A Framework for Product Line Adoption

Grady H. Campbell, Jr., Prosperity Heights Software

Under particular conditions, a conventional software development process is inherently inefficient. Conventional processes were conceived for the purpose of creating a single solution to a well-defined problem. Such processes are not ideal for the needs of a product line business, to repeatedly build similar products for customers having diverse and changing needs. A process that has been conceived for building a product line provides the means to produce higher quality products faster while reducing redundant effort across those products. This paper describes a framework within which an organization can quickly adopt a product line approach tailored to its needs, capabilities, and preferences.

Background

In 1990, the Software Productivity Consortium began developing the Synthesis methodology[4] based on the concept of product families[5] extended to encompass all the acts and artifacts of a software production. In 1992, it introduced a reuse adoption method[2] to address the impediments to instituting a Synthesis process. In 1993, attempting to formulate Synthesis in terms familiar to its industrial members, it restated the context for Synthesis through two definitions:

Business area: A coherent market of customers having similar needs.

Product line: A collection of products (existing and potential) that address a designated business area.

In 1995, Boeing applied Synthesis as part of a demonstration project for the US Department of Defense[6], providing significant insights into the challenges and strategies for instituting a product line approach.

In 1996, Thomson-CSF initiated a corporate technology transfer initiative for the adoption of Synthesis in four business areas[3]. This initiative applied the SPC reuse adoption method as an adjunct to the SEI Software Capability Maturity Model[1] (SW-CMM).

These and other experiences provide the basis for a disciplined approach to product line adoption.

Repeatedly building similar products

Many organizations conduct de facto product line businesses, repeatedly building custom products as solutions to similar problems. Others do so by creating a single generalized solution to all such problems by repeatedly adding capabilities and options to a single base product. Although the former costs more than the latter, it provides customers with products that are more responsive to their specific needs. The motivation for a product line approach is to give organizations the ability to deliver custom products with a level of effort closer to that of building a single generic product.

Traditionally, a product line business has been operated as a set of independent product development efforts that function by drawing from an available pool of developers who have previously built similar products. This concentration of expertise repeatedly applied to different products can result in good levels of productivity and quality.

Such businesses succeed because experienced developers tend to build similar solutions to similar problems. They know how to accommodate differences without starting over with an entirely new solution. However, an effort may fail or deliver an inferior product if a key developer is not available when needed.

Different people often solve similar problems differently. Describing similar problems differently can also lead to different solutions. However, the natural tendency of an experienced developer on seeing a familiar problem is to derive a variation on a past solution.

Analogously, a product line organization tends to pursue customers having similar needs because it has both the knowledge and expertise required to build responsive products and the practical experience to manage cost and schedule constraints. These qualities give the organization a competitive advantage over less specialized organizations.

The motivation for process improvement efforts is to eliminate recurring causes of quality problems. A secondary motivation is to improve productivity and reduce costs. For software development, improvement efforts usually focus on the institution of organization-wide standard practices. Standard practices reduce differences in how individuals work, including ensuring that needed information about both development efforts and resulting products are recorded and retained.

Formalizing a product line with a domain-specific process makes an organization’s scope of expertise explicit and creates an enhanced capability for building similar products of a particular type.

Reflecting these views, adoption of a product line approach is the means by which an organization can enhance its ability to deliver high quality customized products rapidly at reduced cost and effectively evolve this ability over time to meet changing market needs.

The Nature of a Product Line Approach

Reuse-driven process improvement (PI_r) is a unified model for adopting and improving a product line approach to product development. PI_r is specifically a definition of the reuse adoption activity of the Domain-specific Engineering™ (DsE) methodology process (Figure 1); the purpose of PI_r is to guide the conception and adoption of a DsE process tailored to the needs of a particular product line business.

DsE is a methodology for domain-specific processes[7]. A domain-specific process is a process that is customized

™ Domain-specific Engineering is a trademark of Prosperity Heights Software.

for the streamlined production of solutions to a particular type of problem. The focus of a DsE process is the evolving needs of a market. A domain-specific process delivers higher quality products faster and with less effort than possible with a generic process. However, tailoring requires focused investment in a product line infrastructure that will not support producing solutions for other types of problems. A domain-specific process is viable only when an organization can justify a need to repeatedly produce similar products.

