CHALLENGES IN SOFTWARE DESIGN IN LARGE CORPORATIONS
A Case Study at Siemens AG

Peter Killisperger[1,2], Markus Stumptner[2]
[1]*Competence Center Information Systems, University of Applied Sciences - München, Germany*
[2]*Advanced Computing Research Centre, University of South Australia, Adelaide, Australia*

Georg Peters
Department of Computer Science and Mathematics, University of Applied Sciences - München, Germany

Thomas Stückl
System and Software Processes, Siemens Corporate Technology, München, Germany

Keywords: Software, Process, Instantiation, Tailoring, Application.

Abstract: Successful software development is still a challenge and improvements in software processes and their application is an active research domain. Siemens has started research projects aiming to improve software process related activities. The adaptation of generic software process models to project specific context and the application of instantiated processes play a major role in these efforts. Several solutions have been put forward in literature in recent years to better standardise and automate these procedures. However, no approach has become a de facto standard. Instantiation and application of processes in practice are still often done manually and not standardised. On the basis of an analysis of Siemens' current practice, a New Software Engineering Framework (NSEF) is suggested for improving the instantiation and the application of software processes within the company. In particular, the paper suggests the development of a gradual instantiation approach.

1 INTRODUCTION

In order to improve software project practice, a number of software process models have been suggested in literature and tested in practice. According to ISO 15504 (ISO/IEC, 1998) a software process is "the process or a set of processes used by an organisation or project to plan, manage, execute, monitor, control and improve its software related activities". Many organisations choose one established model (e.g. the V-Model (BMI, 2004)) as their internal standard or even develop and define their own process.

Today's software development projects are highly individual and in order to apply processes to a spectrum of projects they are described in a generic way. To use them in a particular project they have to be adapted to the individual needs of a project in turn. This instantiation comprises tailoring of the process and resource allocation. The former is "the act of adjusting the definitions and/or of particularising the terms of a general process description to derive a new process applicable to an alternative (and probably less general) environment" (Ginsberg and Quinn, 1995). The latter is the assignment of resources to activities to carry them out (Aalst and Hee, 2004). Recent studies show that this is still a wide open field for research (Pedreira et al., 2007).

Siemens' software developing business units define their processes according to their needs within a general corporate wide process framework. The application of processes in projects, including the instantiation for individual projects, is rarely standardised and thus differs widely within Siemens. Siemens Corporate Technology, the University of South Australia and the University of Applied Sciences - München are collaborating to improve Siemens' current practice and to develop a general approach.

The goal of this paper is to introduce to Siemens' Software Processes, identify shortcomings of the current practice of instantiation and application and to

Killisperger P., Stumptner M., Peters G. and Stückl T. (2008).
CHALLENGES IN SOFTWARE DESIGN IN LARGE CORPORATIONS - A Case Study at Siemens AG.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems - ISAS*, pages 123-128
DOI: 10.5220/0001683301230128

provide an approach to overcome these issues. Our work follows the long stipulated research in real-life situations (Boehm, 1976)(Zmud, 1997).

The paper is structured as follows. Section 2 introduces into Siemens' Software Processes by describing Siemens' practice in modelling and applying software processes and shortcomings of the current practice are revealed. In Section 3 a New Software Engineering Framework (NSEF) is proposed aiming to improve the current practice and to identify design issues. Relevant existing approaches are analysed in section 4.

2 SIEMENS' SOFTWARE PROCESSES

Siemens' software developing business units model their software processes within a company-wide framework according to their individual needs. The processes have several levels with the higher levels being more abstract. Figure 1 shows a simplified example of a low level process modelled as an Event-Driven-Process-Chain (EPC) (Scheer, 2000). The process is started when the state "Implementation decided (Component)" in the sub process "Implement Product (Subsystem)" is active. After reaching the state "Component Implemented", work will continue in the sub process "Implement Product (Subsystem)" as indicated in Figure 1. Sub processes can be executed several times within a project, even in parallel, and dependencies can exist between them. For every activity, a Function-Allocation-Diagram (FAD) (Scheer, 2000) is defined. It contains a description of the activity, the roles involved, input and output documents and links to further information such as document templates.

