in tandem, and the parts lists for these very two different aircraft overlap by about 60%, achieving significant economies of production and maintenance. But software product lines based on interproduct commonality are a relatively new concept that is rapidly emerging as a viable and important software development paradigm. Product flexibility is the anthem of the software marketplace, and product lines fulfill the promise of tailor-made systems built specifically for the needs of particular customers or customer groups. A product line succeeds because the commonalities shared by the software products can be exploited to achieve economies of production. The products are built from common assets in a prescribed way.

Companies are finding that this practice of building sets of related systems from common assets can yield remarkable quantitative improvements in productivity, time to market, product quality, and customer satisfaction. They are finding that a software product line can efficiently satisfy the current hunger for mass customization. Organizations that acquire, as opposed to build, software systems are finding that commissioning a set of related systems as a commonly developed product line yields economies in delivery time, cost, simplified training, and streamlined acquisition.

But along with the gains come risks. Using a product line approach constitutes a new technical strategy for the organization. Organizational and management issues constitute obstacles that are critical to overcome and often add more risk, because they are less obvious. Building a software product line and bringing it to market requires a blend of skillful engineering as well as both technical and organizational management. Acquiring a software product line also requires this same blend of skills to position the using organizations, so they can effectively exploit the commonality of the incoming products, as well as lend sound technical oversight and monitoring to the development effort. These skills are necessary to overcome the pitfalls that may bring failure to an unsophisticated organization.

Introducing Acquisition

Acquisition is the process of obtaining products and services through contracting [Bergey 1999b]. This practice area applies to those organizations that are purchasing or commissioning, rather than developing, at least some of the products, or parts of the products, that they turn out. The growing trend towards outsourcing or "off shoring" makes this practice area much more common than ever before. If your organization has limited resources, you can use acquisition in a variety of ways, from augmenting your own development effort to commissioning the development of an entire product or group of products. Other motivations to use acquisition include being able to

- globally gain access to additional resources on a 24/7 basis to maximize the workforce's availability and shorten the development cycle

- increase competitiveness by leveraging the most economic resources available to perform a specific activity in a timely manner
- take advantage of geographically distributed company resources and their potentially unique perspectives and talents

If you commission an outside source to provide assets for any part of your operation, you will have to incorporate the means of managing the work performed under contract and the resultant contract deliverables. To be effective in contracting, you must develop an acquisition strategy so that you can mitigate the risks associated with acquiring technical products and services from external sources and integrating them into your operations.

Because an acquisition strategy is of great importance to those organizations that primarily acquire rather than develop, this practice area is especially important for government agencies, such as the U.S. Department of Defense (DoD). Although industry may rely heavily on acquisition for obtaining software, corporations are less likely to acquire an entire system. For the DoD, this is closer to the norm. The DoD is an acquisition-based organization.

For acquisition-based organizations, a traditional acquisition approach is to have a separate acquisition for each system and to maintain the system independently throughout its operational life. This is a case of n acquisitions for n system developments followed by n maintenance efforts that may require another multiple-of- n acquisitions (because maintenance contracts may have to be rebid several times for long-lived systems) in order to provide ongoing support for the life of the systems.

Developing an acquisition strategy involves analyzing alternative contracting approaches (some of which are offered as example practices), considering pros and cons, and performing tradeoffs. The strategy should address

- establishing the contracting approach and number of contracts needed to satisfy the acquisition requirements
- choosing the contract types, funding alternatives, contract options, and source selection procedures
- building in provisions to provide tasking flexibility
- establishing whether a request for information (RFI) should be used to narrow the field of offerors who could potentially provide the desired products and services
- determining how the continuity of contractual products and services can best be sustained over the projected life cycle
- establishing an approach for obtaining final acquisition approval

For each contract, the strategy should answer the following questions:

- What should be specified in the request for proposal (RFP), which is the initial solicitation by the acquiring organization to potential contractors?
- What should be included in the statement of work (SOW), which defines the work and work products that will be provided by the successful bidder?

