Perspectives on Software Product Lines

Report on First International Workshop on Software Product Lines: Economics, Architectures, and Implications
Workshop #15 at 22nd International Conference on Software Engineering (ICSE)

Peter Knauber
Fraunhofer Institute for
Experimental Software Engineering (IESE)
Sauerwiesen 6
D-67661 Kaiserslautern, Germany
(+49) 6301 – 707 242
Peter.Knauber@iese.fhg.de

Giancarlo Succi
Department of Electrical and Computer Engineering
The University of Alberta
238 Civil / Electrical Building
Edmonton, Alberta, Canada, T6G 2G7
(780) 492-7228
Giancarlo.Succi@ee.ualberta.ca

1 INTRODUCTION

Product line engineering is a concept that has emerged in the 80's in the business schools and is now among the hottest topics in software engineering.

Software product lines aim at achieving scope economies through synergetic development of software products. Diverse benefits like cost reduction, decreased time-to-market, and quality improvement can be expected from reuse of domain-specific software assets. But also non-technical benefits can be expected as result of network externalities, product branding, and sharing organizational costs.

Product lines introduce additional complexity. In a sense they go against the common adage of "divide and conquer." Planning and/or developing of more than one product at a time have to be managed technically and organizationally.

However, the rate of innovation of the technology and the intrinsic nature of software products do not let alternatives to developers: users like to jump into the bandwagon of new products, and old products often drive preferences to new products.

Research has been conducted in software product lines for the past few years. Some of it has focused on demonstrating that existing systems and approaches were indeed instrumental for product line development, such as generative techniques, domain analysis and engineering and software components. Another portion of the research effort has tried to determine how it is possible to create a comprehensive methodology and an associated tool for product lines, starting from the business idea of line of products down to the development of a product and trying to exploit all the possible synergies existing at each phase, from network externalities to component reuse.

2 WORKSHOP STRUCTURE

The workshop was structured into the following parts:

- Two invited talks were starting point and introduction into the morning and the afternoon sessions. The first one was given by Stefano De Panfilis who reported about his experience using a product line approach in the domain of fund management. In the second talk the largest European project addressing software families, ESAPS, was presented by its project leader, Frank van der Linden.
  More details on these talks are given in section 3.
- Three technical sessions gave room to present theoretical and practical issues concerning product lines and their use in practice. The first session addressed economic and organizational aspects of product line engineering. In the second session, experience from using product line methods in case studies, experiments, or industrial projects was reported. Several new product line approaches and their specifics were presented in the third technical session. At the end of each session, some time was reserved for discussion.
  The most important topics of each session are briefly summarized on section 4.
- A final panel discussion concluded the workshop. Six panelists answered questions from the audience and discussed with each other. Unfortunately, there was too little time to resolve many open issues.

More information about the panel and the topics discussed there is given in section 5.

3 THE INVITED TALKS

Experience in the domain of fund management
Stefano De Panfilis reported about the application of product line in banking systems, and their specific fund management systems. There is a large common core of functionality among the products that are then customized to the customer's needs. De Panfilis pointed out that this fact is one out of two inevitable prerequisites for successful product line application. According to him, the second prerequisite is, that the customization of products offers a competitive advantage.

The ESAPS project
ESAPS (Engineering Software Architectures, Processes and Platforms for System Families) is Europe's largest research project, coordinating the work of 21 companies and research institutions across Europe. The project manager, Frank van der Linden, reported on the goals as well as the history of ESAPS: some of the project partners have already worked together in two previous projects, ARES and PRAISE. Overall goal of ESAPS is the achievement of significant higher levels of reuse and improved system quality through better engineering of architectures, processes, and platforms for system families. The first phase of ESAPS will concentrate on the development of the approach and laboratory scale validation of the individual technologies and technology integration framework. The second phase will focus on the integration of the individually validated technologies, automation of the approach and industrial scale validation in various domains.