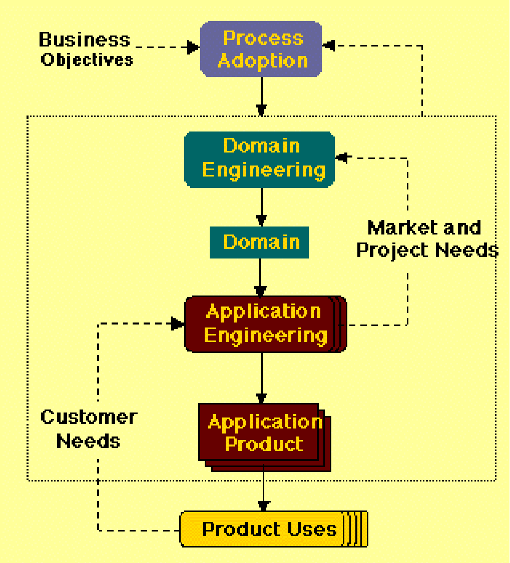

Figure 1. The DsE Process

Reuse-driven Process Improvement

Process improvement in general concerns three concepts:

- *Capability* is the range of expected results that can be achieved following a given process.
- *Performance* is the actual results that an organization achieves following a process.
- *Maturity* is the predictability with which an organization is able to perform at a targeted level of capability.

Conventional approaches to process improvement focus on improving an organization's maturity in performing an identified process.

PI_r focuses more broadly on how a product line business can attain both a higher capability process and greater maturity in following it. Three properties distinguish PI_r from other improvement methods:

- Reuse as a means to improving productivity and product quality is a specially noted driver of the improvement effort. This has significant implications for all aspects of a product line business.
- The scope of an improvement effort is limited to a single product line business organization. Different organizations, even in a single enterprise but building different types of products, may be best served by very different product line approaches.
- Changing the fundamental structure of the software process, based on domain-specific considerations, is a primary means for achieving productivity improvements. A conventional process, with its tendency to isolate product efforts from each other, unavoidably limits the potential benefits of a product line.

The Elements of PI_r

PIr is built on a set of goals, a model supporting each goal, and a process for achieving these goals. The goals of PI_r (Figure 2) interact to clarify an organization’s focus on a viable product line market supported by a tailored DsE process based on sound disciplines for the engineering and manufacture of software-based products.

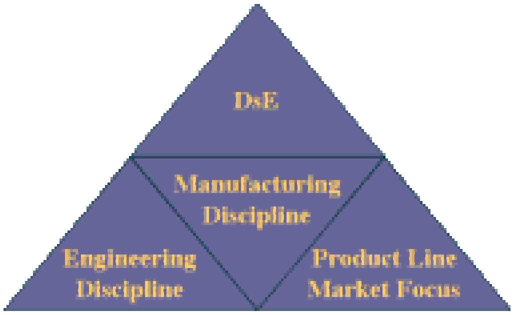

Figure 2. PI$_r$ Goals

Product line market focus is the ability to focus organizational resources on a coherent market needing a set of similar products

Engineering discipline is the ability to predictably produce products having needed capabilities and properties

Manufacturing discipline is the ability to optimize the time, effort, and resources required to produce a product

DsE is the ability to leverage organizational product line knowledge and expertise in the creation of similar products

PIr associates a model (Figure 3) with each of its goals for use as an aid to achieving the goal in a corresponding activity. Models for domain viability, process maturity, and process capability help an organization achieve a consensus awareness of its needs.

- Domain viability concerns whether a product line approach will be viable and effective for an organization.
- Process maturity concerns whether an organization's management and engineering practices can be improved for a more effective product line effort.
- Process capability concerns finding the organization’s best strategy for instituting a product line approach.

A product line strategy model provides the framework for creating a roadmap to how the organization intends to organize and carry out its product line effort.