SOFTWARE ENGINEERING INSTITUTE | CARNEGIE MELLON UNIVERSITY

- What technical evaluation criteria should be used to choose among competing bidders and judge the quality of the delivered products?
- What kinds of incentives are appropriate?
- What deliverables should be required?
- What data rights should be incorporated, and, if open source software is involved, how will the data rights to the system be protected?
- What means shall be included to adequately monitor the contractor's technical performance and progress and promote accountability?

Multiple contractors may be involved in acquisitions that incorporate competitive runoffs, multiple suppliers, or teaming arrangements.

An acquisition plan is the artifact used to document the acquisition strategy. The plan should also record the costs, risks, and considered alternatives to the adopted strategy.

Aspects Peculiar to Product Lines

A product line acquisition strategy is a plan of action for achieving a specific product line goal or result through contracting for products and services. In the case of a software product line, the types of software *products* acquired through contracting include core assets and derivative products built from those core assets. Potential software *services* include elements of core asset development, product development, and management. Acquiring services means contractually engaging an identifiable task rather than furnishing an end product.

If your make/buy/mine/commission analysis returned "commission" for any product line asset, you have made a decision to acquire that asset and therefore need an acquisition strategy. Even if you rarely acquire software, you still need an acquisition strategy for those occasions when acquisition is your chosen (or only) option.

Acquisition for product lines is typically structured much differently than acquisition for single systems. First, the acquisition of core assets is usually the result of a contracting effort apart from the acquisition of products. Second, the fundamental role of architecture in a product line imparts opportunities for contracting flexibility. An umbrella acquisition might provide the entire product line production capability—architecture, infrastructure, and core assets—or the production capability might be acquired in pieces. The architecture itself might be acquired as an independent step. The acquisition of software components or services (whose interfaces and interoperating requirements will have been specified by the architecture) might be distributed among several suppliers. Products built from the core assets might be acquired individually or under contract for an entire set. A small number of follow-on acquisitions can accommodate ongoing product development and support for sustaining and enhancing the core assets and products over the life of the product line.

Plan your acquisitions early. The contracting process is often arduous and can consume an enormous amount of time before a contract is in place and operative. Plus, an acquisition strategy cannot be de-

veloped in isolation. It exists to serve the product line goals and interacts with the results of other practice areas (named below), some of which inform us about what we wish to acquire. Accordingly, you will need to allow sufficient time to coordinate and interact with those carrying out the work of these other practice areas.

If acquisition will fulfill a major role in your product line, you should form a small acquisition team to develop and implement an acquisition strategy. Begin by drafting a charter to empower the team, define roles and responsibilities clearly, and ensure the participation of key stakeholders. The team leader should come from the product line organization. Team members should include a contract negotiator, a representative from the product line organization, representatives from participating product groups, individuals with significant acquisition experience, and key product line stakeholders who have a vested interest in the acquisition. Make sure that you

- select motivated individuals who have a "can-do" attitude
- keep the team together through contract award and the start-up of operations to ensure the accountability and continuity of effort
- obtain the early buy-in of those responsible for approving the acquisition plan before committing to a particular strategy

The acquisition strategy team should begin by understanding the organization's concept of how the product line will operate and the role that acquisition will fill. Therefore, the team should coordinate with those carrying out the activities in practice areas in which these issues will be decided. These practice areas include

- "Scoping," in which the product line is defined
- "Funding," in which the strategy for paying for the core assets is determined
- "Architecture Definition," in which a large segment of the necessary core assets is first identified
- "Make/Buy/Mine/Commission Analysis," in which decisions are made about which artifacts are acquired externally
- "Technology Forecasting," which may provide insights about what assets might be acquirable in the future

A product line approach is a natural fit for specifying and coordinating efforts across a distributed or geographically dispersed workforce. Facets of a product line approach that support workforce distribution include