4 THE SESSIONS

Session 1: Economic and organizational aspects of product line development
In this session, the economic aspects of software product lines were addressed: when and where it is worth to introduce a product line approach in an organization, how the benefits and risks involved can be quantified, how the product line can be focused best to match customers' needs, and how the ageing of a product line can be determined in order to decide about its retirement.

The four papers presented here tried to answer these questions and the discussion showed strong interest of the audience.

- The approach presented by *Goiuria Sagarduy* tries to qualtify the potential benefits and risks involved in the decision to go towards a product line development in order to give company owners and managers a good idea about the economic impacts of such a decision.
- In his talk, *Stuart Faulk* presented a new software development process including Customer Value Analysis to link relevant software design decisions to tactical and strategic business objectives. Customer Value Analysis here is used to denote a product's perceived oberall benefit.
- *Klaus Schmid* structured his product line scoping approach in three steps: the first one (product line mapping) produces a high-level dscription of the product line in terms of domains; the second one (domain-based scoping) does basically an assessment of reuse potential and viability of these domains; this information is then used to select the most promising ones. The third step (feature-based scoping) produces the quantitative benefit of implementing certain features in a generic way, that is, reusable. These three talks were about product line introducing and running them most effectively.
- In the last talk, *Susanne Johnsson* discussed the evolution of product lines, how the costs of maintenance increase over time and the relative division of effects resulting from maintenance tasks. These considerations are the basis of a model for identifying architecture erosion which can be used in the decision processes surrounding the reorganization and retirement of software product lines.

Session 2: Case studies, experiments, reports from industrial projects
In this session, industrial projects, experiments, and case studies together with the respective approaches used, their results, as well as lessons learned were presented. The intention was to help other organizations that intend to invest in product line development to get a feeling for the main risk factors and the critical issues to consider. The very lively discussion after this session was continued in the final panel discussion.

- *Ali Mili* reported about a classroom experiment, where four domain engineering methodologies (FODA, JODA, Synthesis, Reuse Business Methodology) were analyzed and compared. Criteria for the comparison were mainly the support for the domain engineering lifecycle, the rationale for domain definition, the support for legacy assets, the guidelines for domain architecture development, the domain engineering deliverables, the reusable assets, the technology and language dependency, and the effort for domain and application engineering.
- *Oliver Lewis* proposed a set of experiments that would help to quantify the space and time overhead due to variability in a product line implementation versus a single system implementation. The resulting data can inform embedded systems engineers about the

behaviour of overhead they might expect in a single system solution built from a product line model.

- In his talk, *Dan Paulish* described his experience with applying the technique of global analysis to plan software projects better by designing product line architectures that anticipate change. The purpose of global analysis is to analyze the factors that influence the architecture and to develop strategies for accommodating these factors in the architecture design. The technique was applied to the design of a meter data processing and control central station platform. The resulting high-level design proved to be very flexible and expandable.
- *William El Kaim* reported on product line experience from an experiment concerning systems used in the simulation for ground vehicle pilot training. Strong emphasis was on the modeling of the architecture from different perspectives (or views), each perspective corresponding to the concern of a stakeholder. The experiment, which was performed in the PRAISE Esprit project, showed also the importance of traceability among assets and the need to convince the people involved to explicitly describe and reuse assets. Tools, even though considered very important, are very expensive while not yet providing the support needed for product line development.
- In his talk, *Paolo Predonzani* described a case study regarding the introduction of domain analysis and object-oriented frameworks in a small software firm with the purpose to set up a development environment based on product lines. Goal of this project was to evaluate the impact and the benefits of the introduction of a domain engineering approach in a specific domain, laying the groundwork for the definition of a corporate reuse program toward the introduction of a product line. The experiment showed satisfactory results in general, the new concepts had positive effects on the software process.

