A Process Family Approach for the reuse of development processes

Maik Thränert and Andrej Werner

Abstract— Development processes are often newly defined for every project. However, a reuse of the process knowledge between different projects rarely takes place. In this paper, we present a concept which permits a general reuse of process knowledge on the basis of the process family approach and as well the project individual customization of processes according to a mass customization.

Index Terms— Organizational management and coordination, Software Engineering Process, Development process reference model, Process model, Mass customization, Process family and Process factory

1 Introduction and Problem Definition

Development projects within those developments are passed through are carried out in many software development enterprises. These development processes are based on known and documented procedures. They are, however, different from project to project, caused by the character of the relative uniqueness of a project.

These development processes can be distinguished in core processes, supporting processes and leadership processes. The real creation of value takes place within the core processes. This creation of value is designed depending on the business, the project, the customers and the product. Supporting processes – e. g. invoicing and reporting – are relatively standardized over the projects. Leadership processes are for control and decision – e. g. in the field of project and multi-project management – and are organized generally for all projects.

The real core processes – as for instance software development, service development or software-service co-design – offer a high potential for the improvement of efficiency. This improvement can be reached during the preparation – in other words, the design of the development process – and the execution of the project. Just like in production processes so-called "operation centre systems" can be used for supporting the development. Such systems allow a process based project control and supervision.

Mass customization is a synthesis of mass production and the production of individual customer products. In this expression, the contrasting concepts of "mass production" and "customization" are combined. With the application of certain technologies and innovative organizational structures, the production of variant-rich and often even customer individual products and services shall be possible at absolutely high product quality and short delivery times to rates of the mass production [1]. The most important aim of the mass customization is the planning and the production of goods and services with that amount of variability, that each customer can find exactly the solution requested by him and the products the he is financially interested in.

In the software sector, reuse and individualization have been playing an important role during the last time. Examples are component development, product line approaches and others. On the other hand, reuse and project-specific customization of process knowledge is inadequately carried out. This leads to a high customization effort during the design of the processes for new projects. The whole development process is frequently completely new developed based on the procedure model, although the projects to be carried out have many common features.

A concept for the adaptation of development processes on the base of individual project features is introduced in this article. The aim is a "mass customization" according to an individual provision of development processes with a high rate of reuse of documented knowledge about development processes. The efficient management of the development process components is therefore required.

First, basic conceptualities are explained in the context of procedure and process models. Then we represent how processes can be defined based on procedure models. The concept of the family based adoption of the procedure models is explained after the exposition of the basic ideas of the system family approach. An example and an outlook will conclude this paper.

2 Basics

2.1 Procedure and Process Models

The explanation of the following concepts is orientated towards the appreciation of the group WI-VM of the German Society of Computer Science, as it is documented e. g. in [2]. Figure 1 shows a summary of the concepts and their relations used here.

The development processes are carried out within the development projects. They are important at the runtime

- *Maik Thränert is with Institute for Computer Science, University of Leipzig, Augustusplatz 10-11, 04109 Leipzig, Germany, E-mail: thraenert@informatik.uni-leipzig.de.*
- *Andrej Werner is with the Information Systems Institute, University of Leipzig, Marschnerstraße 31, 04109 Leipzig, Germany, E-mail: a.werner@wifa.uni-leipzig.de.*

September 30th, 2006.

T. Sobh (ed.), Innovations and Advanced Techniques in Computer and Information Sciences and Engineering, 309–313.

time of the project. These development processes – and the project itself too – are part of the life cycle of the complete product, e. g. the software life cycle.

The development processes can be represented in a

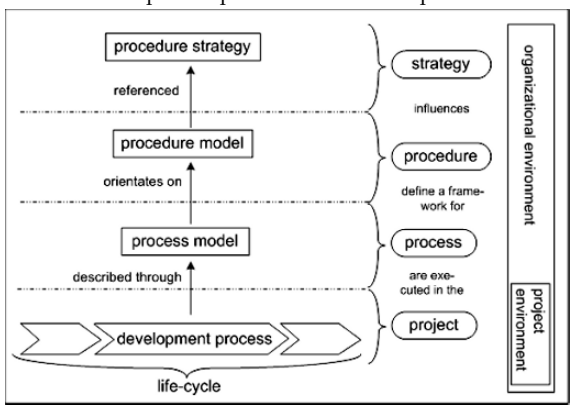

Fig. 1. Basics concepts.

process model. This process model is the result of the process design for the current development project. It depends on the features of this project and is therefore only suitable for this specific project. The process model is consequently also part of the project plan.

A procedure model is the reference model for the development processes. It describes in an abstract way in which states the system to develop can be. A procedure model is the base for the process model, which is formed depending on the specific project features.

The procedure model is influenced by the basic procedure strategies. Such a procedure strategy describes a vision or an idea about the development of the application [3, p. 33]. Version oriented or risk oriented approach are examples of procedure strategies.

Variants for the derivation of process models

To derive process models from procedure models, different ways are conceivable as represented in the following.

Variant 1: A generic procedure model from the literature which seems to fit to the project environment is adapted to the conditions of the organisation and the concrete project requirements. A specific procedure model is designed for the organisation and the projects on that base.

This specific procedure model is then refined to a project-specific process model. This procedure reoccurs correspondingly at every project. It is obvious that the reuse is carried out only on the high abstraction level of the generic procedure model. The same or similar customizations are performed repeatedly, see figure 2.

Variant 2: A reference procedure model, suitable for the

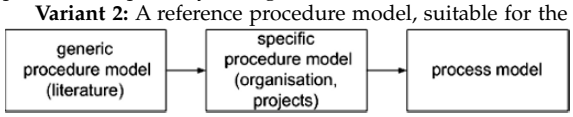

Fig. 2. Variant 1.

organisation and the project type and based on a generic procedure model, is selected. This reference model is adapted to organization-specific and project-specific conditions. If a reference process is also part of the reference model, then this can be adapted for the concrete project (customization). If not, a process model is made based on the adapted reference model. The reference model (procedure and/or process model) can be extended by organization-specific or project-specific attributes and therefore serve as a template for further projects.

The level of reuse in this variant is higher than in the first one, because specific features are already documented in the reference models, see figure 3.

Variant 3: Here the reference procedure model is the

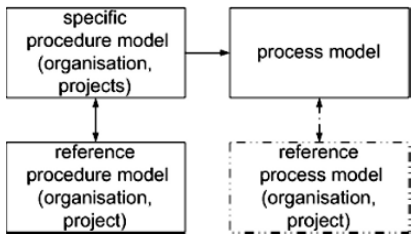

Fig. 3. Variant 2.

foundation for the adaptation. This reference procedure model must be suited for the organisation and project-specific conditions. If a reference process is available, then this is adapted correspondingly (customization). If not, the reference procedure model must be used as starting point of the customizations. This way is the same compared to Variant 2.

The project features which influence the customization of the process model are now identified. The common and different features are documented in a feature model. With every project, the knowledge therefore increases. The probability increases that certain development elements (so called process components which are different from the reference model) can be applied for further development processes. It is prerequisite that features of a process component are related to features of a project.

With this approach, the development process does not have to be described again and again from scratch. Common and varying process components are reused: a configuration of the process model. The result is a development process which was configured on the basis of already existing process components. Process components that do not have a fitting template till now are created and added to the knowledge base. So they are able to be used again in the future.

The highest reuse of process components of a procedure model is achieved with the variant 3. The costs for the individualization of a development process are the lowest. By reusing of the process knowledge, the quality and the maturity also increase. This aggregated knowledge can also be changed.

This variant is introduced in detail in chapter 3. Basic ideas of system families are explained in the following part.