

## **PLEES'01: International Workshop on Product Line Engineering – The Early Steps: Planning, Modeling, and Managing**

Klaus Schmid  
Fraunhofer IESE  
Sauerwiesen 6  
D-67661 Kaiserslautern  
schmid@iese.fhg.de

Birgit Geppert  
Avaya Labs Research  
Basking Ridge, NJ USA  
bgeppert@research.avayalabs.com

In the last five years product line (PL) Engineering has become a major topic in industrial software engineering. It introduces a focus-shift from the development of single systems to the development of complete system families. This paradigm shift aims at the efficient and cost-effective development through large-scale reuse by exploiting the family members' commonalities and by controlling their variabilities. Reported results are indeed encouraging. These include error and effort reductions by a magnitude as well as strong time-to-market improvements by one third [5, 6, 7, 8].

The research focus until now has been on the technical, implementation-centered activities related to family-based development [1]. However, in order to be successful these activities need to be performed in an organizational and technical framework that is appropriately prepared to support them. This raises organizational issues, issues of PL planning, as well as issues concerning requirements modeling and management in the context of product lines. These more up-stream activities have not yet seen as much attention, but they are nevertheless critical for the successful introduction of a product line approach [2, 3].

In this workshop we focus on the issues that are particularly important when introducing a product line approach in an industrial environment. This discussion is centered around the following four key issues:

- Transition Approach
- Organizational Issues
- Product Line Management
- Variability Modeling

### **Transition Approach**

When transitioning a running organization into a product line engineering organization, several issues must be considered that range from bridging technological gaps to solving people issues. Existing development units may not be organized to allow product line engineering, but follow an independent work style with design decisions made on a product by product basis. This leads to the duplication of features that are developed by different units and for different products. An organizational re-structuring is necessary that takes into account a joint development of the common product line infrastructure and an accompanying re-assignment of responsibilities.

In most cases this transition can't be done in one step. Rather, a product line engineering approach must be introduced incrementally. This raises issues of

determining the best starting point in terms of products, functional areas, and organizational units. An important success factor in this transition is to convince and motivate people to follow the new organizational structure and accept the new assignment of responsibilities.

## **Organizational Issues**

A key factor of successful product line engineering is an appropriate organizational structure of the software development organization. Several different organizational alternatives are possible, ranging from one single development unit that is responsible for developing and maintaining the product line infrastructure as well as deriving the single products, to a hierarchical structuring of the development units with the tasks for developing and maintaining the infrastructure as well as the single products distributed over several units [9, 10].

Identifying the organizational structure that is best suited for a given situation is a complex task that is only partially understood yet. There are many different factors that influence this decision process. Examples are the existing organizational structure, the potential for people to experience direct benefit from reuse, the strategic positioning of the product line in the market, or the size of the organization. But also technical aspects do impact the decision. For instance, the organizational structure needs to match the software architecture of the product line so that tasks and responsibilities can be assigned appropriately to the different development units. This is actually a requirement on both, the organizational structure as well as the software architecture. Consequently, both need to co-evolve.

So far, identifying and establishing the right organizational structure is not sufficiently understood. Thus, we see at this point in particular the need to further study existing and working product line organizations.

## **Product Line Management**

When turning its attention to product line development, an organization faces important challenges that go all the way from the initial planning stage, through the early development, to the continuous evolution of the product line.

During planning the appropriate alignment with the overall product portfolio of the company and its long-term strategy needs to be established. It is already at this point that key decisions are made that will determine the overall economic benefit of the PL to the company. These decisions need to be performed in an integrated manner on several levels. First of all a commitment needs to be made on the specific systems that will be developed as part of the product line. From a strategic point of view, leaving inappropriate products out can be a key benefit to the organization, just like finding the right systems to include. Here, product line development directly interfaces with strategic management of the company. This interface needs to be appropriately managed to ensure success. Similarly, once the products have been identified the key assets that need to be developed for reuse must be identified. Again, identifying the right assets is key to the economic success of the PL. This kind of bounding decisions are in this extent unique to product line development and are addressed under the heading of scoping [4].

Once we identified the initial development plan for the product line, we need to put it into practice. This is non-trivial, as the various projects that are part of the product line need to be developed in an integrated manner as the shared asset base creates

important dependencies among the projects. Sustaining these links is crucial as otherwise the survival of the product line is at risk! For example, a disconnect between the development of the platform and the individual systems risks the overall economic benefits of product line development.