Session 3: New product line approaches

In this session, new concepts for the development of product lines and the management of their evolution over time are presented. These include addressing single aspects or the complete product line life cycle. Specific attention has also been posed in tools to support product lines, government initiatives and small and medium size companies, which constitute a large and significant part of the software market.

- *Paul Clements* detailed a framework for the establishment of software product line practices. He identified the following key areas to target before the introduction of software product lines: domain understanding, asset mining, architecture exploration and definition, architecture evaluation, COTS utilization, software systems integration, data collection, metrics, and practices, product line scooping, configuration management, technical risk management. In addition he emphasized the importance of a suitable launch of the product line initiative in a company.
- *William El Kaim* presented a work of *Michel Coriat* and *Frédéric Waeber*. He introduced Wheels, a process framework to introduce and institutionalize product line practices. Three are the tenets of Wheels: **(a)** the use of UML for its metamodeling; this is intended to increase the understandability of the metamodel; **(b)** the adoption of a matrix approach in defining those practices that are required in a company to implement a product line approach; **(c)** the definition of a written handbook to detail the process patterns of the company. Wheels is being experimented in real projects by Alcatel and Thomson-CSF.
- *Giancarlo Succi* discussed the essential requirements for the establishment of a tool to support product lines. The key problem is the need to track and integrate the multiple activities that are required for the sound design of a product line. There is also the need to support domain specific advice. In brief, these requirements are: **(a)** link consistency management, to ensure that link traces make sense; **(b)** queries on dependency links: filtering, sorting, and other relationships; **(c)** change consistency management between different activities, to ensure that all changes are propagated correctly; **(d)** management of multiple users working concurrently; **(e)** data integration with COTS tools: to allow COTS tools to communicate with the framework; **(f)** semantic domain specific assistance. He also detailed the relevance of a design critiquing system.
- *Jorge Díaz-Herrera* presented a methodology for establishing product lines in the domain of embedded systems, a part of the Yamacraw Embedded Systems (YES) program, funded by the Georgia government. A detailed description of the project can be found in its web site: http://www.yamacraw.org. The idea is to exploit a synergistic design of multiple embedded systems together. Reuse is expected to play a major role, and a major promoter of reuse is the definition of standardized interfaces for the different pieces of hardware devices. The overall product line effort is divided in two groups of activities: **(1)** Modeling activities, including *(1.1)* requirement engineering for embedded systems, and *(1.2)* smart compilers for embedded systems; and **(2)** Engineering activities, including *(2.1)* personal embedded computing environments, *(2.2)* networked and enterprise embedded applications, and *(2.3)* home computing applications.

- *Joachim Bayer* raised the problem of how to help small and medium size companies to implement software product lines. He suggested that a product line methodology for such kind of companies should be able to **(a)** cope with immaturity of the development environment; **(b)** introduce techniques and concepts step by step, to evidence clearly the progress; **(c)** continue the work-in-progress, to avoid any disruption of the development, which would not be bearable in a small environment; **(d)** focus on the evolution of the products and not the variants, since SMEs are more likely to have new releases of products, rather than different versions of the same product; **(e)** rely on existing techniques and tools, for which there might be already expertise in the company: SMEs are not likely to take risks associated with new ideas. He introduced KobrA, a product line methodology specifically targeted to SMEs.
- *David Rine* evidenced that in a product line there are three major stakeholders: a management, system developers, and a reuse team. They all need to have coherent views of the product line, however, such views may be significantly different. Typical views for software product lines are the "product line overview," the "product line architecture," the "products," the "product release architecture," the "components." Each view is characterized by its own attributes. These views can be provided for the major steps involved in developing a product line: **(a)** deciding on the adoption of a product line strategy; **(b)** planning the product line; **(c)** utilization and management of a product line; **(d)** expansion of a product line.

5 THE PANEL

A panel concluded the workshop. The panelists were J. Bayer, P. Clements, J. Díaz-Herrera, D. Rine, and W. El Kaim.