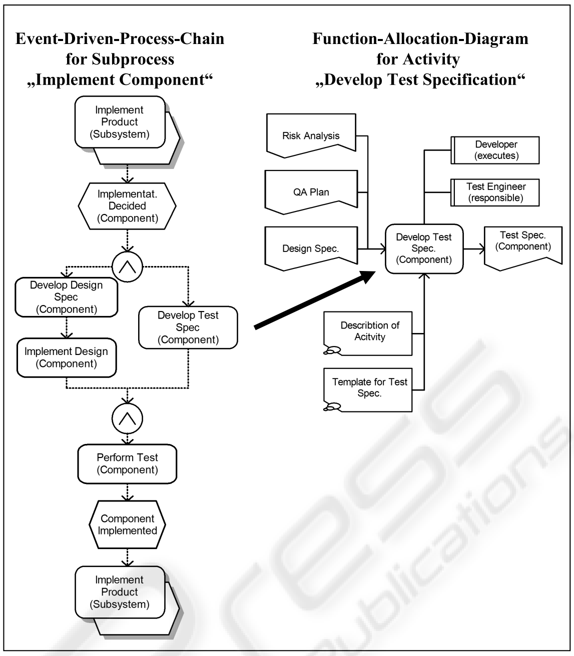

Figure 1: Example of a Low Level Process.

2.1 Current Instantiation and Application

A reference process of a particular Siemens organisation can comprise several thousand elements. It functions as a guideline for any software project within that organisation, but still cannot be applied in projects without modifications. Instead, it is intended to be a sample or framework from which the actual process, in the form of project specific plans such as project plan, quality assurance plan or test plan, is derived.

In addition, the reference process provides a knowledge base for project management and members. In order to use the reference process, project members have to interpret its information for the project at hand. A project specific process visualisation is not available.

The purpose of the established reference processes is to be used by humans and not to be supported or automated by any kind of information technology. Therefore they are designed to be easily read and understood by humans.

2.2 Issues of Current Approach

There is essentially no standardisation of the manual transformation process from the reference process to project-specific plans, or for the way in which reference processes are currently applied. Siemens has identified the following significant disadvantages of their current approach:

- Since the adaptation process for creating the project-specific plans is not standardised, the adaptation and the resulting plans depend largely on the experience and preference of the performing expert. This leads to non-standardised plans and can implicate poor process adherence.
- Manual adaptation is time consuming and expensive.
- Due to missing a visualisation of the instantiated software process, interpretations of the reference process by project members are necessary. From this follows that the perception and application of

the process varies widely. This can reduce effectiveness and efficiency of software projects.

- Missing support of information technology in the application of processes can result in cumbersome and inefficient project management.
- Decisions made during setup of the plans and interpretations of the reference process by project members are not documented and thus unreproducible.
- Due to missing standardisation, improvement of the reference process as well as the adaptation process is difficult to achieve.

Standardisation and support by information technology are desired for the instantiation of software processes and their application in projects.

3 NEW SOFTWARE ENGINEERING FRAMEWORK

A survey with practitioners, namely project members and consultants, was conducted at Siemens, in order to find out how processes can be better applied in daily work and how the issues described above can be addressed.

In our project, we asked them how (from their point of view) the application of processes in projects can be better supported and how the reference process has to be transformed to support their desired application of processes.

Because of the complex domain and diverse perceptions of ”perfect support in application of processes”, we decided to use unstructured interviews (Fontana and Frey, 1994) for questioning of individuals and groups. They provide a good means to understand complex context without the limitation of a predefined categorisation (Fontana and Frey, 1994) by providing some structure through predefined areas to be covered (Simmons, 2002).

On the basis of the information collected in the interviews a New Software Engineering Framework (NSEF) for improving the instantiation and application of processes at Siemens is proposed (see Figure 2).

Starting with a reference process, first a high level instantiation is conducted, followed by a gradual detailed instantiation. The result is a project specific process which is implemented as process visualisation, project plan, workflow support and management of project documents. The arrows leading back to the instantiation indicate the iterative character of our approach. Iterative instantiation is proposed because it

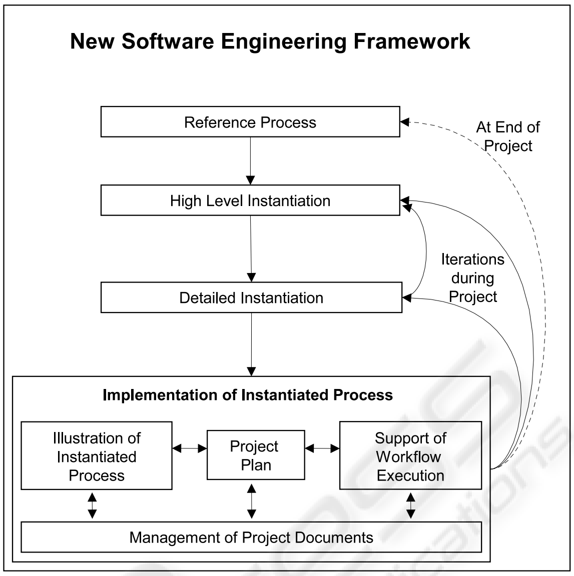

Figure 2: NSEF - New Software Engineering Framework.