Finally, a PL needs to evolve over time. New products need to be integrated into the product line and old systems need to be phased out. This has to be supported by the development of appropriate reusable assets. Also the shift in the product line needs to be aligned with the overall strategy of the company. These topics have so far hardly been addressed. Product line evolution complicates in particular change management, as for any change decisions have to be made, about affected systems, actually changed systems, and so on.

## **Variability Modeling**

A key principle of product line development is the codification and reuse of knowledge. In the end this boils down to the reuse of code components that are generically reusable throughout the product line. However, in order to enable and sustain this generic code development, the product line perspective needs to be pervasive throughout all artifacts across all life cycle steps. Be it requirements, design, test cases, we need to be able to describe relevancy to the product line as a whole as opposed to single systems. As not everything will be relevant to every system to the same extent, we need to make explicit the commonalities *and the variabilities* of the artifacts. While this is probably at this point the most well-studied topic within the scope of the workshop, it is still more of an art form than a daily practice in industrial environments.

At this point we are still lacking general approaches that can be applied to all forms of artifacts and that also support the efficient instantiation for product-specific situations. But even more important: how can we elicit the information that is required as a modeling basis in an efficient manner? We may explicitly not restrict ourselves to a system-specific focus as otherwise the resulting assets will be too narrow, but we may also not take on an everything-is-important attitude as this would lead to a waste of resources and an information overload that may hinder subsequent activities. Thus, a major question is: how do we focus beyond a system?

## **Summary**

In this workshop we focus on the early steps in product line engineering, i.e., those steps that have not yet seen as much attention as the implementation specific tasks, but that are nevertheless critical for a successful product line engineering project. This comprises the transitioning of a running organization into a product line organization, issues of how to organize a development organization to best support product line engineering, tasks of managing a product line which ranges from the initial planning to an ongoing evolution, and finally the identification of a product line's reuse potential and its packaging into a product line architecture and accompanying reuse components. The intent of this workshop is to enable a discussion around the four aforementioned topics among practitioners and researchers in order to allow an exchange of industrial experience and academic results.

The proceedings of this workshop will be available as Technical Report from Fraunhofer IESE at <http://www.iese.fhg.de> or directly from the workshop organizers.

## References

1. Software Product Lines: Experience and Research Directions; Proceedings of the First Software Product Line Conference (SPLC1). Patrick Donohoe, editor. Kluwer Academic Publishers, 2000.
2. J. Bayer et al. PuLSE: A methodology to develop software product lines. In Proceedings of the ACM SIGSOFT Symposium on Software Reusability, pages 122-131, 1999.
3. Paul Clements and Linda Northrop. A framework for software product line practice - version 3.0. <http://www.sei.cmu.edu/plp/framework.html>. Software Engineering Institute, Carnegie Mellon University, 2000.
4. Klaus Schmid. Scoping software product lines - an analysis of an emerging technology. In Patrick Donohoe, editor, Software Product Lines: Experience and Research Directions; Proceedings of the First Software Product Line Conference (SPLC1), pages 513-532. Kluwer Academic Publishers, 2000.
5. David M. Weiss and Chi Tau Robert Lai. Software Product-Line Engineering: A Family-Based Software Development Process, Addison-Wesley, 1999
6. James C. Dager. Cummin's Experience in Developing a Software Product Line Architecture for Real-Time Embedded Diesel Engine Controls. In: Software Product Lines - Experience and Research Directions, Proceedings of the First Software Product Lines Conference (SPLC1). Ed. Patrick Donohoe, pp. 23-46, Kluwer Academic Publishers, 2000.
7. Peter Toft, Derek Coleman, and Joni Ohta. A cooperative model for cross-divisional product development for a software product line. In Patrick Donohoe, editor, Software Product Lines: Experience and Research Directions; Proceedings of the First Software Product Line Conference (SPLC1), pages 111-132. Kluwer Academic Publishers, 2000.
8. Paul Clements, Cristina Gacek, Peter Knauber, and Klaus Schmid. Successful software product line development in a small organization. In Paul Clements and Linda Northrop, editors, Software Product Lines: Practices and Patterns, chapter 11. Addison Wesley Longman, 2001.
9. Jan Bosch. Software product lines: Organizational alternatives. In Proceedings of the 23rd International Conference on Software Engineering, pages 91-100. IEEE Computer Society Press, 2001.
10. Klaus Schmid. People issues in developing software product lines. In Proceedings of the Second ICSE Workshop on Software Product Lines: Economics, Architectures, and Implications, 2001.