- an overarching concept of operations (CONOPS) that unifies product line activities according to predefined roles and responsibilities and specified practices
- an architecture-driven approach that provides a basis for organizing distributed development
- a generic production plan that prescribes how individual core assets are combined to form derivative products

Application to Core Asset Development

Acquisition can serve several roles in obtaining all, or part, of the core asset base. They include the commissioning of one or more contractors to

- develop the product line architecture
- develop other core assets (for example, components and supporting artifacts)
- mine legacy assets for inclusion in the asset base
- manage, sustain, upgrade, and enhance the existing core asset base and provide support to product developers

Core asset acquisition may also involve the acquisition of contractor services, such as scoping, domain analysis, configuration management (CM), testing, and training. An acquisition organization can fill these core asset development needs and/or acquire these services through a single acquisition or multiple acquisitions involving one or more contracts (or contract options) that run sequentially, run concurrently, or overlap.

If an organization is going to commission the development of core asset software components, an architecture will have to be specified, because the architecture places constraints on the components or services that will be included in the core asset base. The architecture also determines which components are common across all products (or at least across subsets of the product line) and defines the necessary variations among instances of those components. Since the architecture plays such a pivotal role in product line operations and constitutes a strategic competency area, the business impact and necessity of commissioning the development of the product line architecture by an external party should be carefully considered.

Application to Product Development

Acquisition is also an effective means of obtaining new products (or parts of products) in the product line. Acquisition can play several roles, including the commissioning of one or more contractors to

- develop a specific product or a set of new products using the core asset base and following the production plan
- maintain, upgrade, and enhance a product or set of products
- incorporate or evaluate new releases of core assets in a product or set of products to promote product compatibility with the current core asset base and the overall integrity of the product line
- provide new assets (created in the course of product development or product enhancement) to be evaluated as candidates for potential inclusion in the core asset base

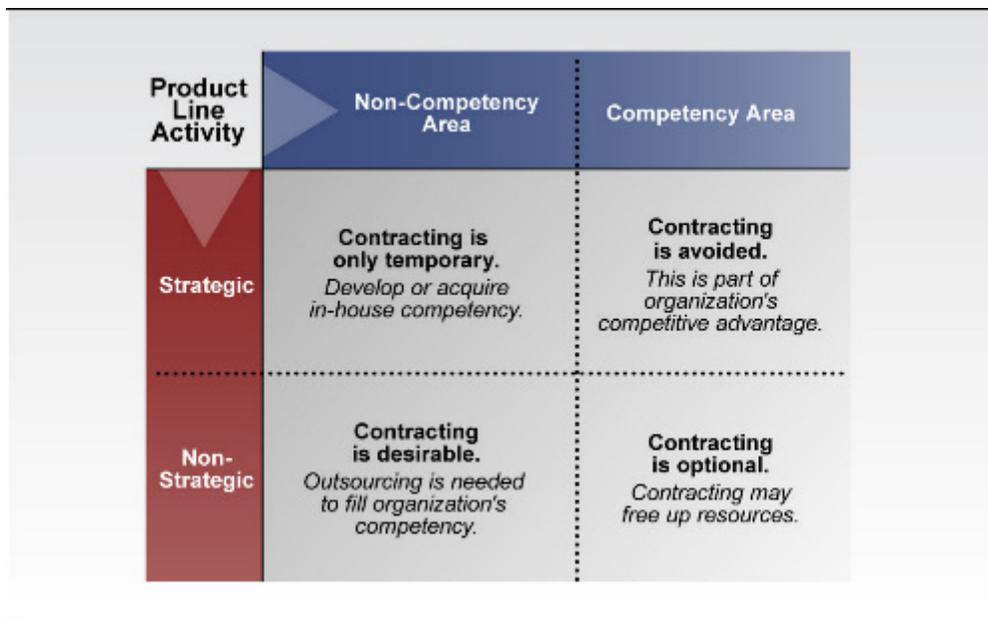
Product acquisition may also involve the acquisition of contractor services, such as requirements engineering, CM, architecture evaluation, software system integration, and testing.