is hardly possible to completely define a project specific process already at the start of a project (Becker et al., 1997). A main reason is the volatile nature of software development. It may also be necessary to adjust the process during projects to achieve optimal performance (Williams and Cockburn, 2003). The arrow leading from the implementation to the reference process symbolises the continuous improvement of the reference process on the basis of lessons learned. In the paragraphs below the components of the NSEF are examined in more detail.

3.1 Reference Process

A reference process of Siemens will be used as the main input for the instantiation stages of the NSEF. Processes are modelled by using EPCs, extended by associated FADs (Scheer, 2000).

3.2 Instantiation

High Level Instantiation. High level instantiation is a first step towards a project specific software process. It bases on project characteristics and information that can already be defined at the start of a project and are unlikely to change. Such characteristics can be e.g. the size of a project (a small project will only use a subset of the process) or required high reliability of the software product which requires certain activities to be added to the process. In other words, high level instantiation is a first adaptation of the structure of the process by e.g. cutting away unneeded parts, the selection of pathways, the addition of objects or the duplication of sub processes. The latter is required if e.g. the system to be developed is split up into the development of subsystems. For every subsystem, instances of sub processes are required. High Level Instantiation can be rerun during the project in case characteristics of the project change severely. Because the adaptation decisions are predefined, they can be approved by Quality Assurance (QA) when they are defined and can be applied without any further approval. Tool support for their execution includes maintenance of the consistency of the resulting process.

Detailed Instantiation. Detailed instantiation is run frequently during the project for the upcoming process activities. It comprises operations which cannot be decided at the beginning of the project. The number of activities which can be instantiated in detail is restricted by the time span for which accurate estimations of the future can be made and has to be adjusted if unforeseen changes occur. Detailed instantiation includes detailed resource allocation including the assignment of persons to activities. Instantiation decisions have to be supported by tools, documented and approved by QA. Required tool support includes ensuring consistency of the resulting process due to the complexity and dependencies in processes.

3.3 Implementation of Instantiated Process

As already mentioned, the instantiated process is implemented by a process visualisation, project plan, workflow support and management of project documents. These components depend highly on each other. Interaction can be accomplished by an integrated solution or by a framework of tools. A major requirement for an implementation is ensuring the consistency of data.

Visualisation of the Instantiated Process. The visualisation has to offer all information required by project members for executing activities. This includes a description of activities, project members involved in execution, links to documents and their status as well as links to further information. The current status of the project has to be tracked and displayed in terms of executed activities which enables monitoring of progress and easy navigation to the next activities to be executed. Links to the project documents, project plan and to the workflow support have to be provided.

Basic Project Plan. A project plan including activities and associated work products has to be generated on the basis of the instantiated process. Links from the project plan to the corresponding project documents, activities in the process visualization and workflow support have to be provided.

Support for Workflow Execution. Support can be implemented by workflow-management-systems (WFMS) for particular sub processes. They are beneficial when several people are involved in a process and the responsibility of execution of activities is transferred frequently between them. These kinds of sub processes are generally executed several times during a project and often triggered by events. Examples are processes for defect and change management, creation of specifications or reviews for milestones, documents or quality gates. Users are informed what they have to do and when. Links from the WFMS to project documents, project plan and process visualisation have to be available.

Management of Project Documents. The management of documents has to provide additional features in comparison to common document management systems. This includes manual access to project documents which is necessary in e.g. follow-up projects where a great deal of documents can be adopted from the previous project.

3.4 Design Issues

The framework above describes the target situation. Below we detail and reason which parts of the framework have to be addressed.

Reference Process. The current form of the given reference process has to be maintained and major changes have to be avoided due to Siemens's corporate policy and in order to enable portability of the instantiation approach.