The panelists and the workshop participants discussed several topics. Here below there is a review of the discussion. We have organized it by few major topics: the costs to establish a product line, the CMM, problems related to small companies, the role of management in a product line initiative, the importance of domain analysis and variability analysis in a product line effort, staging models for introducing a product line, how product lines support competitive biddings for large government contracts, and the relationships between product line efforts and lightweight methodologies. Clearly, there are significant overlaps among these topics.

Costs. Clearly, the overall cost to develop a product line depends on the size, the kind, the peculiarities etc of the target domain. Everything depends on how the costs are defined and measured, and this is not a trivial task.

Whenever a domain is scoped for a product line initiative there is an economic rationale of optimality. This then results in all the subsequent decisions, including the tradeoffs between generality and specific implementation. Studying such optimality is key in determining the overall cost/benefit analysis of the product line effort. This is not purely an economic analysis but also involves other issues related to management. For instances, there are problems related to legal aspects. Small businesses often have to conform to additional legal requirements to receive targeted business supports.

Simple numeric answers based on immediate cash flow have been proposed in the past for the CelsiusTech and the Boeing experience; the old adage has been repeated: "the cost of doing a product line is 2 to 2.5 the cost of doing one product the old way."

A non-monetary cost is represented by the time to market. For the introduction of a product of a brand new line, a product line approach may result is longer time-to-market, with risk of failure of the marketing effort. However, once the product line is institutionalized, it is faster to come up with new products in the line. An approach consists of starting small, incrementally, possibly growing from existing products.

A further approach to reduce the time to market is to create a statewide infrastructure, like what the state of Georgia is doing with the Yamacraw project.

There are also circumstances when time to market is not the key consideration. There are companies who take a break from production to grow in size. In these cases, a product line strategy is a way to provide a reasoned growth, with a clear definition of the core of the company and all the additions.

CMM. There is no one-syllabus answer on the relationships between the maturity in the CMM scale and the feasibility of a product line development. It seems that maturity helps but it is not a prerequisite. A speaker suggested that the CMM level 3 should be required for the business unit.

Anyway, it is evident that product lines practices and process improvement –in whatever scale it is measured, go hand-in-hand. This is especially true dealing with frameworks-based product lines.

The situation is particularly critical for **small companies.** Small companies represent a large part of the software development market; for instance, 80% of the companies in the Washington DC area are small companies.

There are indeed differences on how large and small companies can approach a product line initiative. A survey by Rine and Nada, to be published shortly, will detail these differences.

One of the problems is getting to the point where small companies leave a service oriented approach and start to see a substantial profit from a line of products, perhaps in synergy with other small companies. Also, the focus of product lines is not just reusing components, but to share knowledge across products, which may be even a more critical issue, since turnover in personnel is harder to manage in small companies.

Small companies alone often cannot afford to undertake certifications, such as the CMM certification. However, often in small companies the amount of variability for new requirements is usually narrow and they relay on a single product. Once the product is shipped, then it is possible to look at further requirements and expand the market.

As previously mentioned, Fraunhofer IESE has developed KobrA, a methodology to introduce product lines in small and medium size companies.

Management. The role of management in a product line effort is essential, especially when there are common assets, shared across departments.

In small companies this is not usually an issue. The management group is often part of the development as well, so there is not an "independent, non-technical management" to convince.

In large companies, the situation is different. For an effective product line strategy, everyone in the command line needs to be involved, and everyone in the command line, from the top down to the bottom, makes key decision.

This is a difficult issue, because there are several units to involve. A possible approach is to focus on the technology viability of the product line initiative first and then, if proven suitable, a limited pilot can be launched. If the pilot is proven successful, then the scope of the product line initiative is broadened. This is iterated to involve always larger parts of the company, till everything suitable is inside a product line.

The SEI framework requires a heavy management support: 2/3 of it is management oriented. Management has to fund, take the risks, put incentives. In addition, management has to participate to issues related to aging of software systems and controlling the evolution of the product line, since such issues are critical for the business successes and costs of the line.