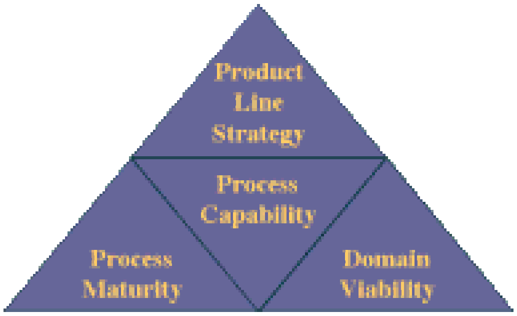

Figure 3. PI$_r$ Tools

Looking at the DsE process from a PIr perspective (Figure 4), the process is organized into activities for addressing each of the goals of PI$_r$.

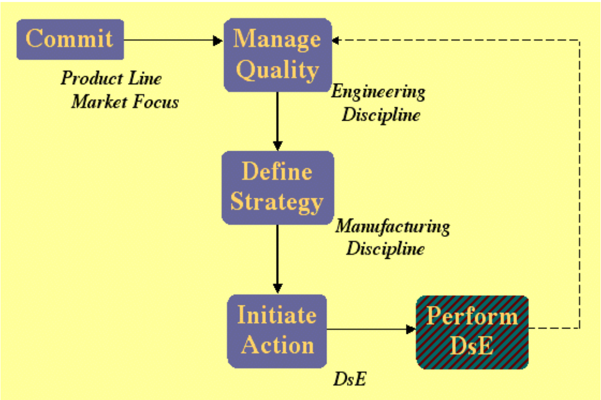

Figure 4. The PI$_r$ Process

The Commit activity is responsible for characterizing a potential product line opportunity and evaluating the benefits and risks of such an approach. Organizational management must commit to a product line adoption effort if they recognize its potential value to the organization. Commitment requires setting business objectives for the envisioned product line effort, allocating resources to the adoption effort, and initiating appropriate monitoring of progress as a basis for subsequently continuing or terminating the effort.

The Manage Quality activity is responsible for identifying and assessing the maturity of an organization's engineering and management practices. The Process Maturity model is used as a guide to identifying and evaluating relevant practices. Needed improvements are identified and submitted for action in the form of proposed action plans. The effects of accepted actions are monitored for future revisions.

The Define Strategy activity is responsible for defining a product line strategy appropriate to the business area. The Process Capability model is used in this activity as a guide to understanding and resolving choices that influence how to tailor the strategy to fit business objectives, technical capabilities, and organizational preferences.

The Initiate Action activity is responsible for implementing the product line strategy. This activity is best performed by the manager and senior staff assigned responsibility for the planned DsE effort. This activity involves obtaining required funding and organizational support, DsE team staffing, documentation and training in DsE practices, establishment of an operational infrastructure, and resolution of any issues that could impede the effort. It also involves prioritizing and allocating resources to implementing process maturity action plans as judged appropriate.

The Perform DsE activity applies the DsE process as tailored to the targeted product line. A DsE process is a comprehensive product line process for domain and application engineering. Domain engineering (Figure 5) is the development and evolution of a domain, represented as a product family and an associated process for deriving instance products. Application engineering projects apply the domain to build customized products for each customer.

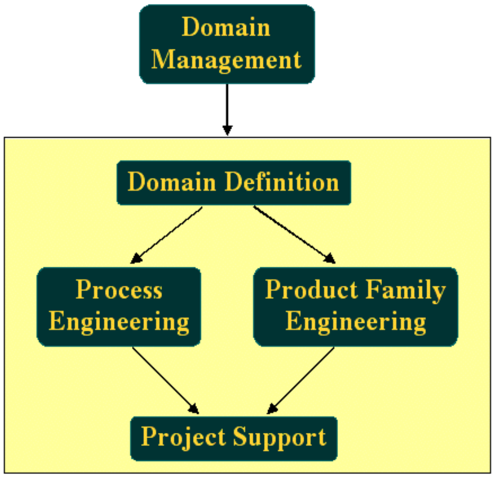

Figure 5. DsE Domain Engineering Process

Domain management organizes, plans, and directs domain engineering efforts.

Domain definition establishes the scope and focus of the domain, including specification based on variability assumptions of a decision model sufficient to distinguish among constructable products.

Product family engineering develops assets and mechanisms for deriving all deliverable and supporting work products for a solution product given a decision model representation of a customer problem.

