

Workshop Summary

10th International Software Process Workshop (ISPW 10):

Process Support of Software Product Lines

Barry Boehm, USC

ISPW Program Chair

(To appear in ISPW 10 Proceedings, IEEE, 1997)

1. Context

The focus of ISPW 10 was on "Process Support of Software Product Lines." Much of the technology currently available to support the software process has focused on the process of developing and evolving a single software product. Increasingly, organizations are finding advantages in product-line software approaches, involving investments in domain engineering, product line architectures, and rapid applications composition with extensive use of commercial-off-the-shelf (COTS) and other reusable software assets. Recent books on software reuse and product line management [Jacobson-Griss-Jonsson, 1997; Poulin, 1997; Reifer, 1997] provide extensive evidence of the advantages: factors of 1.5 to 4 improvements in development time, factors of 1.5 to 6 in productivity, and factors of 2-10 in defect rates.

Software product line management involves significant software process challenges: for example, large-scale software packages (COTS and in-house) often provide so much of an application system's desired functionality that the most effective software approach is for the COTS/reuse capabilities to drive the requirements, rather than

the traditional requirements-to-capabilities process model.

Previous processes and some current process approaches have inhibited reuse. The waterfall model and top-down programming calculus approaches are good examples, as reuse has major bottom-up drivers. Some process maturity models and best-practice initiatives are reuse-insensitive. They are basically best practices for building one-off "stovepipe" systems. Some current process initiatives - EIA/IEEE J-STD-016, ISO/IEC 12207 – accommodate reuse overall, but are still light on guidance about how reuse affects intermediate milestones and key practice elements. Others, such as SPICE, Trillium, and the Systems Engineering Capability Maturity Model, are beginning to fully address and integrate reuse and product line considerations: see Tully's paper in these Proceedings. Overall, however, there is clearly a need for deeper understanding of how reuse and product line considerations affect software processes and software process infrastructure.

The remainder of this Workshop Summary identifies the Workshop objectives and issues to be addressed; the Workshop approach and structure; the working definitions of reuse and product line-oriented terms adopted for the Workshop; the overall Workshop results; and a summary of challenges to the software process field. Additional summaries are provided in the Proceedings for the major subtheme areas of Product Line Implications for Process, Process Element Reuse, and Effects on Current Process Technology.

2. Workshop Objectives and Issue Areas

The primary objective of the Workshop was to achieve deeper levels of understanding of how reuse and product line considerations affect software processes and process infrastructure. The primary focus was on process support of software product lines and of software asset reuse. But another important focus was on reusable process assets and process families. The applications focus was on large-scale industrial applications.

The primary issue areas identified in the Workshop Call for Submissions and addressed during the Workshop were the following:

- Reuse-Sensitive Process Models. How are reuse considerations reflected in models of requirements engineering, architecting, risk-management, development, testing, evolution, and process maturity? What is the effect of reuse-based application generator usage on the above process models?
- Product-Line Oriented Process Models. How are process models affected if the focus is not on a single applications product, but on a product line family of applications? What are the implications for modeling the coordination of multiple concurrent processes for each application in the product line?

- Reuse of Process Elements. To what extent do good techniques for product reuse hold for the reuse of software process elements? Are there similar roles for process domain engineering, process architecture-based reuse, reusable software process asset management and the certification, cataloging, and configuration management of software process assets?
- Effect on Current Process Technology. How do these considerations of process support of software product lines affect process representation languages, process enactment support, process management tools, the role of human beings in the software process, and the selection of representative example problems for the software process community to address?

3. Workshop Approach and Structure

ISPW10 continued the ISPW tradition of a three-day plenary discussion of key issues by 35-40 participants selected on the basis of submitted position papers. Context was established by providing a read-ahead of the selected position papers, a summary of issues to be addressed, and a set of proposed working definitions of reuse and product line-oriented terms. These were assumed to be read by each participant prior to the Workshop; this was generally the case.

In addition, context was established for the focus on large-scale industrial applications via three invited speakers. Nancy Staudenmayer (MIT-Sloan School) presented a comparative analysis of product line-oriented software processes at AT&T/Lucent and Microsoft. Stephane Doublait (Sodalía) summarized process experience with the Sodalía Asset Library Management System (SALMS). Maria Martinez (Motorola) presented the Motorola Cellular Infrastructure Group's experience in developing and applying reusable process assets.

