Transfer of Product Line Technology into Industry

Dirk Muthig

Fraunhofer Institute for Experimental Software Engineering
Sauerwiesen 6, D- 67661 Kaiserslautern, Germany
Email: muthig@iese.fhg.de

Abstract The motivation for almost every technology transfer project is an organization's problem or goal. Software reuse is a discipline that is a promising candidate for improvement activities in case of many problems software organizations typically have. Nevertheless software reuse (and product line engineering) is only one of many fields in which an organization can improve. The theory and experiences in transfering product line engineering into industry are presented at the conference.
This paper describes the starting point of the presentation: it motivates product line engineering, briefly describes its history, and introduces PuLSE™ (Product Line Software Engineering)[1].

1 Motivation

The initial motivation for almost every technology transfer project is an organization's problem or goal (the distinction between problem and goal depends on whether you are an optimist or a pessimist). The technology to be transferred is supposed to help solving the problem and achieving the organization's goal. As usual, there are several, alternative ways to do so, that is, improvements in different fields or disciplines can eventually achieve identical goals. If there is a best way at all, it surely depends at least on diverse characteristics of an organization, its current situation, the concrete problem, and the stated goal.
Software reuse is a discipline that is a promising candidate for improvement activities in many cases. Nevertheless software reuse is only one of many fields in which an organization can improve. Product line engineering again is only one of many possible orientations towards software reuse. Consequently, product line engineering must compete with alternative reuse approaches and also with approaches from other fields before a transfer project, for example, focusing on processes, tools, or planning. Unfortunately, most of the product line engineering methods ignore this competition by describing only the final, ideal state of a product line organization. They typically do not provide guidance on how an organization can or should transition into this final state in a systematic and controlled way.
Without this guidance, the introduction of product line engineering into an organization is more like a risky adventure than a controlled improvement project.

[1] PuLSE is a registered trademark of the Fraunhofer Insitute for Experimental Software Engineering.

2 Product Line Engineering

Product line engineering is the systematic development of a set of similar software systems by understanding and controlling their common and distinguishing characteristics. Historically, product line engineering is an approach towards software reuse. In a basic reuse model, an organization develops product by product and stores the assets the products consists of in an asset base with the intent to reuse these assets during subsequent projects. However, as many organizations experienced, such a simple, straightforward approach usually does not achieve the expected improvements in reuse. The simple approach typically does not define means for organizing and managing reusable assets in a way that effectively supports projects.
In domain engineering, the reuse challenges are tackled with the concepts of a real-world domain, in which the software products are built.
Domain engineering pioneered the idea of planning and developing similar systems, systems in the same application domain, concurrently. The domain, its abstract concepts, entities, and relationships are analyzed and used to built a reference model for systems in the domain including domain-specific reusable assets. Concrete applications could then be constructed mainly by reusing the domain-specific assets, which represent the domain concepts or features required for the concrete application.
Although domain engineering represents a significant step forward in the support of reuse, practical experience has shown that the definition of the domain under consideration is problematic. When the domain is chosen to be too small, the domain model and the reference architecture fail to address important issues. As a result, significant changes will typically be necessary in future projects, undermining the value of the whole domain engineering effort. On the other hand, when the chosen domain is too big, the effort invested in domain engineering is higher than is really necessary, and the already significant investment involved in domain engineering goes beyond that which is cost effective. In short, although domain engineering provides a powerful set of solutions to the fundamental reuse problem identified previously, the success of a domain engineering effort is highly sensitive to the correct scoping of the domain. If a domain is chosen to be too big or small, domain engineering can do more harm than good.
Product line engineering solves this problem by using the characteristics of a set of concrete (existing, planned, or future) products to scope the domain. Everything required by a concrete product is part of the domain - everything else is outside.

3 PuLSE

PuLSE is a method for enabling the conception and deployment of software product lines within a large variety of enterprise contexts. This is achieved via a product-centric focus throughout its phases, customizability of its components, an incremental introduction capability, a maturity scale for structured evolution, and adaptions to a few main product development situations.

.

PuLSE is centered around three main elements: the deployment phases, the technical components, and the support components. The deployment phases are logical stages of the product line life cycle. The deployment phases are:

- PuLSE initialization: PuLSE is customized to the context of its application. The principle dimensions of adaption are the nature of the domain, the project structure, the organizational context, and the reuse aims. The initialization phase is realized by the technical component for customizing, PuLSE-BC.
- Product line infrastructure construction: The product line infrastructure is set up. This is done by scoping, modeling, and architecting the product line. These activities are realized by the corresponding technical components PuLSE-Eco, PuLSE-CDA, and PuLSE-DSSA, respectively.
- Product line infrastructure usage: The product line infrastructure is used to create a single product line member. This is done by instantiating the product line model and architecture. The PuLSE-I technical component realizes this phase.
- Product line infrastructure evolution: Concepts within the domain or other requirements on the product line may change over time. The evolution of the product line is handled in this phase. The process for controling evolution is realized by the PuLSE-EM technical component.

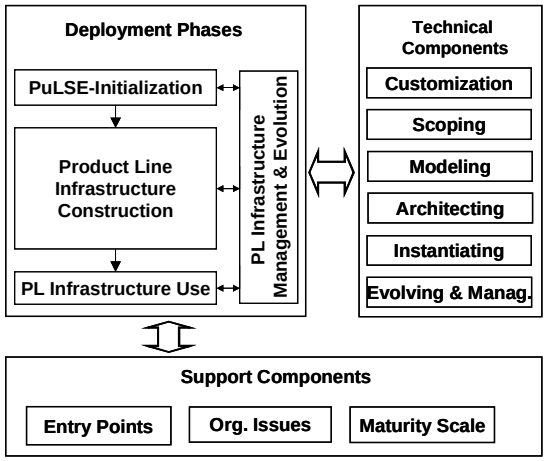


Fig. 1. overview on the PuLSE framework

3 Links

For more information visit PuLSE on the web at http://www.iese.fhg.de/PuLSE
or the bibliography at http://www.iese.fhg.de/Domain_Engineering