Process engineering defines a standardized application engineering process and a supporting infrastructure for specifying problems in decision model terms and deriving corresponding solution products.

Project support ensures that the domain meets business, project, and market needs.

Models Supporting PI$_r$ Activities

Domain Viability

Domain viability represents the factors that determine whether a product line approach is suitable for a particular business area organization. The PI$_r$ domain viability model is used in determining whether there is sufficient rationale for a product line approach.

PIr identifies three factors in establishing domain viability:

- Market opportunity – Are there customers for a line of similar products?
- Technical expertise – Does the organization have all of the expertise needed to build envisioned products?
- Business commitment – Is there a credible case and a willingness to invest in this business?

PIr provides a series of detailed questions that give an organization's leaders insight into whether a product line approach is viable from each of these three perspectives. The decision on viability is supported further by a simple experience-based cost model and an emphasis on progressive commitment of effort based on a periodic assessment of actual results.

Process Maturity

Process maturity indicates the degree to which an organization has engineering discipline suitable for a successful product line effort. The PI$_r$ process maturity model guides improving the quality of management and engineering practices applied in the development and use of a product line.

PI$_r$ process maturity has generic and reuse-specific facets. The approach taken for PI$_r$ is to build upon the use of any conventional process improvement method (such as for the SW-CMM). Such methods adequately address the generic factors of management and engineering that determine process maturity. PI$_r$ refines or adds to these factors as needed to encompass reuse-specific aspects of process maturity.

Generic factors focus on six concerns (with reference to key process areas of the SW-CMM as illustration):

Project management
Working within budget and schedule constraints (Software Project Tracking and Oversight, Software Project Planning, Integrated Software Management, Quantitative Process Management)

Engineering methods
Properly performing technical activities (Software Product Engineering, Intergroup Coordination)

Product quality and integrity
Achieving quality goals for products (Software Quality Assurance, Software Configuration Management, Peer Reviews, Software Quality Management, Defect Prevention)

Customer/supplier relationships
: Managing external interactions effectively (Requirements Management, Software Subcontract Management)

Organizational infrastructure
: Establishing effective support for common needs (Organization Process Focus, Training Program, Technology Change Management)

Process predictability
: Reducing variation in results experienced across projects (Organization Process Definition, Process Change Management)

PI$_r$ reuse-specific factors extend these generic factors. Extensions (covering 17 of 52 SW-CMM Key Process Area goals) deal with how well an organization's reuse practices reflect generic concerns. For example, configuration management and quality assurance are at least as important for reusable assets as they are for deliverable work products.

Beyond reuse-extended generic factors, PIr adds reuse-specific factors that focus on three additional concerns specific to product lines:

Product line strategy and management
: Do institutional-level practices support a product line approach? (factors such as costing/pricing and commonality and variability assumptions)

Raw materials and assets
: How well can available materials be used for the product line? (factors such as accessibility and adaptability)

Organizational and technical infrastructure
: Does the infrastructure support product lines? (factors such as legal guidance and tool support)

The emphasis in PI$_r$ is on rapid self-assessment by a product line organization to identify needed improvements. Organizations that have previously instituted a conventional process improvement effort need only revisit past conclusions to ensure that needs unique to the product line have not been distorted by imposition of enterprise-wide standards. They then need to determine that results properly encompass reuse extensions of generic factors and that reuse-specific factors are addressed.

The results of this self-assessment are a set of identified improvement actions to be prioritized and undertaken as appropriate through the Initiate Action activity. Resulting actions generally take the form of training and guidance for product line managers and engineers focused on improved shared practices for the product line business.

Process Capability

Process capability represents a range of choices about how a product line effort could function. The PI$_r$ process capability model guides analysis of issues that determine what sort of product line approach would best fit the market objectives, technical capabilities, and cultural preferences of the adopting organization.

PIr focus on four factors that influence process tailoring:

- Management integration – Should projects operate independently or is coordinated planning an option?
- Needs orientation – Should efforts focus on immediate payoff or long-term advantage? Can there be a unified view of customer needs or

must requirements for each product be unique to each customer?

- Product integration – To what degree must projects focus on the form and content of different work products versus representations of the product as a whole? To what degree can the form of work products be changed for better uniformity across products?
- Stability-optimization – To what degree is the organizational culture amenable to changes in how work is performed?