Table 1 presents the Workshop agenda. It reflects the primary emphasis of exploring the process implications of software product line development, including an additional invited speaker, Elisabetta Di Nitto (CEFRIEL) to summarize the results of the first day's discussions. The subsequent sessions addressed software process element

reuse and the effects of software reuse on current process technology, kicked off by an invited presentation by Bob Balzer (USC-ISI).

4. Working Definitions of Reuse and Product Line-Oriented Terms

In order to minimize the inevitable initial Workshop overhead involved in comparing and contrasting participants' definitions of terms, the Program Committee met in advance and developed a set of working definitions of reuse and product line-oriented terms. The following definitions were reviewed in advance by participants and basically adopted for Workshop use.

Software reuse covers any approaches for capitalizing on previously-developed software assets. It has several dimensions:

- The assets may be product or process assets. They may include code, specifications, plans, or procedures.
- The means of capitalization may involve composition of components, customization of generic components, or generation from specifications.

A software product line is a set of assets which supports a family of software products and/or processes within a given application domain.

An application domain is a family of applications with a set of common characteristics. Domains may vary by breadth, as in this example of increasingly narrow domains:

- Transaction processing systems

- Message processing systems

- Medical message processing systems

- Military medical message processing systems.

An architecture is a compatible composition of specifications of system components, connectors, and constraints, along with an associated rationale. The rationale substantiates that a system built to the system's architecture would satisfy the system's requirements, support the system's concept of operation, and be implementable within a given set of resources (e.g., budget and schedule). Architectures may have many dimensions:

- Product or process architecture

- Snapshot or life-cycle architecture

- Individual-system or product-line architecture.

A domain architecture is a product-line architecture for a given application domain.

An enterprise architecture is a set of interface specifications for interoperability of individual system product lines across a given enterprise.

An architectural style is a set of common characteristics of architectural components, connectors, and constraints. It may span multiple domains. Examples are main-subroutine, pipe-and-filter, event-based.

5. Overall Workshop Results

The Workshop participants achieved a strong consensus on five major conclusions:

1. Product line considerations are valuable for both software product and process assets.
2. Software product line considerations require rethinking of single-project oriented processes.
3. Software process solutions are goal-dependent and environment-dependent.
4. Software product strategy insights are a good source of software process strategy insights.
5. Software product line considerations require some extensions of software process technology.

There was much less consensus on each of these conclusions at the beginning of the Workshop. Each conclusion is elaborated below.

5.1 Product line considerations are valuable for both software product and process assets.

For product assets, the experience presentations plus related participant experience made it clear that the payoffs were significant for improving development cost, schedule, and quality, corroborating the evidence in the three books cited in Section 1 above.

For process assets, experience (particularly Maria Martinez' Motorola experience with reusable process "chips") indicated that the quantitative gains were not so large as for product assets, but still appreciable. The major payoffs in cost, schedule, and quality came from reducing project startup effort and process reinvention, more consistent application of processes and procedures, and more effective assimilation of new people. Process reuse could be applied across wider domains than product reuse, but universal process solutions did not work.

5.2 Product line considerations require rethinking of single-project oriented processes.

Product line operations require both different individual-project processes and additional organization-level processes spanning a family of individual projects.

Product line asset use on individual projects requires additional processes for asset search and evaluation, asset tailoring, asset problem handling, and asset assimilation into the delivered product. It changes the overall orientation of the project process from a requirements-to-capabilities orientation to a capabilities-to-requirements orientation. And it imposes additional process elements related to downstream use by other projects of the project's developed product assets. These additional process elements include determining asset reusability

requirements, designing for reuse, and certifying fitness for reuse as well as use.

The additional organization-level processes involve asset-base investment processes: domain scoping and modeling; domain architecting of product-line structure, infrastructure, and interface specifications; and requirements specification for asset generalization. They involve additional asset base management processes: asset catalog and library management; and asset-base workflow and accountability-related processes.

In addition, the inevitable changes in technology, markets, and competition create new evolution-related processes at both the individual-project and organization level. These include domain architecture evolution; generalized asset evolution; evolution of modes of composition (as the domain matures, asset composition and tailoring can evolve toward semiautomated and automated application generation); and synchronization of multi-stakeholder evolution processes (product line marketing, architecting, development, and legacy sustainment).