A commercial organization must carefully consider how to protect intellectual property rights when commissioning the development of products from its core asset base. Lack of care could result in the

supplier becoming a competitor. One way to reduce this risk would be to commission the development of product-unique parts or to contract for specific product-creation services. One of the example practices below provides guidance in this area.

Example Practices

Considering core competencies and strategic directions: For commercial organizations in particular, selecting an acquisition strategy should be dependent on what core competencies the organization possesses and the strategic business impact of acquiring products and services from external sources. The criteria embodied in the following figure provide insight into the considerations that should potentially govern whether a product line operation should be included in the acquisition.



Criteria for Outsourcing a Particular Product Line Activity

The considerations for these criteria are

- Certain product line activities may have a strategic impact on an organization's business.
- Every organization has areas of competency and non-competency.
- An organization must be competent in areas that are important to its strategic business interests.
- An organization should avoid contracting out any activity that could negatively impact an organization's strategic business interests.

Clemons and colleagues provide in-depth insight into the issues that inform and motivate such strategic sourcing decisions and their associated risks [Clemons 1997a]. The criteria shown in the above figure are useful for establishing what product line activities are candidates for outsourcing. The selected activities become the basis for developing a comprehensive acquisition strategy that can achieve the desired end state.

Setting the acquisition context with the SEI *Adoption Factory* pattern: Even if an organization relies heavily on acquisition to achieve its goals, it will still have a major role in the execution of many of the product line practices. In setting the context for the product line, the acquisition organization will have to specify what constitutes the product line and establish structures and practices that parallel those of an organization that does most of its work in-house—a key difference being the emphasis on "doing" versus "overseeing the doing."

The "Launching and Institutionalizing" practice area contains a discussion of the Adoption Factory pattern, which provides a general roadmap for product line adoption [Northrop 2004a]. This pattern is a useful guide for helping acquisition organizations determine what they should do first. The following are some of the important early tasks listed in the pattern's Establish Context phase:

- determining what should be in the product line. (See the "Scoping," "Building a Business Case," "Market Analysis," "Understanding Relevant Domains," and "Technology Forecasting" practice areas.)
- establishing product line acquisition processes. (See the "Process Discipline" practice area.)
- preparing the acquisition organization for product line operations. (See the "Launching and Institutionalizing," "Funding," "Structuring the Organization," "Operations," "Organizational Planning," "Customer Interface Management," "Organizational Risk Management," and "Training" practice areas.)

Putting architecture first: One acquisition strategy, which is most applicable to government agencies that rely very heavily on acquisition, involves procuring only an architecture in the first stage, procuring other core assets in the second stage, and procuring products (built from the core asset base) in the third stage. The first stage would then focus on devising a strategy for acquiring the architecture. Listed below are four potential strategies for architecture acquisition that are distinguished by the source of the architecture.¹

1. **systems architect:** A contractor is hired to develop the architecture for the system, but the contractor does not build the system or components. The acquiring organization funds, owns, and controls the product line architecture. Another contractor is hired for the implementation of the architecture. This strategy helps to curtail the risk of getting an architecture that does not fulfill program needs, because the program retains control over the architectural development. However, there is still a risk in terms of whether the architecture will be implemented correctly when the system is built, as well as other risks in terms of cost and schedule adherence.
2. **standards group:** An architecture that is either built by a "standards group" or conforms to established standards is acquired. Industry and/or government collaboration creates a public architecture. The acquiring organization influences but does not control or own the product line architecture.

¹ Fisher, M. CECOM *Architecture for Practitioners Course*. SEI course given in 1998.