These are typically neither easy nor familiar questions for an organization to answer. PI$_r$ provides a prototypical set of choices for each that point to standard ways of organizing a product line process which provide increasing levels of capability and associated risk-reward. PI$_r$ identifies four capability levels as alternative targets and provides optionally for incremental transition from lower to higher levels:

- Increase project-level reuse of individual work products within an existing process.
- Collaborate across projects to improve support for similar solutions to all problems.
- Unify and standardize products and streamline the process to address the needs of the targeted market as a coherent whole.
- Coordinate market and domain evolution to achieve mutual synergy in building future products.

A product line organization targets the level of process capability that best matches their market objectives and risk-reward tradeoffs.

Based on analyses using the Process Capability model, the organization develops an appropriate product line strategy.

Product Line Strategy

A product line strategy casts business objectives and organizational preferences into a description of how the business is to be configured and operated. The PI$_r$ product line strategy model guides derivation of a product line approach tailored to the organization’s business objectives and technical capabilities. A product line strategy has six elements:

- Market/products focus – A refined statement, based on prescribed product line objectives, of the market and products targeted by the product line.
- Business model – How product line efforts are affected by legal and financial considerations.
- DsE process – How the process is tailored to reflect a targeted process capability and preferred management and engineering practices.
- Organizational structure – How the organization is structured given the nature of the tailored product line process.
- Support environment – Tools, infrastructure, and legacy assets that will support product line efforts.
- Transition strategy – A plan for directly adopting or incrementally transitioning to any of the four levels of process capability

PIr identifies detailed factors and alternatives for resolving each of these elements to establish an appropriate strategy. This strategy is then taken as the charter, along with identified maturity improvement actions, to guide performance of the Initiate Action activity.

Properly Applying PI_r

A basic premise underlying DsE is that new ways of working are understood, learned, and refined only through repeated practice on real problems. The corresponding premise of PI_r then is that adoption and improvement should be a rapid iterative process. Minimally, assuming key managers and engineers participate without distraction, the initial pass of all PI_r activities up to the point of initiating a DsE effort can be completed in one week. An additional week of training and mentor-guided planning is appropriate for an initial 1-2 month learning increment of the DsE process.

Following the first DsE increment, PI_r activities are repeated to reflect lessons learned and another short increment of DsE is performed. This cycle is repeated until a case is established that the product line approach is or is not going to be effective for the organization. Only after that point are deeper analyses of specific issues such as particular aspects of process maturity justified. Enhanced technical capabilities aside, an important early benefit of this cycle to an organization is significant improvements in business and market understanding and effectiveness by participating managers and engineers alike.

Acknowledgments

Thanks are due to all the past contributors and adopters of the Synthesis methodology, particularly Pascal Maheut of Thomson-CSF.

References

1. M.C. Paulk et al., *The Capability Maturity Model: Guidelines for Improving the Software Process*, Addison-Wesley, Reading, Mass., 1995.
2. *Reuse Adoption Guidebook*, SPC-92051-CMC, Software Productivity Consortium, Herndon, VA, 1993.
3. M. Ezran, M. Morisio, and C. Tully, "Diversity in Reuse Processes", IEEE Software 17, 4 (July/August 2000), pp. 56-63.
4. *Reuse-driven Software Processes Guidebook*, SPC-92019-CMC, Software Productivity Consortium, Herndon, VA, 1993; www.domain-specific.com/RSPgb/.
5. E.W. Dijkstra et al., “On Program Families” in *Structured Programming*, Academic Press, London, 1972, pp. 39-41.
6. R.R. Macala et al., “Managing Domain-Specific, Product-Line Development”, IEEE Software 13, 3 (May 1996), pp. 57-67.
7. G.H. Campbell, “Domain-specific Engineering”, Embedded Systems Conference, 1997.

Grady Campbell is a principal of Prosperity Heights Software (www.domain-specific.com). His interests encompass all aspects of instituting, building, and using product families as a realization of software mass customization. Contact him at Prosperity Heights Software, 8457 Van Court, Annandale, VA 22003; GradyCampbell@domain-specific.com.