5.3 Software process solutions are goal-dependent and environment-dependent.

This conclusion is not specific to product-line development and evolution processes, but emerged strongly under consideration of how product-line goals and environments affect the resulting preferred process solutions. Nancy Staudenmeyer's comparison of AT&T/Lucent and Microsoft product line-oriented processes provided particularly strong evidence for this conclusion. Each has evolved different processes tailored to its customer relations, competitive marketplace, and corporate ethos.

In addition, a number of participants' experiences, both involving and not involving product lines, indicated that there was considerable risk in attempting wholesale adoption of process assets obtained elsewhere. The main risks stemmed from differences in goals (development speed, cost, quality, and predictability), corporate cultures, and environment factors (e.g., industry sector producer-consumer relationships).

5.4 Software product strategy insights are a good source of software process strategy insights.

The validity and operational utility of this "software processes are software too" hypothesis have been under considerable debate at process workshops, particularly since the Lee Osterweil-Manny Lehman point-counterpoint keynote session at ICSE 9 [Osterweil, 1987; Lehman, 1987].

At ISPW 10, it was generally agreed that insights and practices found to be useful for software product asset reuse (domain scoping and architecting, asset generalization, asset base management, asset evaluation and tailoring, asset experience feedback and evolution) were a good source of insights likely to be useful for software process asset reuse --under the caveat that the insights be considered suggestive rather than prescriptive.

Given that this agreement and its associated caveat included Osterweil and Lehman as ISPW 10 participants, this was considered a healthy sign that the process field was progressing, in being able to make sharper distinctions about the conditions under which process generalizations may or may not be valid. However, some participants felt that it indicated that the process field was overly mellowing, and that it needed some more good old fashioned controversies to keep its pot bubbling!

5.5 Software product line considerations require some extensions of software process technology.

One source of extensions involves the specification of software processes, particularly capabilities for representing intermediate abstractions relating to product line considerations. Examples are specification of product line objectives (e.g. asset certification and regression testing across the product line); specification of process commonalities and variabilities across the product line (e.g., for international localization); and specification of process interdependencies across the product line (e.g., adapting to product line architecture changes).

Another source of extensions involves related capabilities for process enactment support, such as synchronization and constraint propagation across projects within a product line. Other sources of extensions involve product line-oriented process tool support, process modeling and simulation support, and process management support.

6. Challenges to the Software Process Field

The Workshop wrapup session also included an activity to identify the primary current challenges in the software process field. This was done independently of product line considerations. The resulting primary challenges, not in priority order, were:

- Product line-oriented elaboration of software processes, including intermediate abstractions addressing

- product line objectives, commonalities and variabilities, and interdependencies.
- Product line-oriented process test cases, such as the ISPW6 example for single-project processes.
 - Techniques of deriving best processes from desired product and process goals (cost, schedule, quality, performance, reusability, etc.).
 - Revolutionary process changes (intertwined with revolutionary product changes). These will be necessary for highly ambitious goals such as achieving 10X development time compression while also achieving six-sigma software quality.
 - Decontextualizing good process strategies. For example, Microsoft's nightly-build process should be usefully adaptable elsewhere, but not exactly as practiced in Microsoft's context.
 - Product line-oriented process standards: next steps beyond EIA/IEEE J-STD-016 and ISO/IEC-12207.
 - Product line-oriented process maturity models, extending current efforts with SPICE, Trillium, and the software CMM Version 2.0. Similar product line extensions are also needed for such CMM's as System Acquisition and Integrated Product Development.
 - Integrating process automation and people considerations, particularly for collaborative processes involving multi-stakeholder integrated product (line) teams.

References

[Jacobson-Griss-Jonsson, 1997]. I. Jacobson, M. Griss, and P. Jonsson, Software Reuse: Architecture, Process, and Organization for Business Success, Addison Wesley-Longman, 1997 (to appear).

[Lehman, 1987]. M. Lehman, "Process Models, Process Programs, Programming Support," Proceedings, ICSE 9, IEEE/ACM, 1987, pp. 14-16.

[Osterweil, 1987]. L. Osterweil, "Software Processes Are Software Too," Proceedings, ICSE 9, IEEE/ACM, 1987, pp. 2-13.

[Poulin, 1997]. J. Poulin, Measuring Software Reuse, Addison Wesley, 1997.

[Reifer, 1997]. D. Reifer, Practical Software Reuse, John Wiley & Sons, New York, 1997 (to appear).