3. **single contractor:** A single contractor is commissioned to develop the architecture. The contract is put up for bids, and contractor selection is based only on the architecture portion of the system design. However, the same contractor may also end up developing components and/or the system under the contract. The contractor supplies the architecture and ownership, and control is negotiated.
4. **collaborating contractors:** A contract is developed that requires a group of contractors to collaborate on developing an architecture that they all can use later. In addition, each of the contractors is given a contract to develop and maintain some of the system's components. Ownership of the architecture is usually shared among the development contractors, with the acquiring organization holding the licensing rights. The acquiring organization funds joint development and manages the architecture requirements.

Other acquisition strategies: There are several other acquisition strategies for moving toward a product line capability. They are distinguished by the initial product line capability that is desired. They include the acquisition of

- a software architecture for the product line (discussed above)
- a system architecture (similar to that discussed above)
- an architecture and set of components (and related artifacts) that conform to it
- a product and some set of core assets

The last strategy listed above, which focuses on product development, also results in the acquisition of an architecture, a set of components and their supporting artifacts, and a product built using these core assets. This strategy is an extension of the third strategy listed above, which reduces the risk of architectural and component incompatibilities. The quality of the architecture and components is demonstrated more thoroughly by building an actual system based on the core assets. Also, this strategy aligns with the natural iterative learning that takes place in establishing a product line. Proving that the core assets can be used to build a system provides valuable credibility for them.

A significant variation on this "core assets plus product" strategy is to acquire an additional system—a second product—that will reuse the core assets. This approach allows the program to reap the benefits from the investment in building the core assets.

Practice Risks

Introducing acquisition into the equation clearly provides an organization having limited resources with an effective means of pursuing a product line approach and tapping skills and resources that would otherwise be unavailable. However, acquisition also has its attendant risks by virtue of introducing a new, and sometimes arduous, paradigm for managing the products and services that are acquired through contracting. A poor acquisition strategy will result in contracts being let that are not in the best interests of the acquiring organization. In the worst case, the goods delivered do not meet the needs for which they were acquired, and the resources (not the least of which are the time and effort spent on the contract and the time spent waiting for the goods) are wasted. Deadlines will be missed as the organization scrambles to recover. A poor acquisition strategy can result from any of the following factors:

SOFTWARE ENGINEERING INSTITUTE | CARNEGIE MELLON UNIVERSITY

- **Poor communication and contract oversight:** The more contractors involved, the greater the risk and the need for robust communication and effective contract management.
- **Failure to accommodate iteration:** Iteration is a handmaiden to product line practice. It occurs throughout the operational picture, especially during a product line's start-up and evolution phases. Contracting, however, is not designed to be iterative. Thus, creative means that indicate a prescribed protocol for managing contracts and accepting contract deliverables must be pre-planned and included in contracts. Such means include specifying interim or partial deliverables, conducting technical interchange meetings (TIMs), including in-progress development checkpoints for events such as architecture evaluations, specifying *services* to enable contractor participation in CM control board meetings, software integration efforts, and technical consultations to assist product development groups.
- **Limited management visibility:** Contracting results in a division of management responsibilities that often reduces visibility into the progress and status of the work being performed under contract. Formulating a suitable set of product line measures to monitor the progress of core asset and product development efforts is one means of obtaining the needed visibility. Visibility and insight into the progress of the work and the underlying technical problems are more critical in a product line approach, because they involve more than just an "end-product view" of systems development.
- **Failure to account for evolving requirements:** The nature of a product line is to manage the commonality and variability of products by means of a requirements-engineering-change management process. Contracting, by nature, works best when the requirements are fixed. Again, creative techniques can be employed to accommodate the management of "evolving requirements" and to mitigate the impact on the contractual tasks. Such management and mitigation are especially important in the core asset sustainment and process refinement phases of product line operations.
- **Failure to account for liability:** In a typical single-system acquisition, the contractor is totally responsible for ensuring that the system complies with the contractual requirements. A product line approach often involves multiple contractors, which raises the question of how liability issues involving the efficacy of an architecture and other core assets will be handled. Liability issues can be especially acute if flaws (including documentation errors) are not discovered until much later in product development.
- **Failure to pin down the roles and responsibilities of the acquirer and contractor:** For what things should the acquirer be responsible? For what things should the contractor be responsible? Does the summation of these responsibilities constitute a cohesive approach that covers all the aspects of product line operations? Letting responsibilities fall through the cracks will result in lost time and increased expense that will need to be recovered when the oversight is discovered.
- **Failure to consider ownership and data rights:** If the prescribed data rights are not consistent with the envisioned product line operations and adoptable by the acquirer and contractor, the product line may well be prevented from growing and evolving.