It is also important to notice that customers may be exposed to tradeoffs of product lines and decide accordingly. The future of the company owning the product lines can be prosperous, since they have a baseline to compete on the market. Product lines can be considered a way to build the future.

Domain analysis and variability analysis. It is not yet clear whether an upfront effort in domain analysis is the requirement for a successful introduction of a line of software products. This is especially important for small companies, which often to don't have the resources for such initial commitment.

Variability analysis is an important part of the establishment of a product line. Variability analysis is scattered across multiple phases of the process of developing a product line. First, variability is studied while understanding the relevant domain for the line of products and while scoping the domain. Variability also lives in architectures.

Two critical issues deal with the situation of embedded systems, when variability is to be solved also with hardware components, and with instilling the knowledge of variability in the corporate knowledge base.

Anyway, a very comprehensive marketing analysis is a crucial prerequisite for a successful analysis of variability.

Standard staging models for introducing a product lines. At the Software Engineering Institute there is ongoing research on staging the introduction of product lines. The idea is to start where most of the benefits occur. A risk assessment is performed at the beginning and then a set of steps is identified, with ample room for improvements and modification.

Competitive bidding for large government contracts. Competitive biddings occur very early in the software lifecycle, when very limited information on the target system is available. Product lines are a big advantage in these situations. Business people can make decisions with more information: **(a)** they can reuse previous instances of the line –analogy can be performed to a much larger extent, and **(b)** they have already byproducts that can be used in the final system.

Lightweight methodologies. Lightweight methodologies are not antithetic to a product line. There are parts of the product line strategy that live well with, say, extreme programming. An example is domain scoping. The development methodology depends on the situation, the business environment, the company skills, etc.

6 LESSON LEARNT AND LINES FOR FUTURE RESEARCH

The workshop has consolidates some aspects of the state of the art on software product lines.

- *Methodologies* have been classified, reviewed, and experimented.
- The pivotal role of *staging approaches* and of *champions* for the introduction of product lines has been reaffirmed.

Several new ideas have been presented, that set the lines for future research in the field.

- There is a need of understanding better how to shape software product lines for *small software companies*. Research has already been performed. Hoiwever, there is not a well established understanding of the issues. More models and more experiments are required, as the ongoing effort at Fraunhofer IESE on the KobrA methodology.
- The *importance of a product line approach beyond the simple reuse* of software components should be more clearly stated and defined. Suitable supporting tools should be developed.
- The *role of government agencies and initiatives* could be critical in establishing state- or country-wide framework that could support product line initiative of local companies or even product lines that span multiple local companies, as is the case for the state of Georgia.
- Clear taxonomies and experimentations of *economic models* for product lines should be developed, to provide companies more precise figures of what they can expect from product lines and what should be their upfront investments.
- *Relationships between product lines*, corporate environments, and other methodologies, process improvement frameworks, and tools should be clarified, to better understand when and how it is appropriate to start a product line. This is in particular important for the ISO and the CMM certifications and for extreme programming and other lightweight methodologies, given their relevance in the software industry.

7 CONCLUSION

Altogether, the workshop has been a very large success, due to the quality of the submitted papers, the level of participation of the audience, and the profile of the panelist. Several positive feedbacks have been received; for this reason, we have decided to publish the papers as a collection in [1].

At ICSE 2001 in Toronto the “Second International Workshop on Software Product Lines: Economics, Architectures, and Implications” will be held. We look forward a lot of papers and participants, to discuss the advancement in the discipline brought by the Y2K and to exchange our experience.

REFERENCES

[1] Software Product Lines: Economics, Architectures, and Implications, Peter Knauber, Giancarlo Succi (Editors), Proceedings of Workshop #15 at 22nd International Conference on Software Engineering (ICSE), Limerick, 2000, Fraunhofer IESE Report No. 070.00/E, 2000