Instantiation. Concepts implementing the stipulated high level and detailed instantiation have to be developed and tool support implemented. Although instantiation in our approach is split up in two distinct stages, it is advantageous if both base upon the same principles. The idea is to define high level instantiation as an aggregation of detailed instantiation (i.e. it is batch processing of basic operations that are executed individually in detailed instantiation). The only difference is that process adaptations in high level instantiation are predefined, depend on the existence of particular project characteristics and are only executed at the beginning of a project or when major project characteristics change. The basic instantiation operations comprise all operations which are necessary to adapt a process to project specific needs. Examples are the deletion of activities, the selection of pathways, the duplication of sub processes or the association of process documents with files implementing them. The approach of using the same principles for both instantiation stages improves flexibility. If for instance a particular adaptation operation is not to be executed in high level instantiation any more, the operation can be easily shifted to detailed instantiation or vice versa. Because of the inherent complexity, tool support for executing these basic operations is essential. For instance the deletion of an activity has an impact on other activities if an artefact is created in the activity to be deleted and used in others.

Implementation of the Instantiated Process. Commercial implementations supporting process visualisation, project plan, automation of workflows and management of project documents exist. Selection of tools depends highly on personal preferences of users and organisational policies. Therefore it is not reasonable to develop implementations of the instantiated process but rather let users and organisations choose tools fulfilling their personal requirements.

In conclusion, our work focuses on the development of a two-staged instantiation process fulfilling requirements derived from interviews with practitioners as described in section 3.2. Both instantiation stages base upon the same principles to allow flexible handling of operations. Input for the instantiation approach is a Siemens Reference Process in its current form as described in section 2 and 3.1. The implementation of the instantiated process is only considered if necessary for developing the instantiation approach. Desired output of the instantiation is a process with its structure adapted to project needs and containing all information required for its application. It can be used for the generation of a process visualisation, a project plan, workflow support and for management of project documents.

4 RELATED WORK

Research in the field of instantiation and application of software processes in projects is not new. In early software process approaches it was thought that a perfect process can be developed which fits all software developing organisations and all types of projects (Boehm and Belz, 1990). In the eighties and early nineties it was already recognised that no such process exists (Osterweil, 1987) (Basili and Rombach, 1991). Instead, every project has individual needs which have to be accommodated. Therefore, the description of the general solution (i.e. the software process model) has to be adapted in order to be applicable to individual problems. Early approaches to tackle this issue followed in subsequent years. For instance, Boehm and Belz (1990) used the Spiral Model to develop project specific software processes. Alexander and Davis (1991) described 20 criteria upon which the best suiting process model for a project can be selected. Since then many different adaptation approaches have been proposed and the need for adaptation of processes is recognised in industry shown by publications about tailoring approaches in practice. For instance, Bowers et al. (2002) describe a case of project-specific adaptation of the agile software process XP. Fitzgerald et al. (2000) detail tailoring a generic process derived from the IEEE 1074 software standard (IEEE, 1992) and from the V software lifecycle model (Sommerville, 1992) at Motorola.

Although so many different approaches have been developed for adaptation, none has become a de facto standard so far. A reason for this is certainly the diversity of software development and the different understandings of the term software process. Therefore a variety of concepts and technologies is used in practice. Instantiation approaches depend on the concepts and technologies which form the basis of the software process. Even when the same technology is used, models can be based upon different concepts. Therefore none of the existing instantiation approaches can be directly used for instantiating Siemens' reference processes. Nevertheless, approaches which define software processes similarly (i.e. as a kind of workflow with tightly coupled activities and artefacts) might contain techniques and concepts which can be adopted and we now discuss some.

The Emergent Process Acquisition method (Jaufman and Münch, 2005) creates a project specific process by choosing the most suitable model from a database and by adapting it further. The latter is done by adapting the Meta model of the process by adding and deleting attributes. The remaining attributes of the process are instantiated. The focus is on milestones and on activities and artefacts which have to be executed and created for the milestone. Tool support focuses on the adjustment of the prescript to the actually executed process by tracking the actually executed activities and by showing the delta between the reference, the instantiated and the executed processes.

The instantiation approach of Yoon et al. (2001) uses processes in the form of Activity-Artefact-Graphs. Activities are connected by artefacts. Semi-automated addition and deletion of activities and artefacts is possible as well as split and merge of activities. Furthermore, syntax checks are available helping process designers maintain consistency when adapting a process. The approach cannot be directly used for the instantiation of the given Siemens process, because the structure of the process used by Yoon et al. is much simpler.

Due to the fact that the given software process has many of the characteristics of workflows, workflow related techniques may also be used for instantiation. Rinderle et al. (2004) compare approaches supporting structural changes of running workflow instances and developed their own approach (Reichert and Dadam, 1998). Although these approaches differ from the given model, considerations regarding maintenance of structural correctness and consistency can be partly applied. This includes control, data and temporal dependencies.