- **Failure to enforce architecture compliance:** What form will the architecture take? Is there a contractual means of verifying that product development is compliant with the prescribed architecture? Failure to ensure architectural compliance in delivered components may result in a set of core assets that will not support a product line.
- **Failure to consider core asset sustainment:** Is there a contractual means of sustaining and evolving the architecture and other core assets? Failure to account for evolution will cause the premature death of the product line.
- **Failure to consider product development support:** Is there a contractual means of providing core asset customization and technical assistance to product developers? If not, the core assets may not be usable in practice.
- **Failure to consider product development:** Is a contractual means in place for projects to commission a product line contractor to develop a product, *or* is there a way for a project's own contractor to obtain core assets and technical assistance for product development?
- **Failure to consider the coupling of contract deliverables:** Is there a contractual means of accommodating product upgrades as changes are incorporated in the core asset base? If not, the product line may produce products that are, in fact, one-of-a-kind systems that drift out of alignment with the core assets.
- **Failure to ensure continuity of support:** Is there a means of ensuring the continuity of acquisition support over the life of the product line?

Tompkins describes 40 major risks when you take the following steps to outsource: strategy, selection, implementation, and management [Tompkins 2004a]. These risks are especially applicable to commercial organizations interested in avoiding the pitfalls and realizing the benefits of outsourcing.

Further Reading

[Bergey 1999b]

Bergey, Fisher, and Jones give a nice overview of the U.S. Department of Defense acquisition environment interpreted for software product lines.

[Bergey 2004a]

Bergey and colleagues provide a companion document to this framework for government organizations that routinely acquire rather than develop their software. This document provides augmented information for acquisition organizations on the essentials of software product lines. In addition, it serves as an interpretation of the framework from a strictly government acquisition perspective.

References

[Bergey 1999b]

Bergey, J.; Fisher, M.; & Jones, L. *The DoD Acquisition Environment and Software Product Lines* (CMU/SEI-99-TN-004, ADA244787). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1999.

[Bergey 2004a]

Bergey, J.; Campbell, G.; Cohen, S.; Fisher, M.; Gallagher, B.; Jones, L.; Northrop, L.; & Soule, A. *Software Product Line Acquisition: A Companion to a Framework for Software Product Line Practice, Version 3.0.* (2004).

[Clemons 1997a]

Clemons, Eric K. & Hitt, Lorin M. *Strategic Sourcing for Services: Assessing the Balance Between Outsourcing and Insourcing.* http://opim.wharton.upenn.edu/~clemons/files/outsourcing_v4_2.pdf (June 1997).

[Northrop 2004a]

Northrop, Linda. *Software Product Line Adoption Roadmap* (CMU/SEI-2004-TR-022, ADA431117). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2004.

[Tompkins 2004a]

Tompkins, J. *40 Risks to Establishing an Outsourcing Relationship.* http://www.tompkinsinc.com/operations/info/40_Outsourcing_Risks.pdf (2004).

Contact Us

Software Engineering Institute
4500 Fifth Avenue, Pittsburgh, PA 15213-2612

Phone: 412/268.5800 | 888.201.4479

Web: www.sei.cmu.edu | www.cert.org

Email: info@sei.cmu.edu

Copyright 2009-2012 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Internal use:* Permission to reproduce this material and to prepare derivative works from this material for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use:* This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

* These restrictions do not apply to U.S. government entities.

DM-0004372