5 CONCLUSIONS

A project to improve Siemens' current approach in instantiating and applying software processes has been put forward. Siemens' software processes were introduced and the current practice of instantiation and application was described. Issues of the current approach were identified and a New Software Engineering Framework (NSEF) for improving the adaptation and application of processes was proposed. Particular goal is the development of a gradual approach of instantiating software processes. The approach distinguishes between high level and detailed instantiation depending on the nature of the operation.

REFERENCES

Aalst, W. and Hee, K. (2004). *Workflow Management - Models, Methods, and Systems*. The MIT Press.

Alexander, L. and Davis, A. (1991). Criteria for Selecting Software Process Models. In *Proceedings of the Fifteenth Annual International Computer Software and Applications Conference*, pages 521–528.

Basili, V. and Rombach, H. (1991). Support for comprehensive reuse. *Software Engineering Journal*, 6(5):303–316.

Becker, U., Hamann, D., and Verlage, M. (1997). Descriptive Modeling of Software Processes. In *Proceedings of the Third Conference on Software Process Improvement (SPI '97)*.

BMI (2004). The new V-Modell XT - Development Standard for IT Systems of the Federal Republic of Germany. URL: http://www.v-modell-xt.de (accessed 08.04.2007).

Boehm, B. (1976). Software Engineering. *IEEE Transactions on Computers*, 25(12):1226–1241.

Boehm, B. and Belz, F. (1990). Experiences With The Spiral Model As A Process Model Generator. In *Proceedings of the 5th International Software Process Workshop 'Experience with Software Process Models'*, pages 43–45.

Bowers, J., May, J., Melander, E., and Baarman, M. (2002). Tailoring XP for Large System Mission Critical Software Development. In Wells, D. and Williams, L., editors, *Extreme Programming and Agile Methods - XP/Agile Universe 2002, Second XP Universe Conference Chicago*, volume 2418 of *LNCS*, pages 100–111.

Fitzgerald, B., Russo, N., and O'Kane, T. (2000). An empirical study of system development method tailoring in practice. In *Proceedings of the Eighth European Conference on Information Systems*, pages 187–194.

Fontana, A. and Frey, J. (1994). *Handbook of Qualitative Research*, chapter Interviewing: The Art of Science, pages 361–376. Sage Publications.

Ginsberg, M. and Quinn, L. (1995). Process tailoring and the software Capability Maturity Model. Technical report, Software Engineering Institute (SEI).

IEEE (1992). 1074-1991 IEEE Standard for Developing Software Life Cycle Processes. Software Engineering Standards Subcommittee of the Technical Committee on Software Engineering of the IEEE Computer Society.

ISO/IEC (1998). ISO/IEC 15504-9 Technology Software Process Assessment Part 9: Vocabulary.

Jaufman, O. and Münch, J. (2005). Acquisition of a Project-Specific Process. In *PROFES*, pages 328–342.

Osterweil, L. J. (1987). Software Processes Are Software Too. In *ICSE*, pages 2–13.

Pedreira, O., Piattini, M., Luaces, M., and Brisaboa, N. (2007). A Systematic Review of Software Process Tailoring. *ACM SIGSOFT Software Engineering Notes*, 32(3):1–6.

Reichert, M. and Dadam, P. (1998). ADEPT$_{flex}$-Supporting Dynamic Changes of Workflows Without Losing Control. *J. Intell. Inf. Syst.*, 10(2):93–129.

Rinderle, S., Reichert, M., and Dadam, P. (2004). Correctness criteria for dynamic changes in workflow systems - a survey. *Data Knowl. Eng.*, 50(1):9–34.

Scheer, A.-W. (2000). *ARIS - Business Process Modeling*. Springer, 3rd edition.

Simmons, R. (2002). *Researching Social Life*, chapter Questioinaires, pages 85–104. Sage Publications, 2nd edition.

Sommerville, I. (1992). *Software Engineering*. Addison-Wesley.

Williams, L. and Cockburn, A. (2003). Agile Software Development: It's about Feedback and Change. *IEEE computer*, 36(6):39–43.

Yoon, I.-C., Min, S.-Y., and Bae, D.-H. (2001). Tailoring and Verifying Software Process. In *APSEC*, pages 202–209.

Zmud, R. W. (1997). Remarks from MIS Quarterly Editor. *MIS Quarterly*, 21(1):V–VII.