

# Adapting PROFES for Use in an Agile Process: An Industry Experience Report

Andreas Jedlitschka<sup>1</sup>, Dirk Hamann<sup>1</sup>, Thomas Göhlert<sup>2</sup>, and Astrid Schröder<sup>2</sup>

<sup>1</sup> Fraunhofer Institute for Experimental Software Engineering,  
Sauerwiesen 6, 67661 Kaiserslautern Germany  
{jedl, hamann}@iese.fraunhofer.de

<sup>2</sup> BMW Car IT GmbH,  
Petuelring 116, 80809 München, Germany  
Thomas.Goehlert@bmw-carit.de  
Astrid.Schroeder@bmw.de

**Abstract.** Background: Agile methods are starting to get established not only in new business organizations, but also in organizations dealing with innovation and early product development in more traditional branches like automotive industry. Customers of those organizations demand a specified quality of the delivered products.

Objective: Adapt the PROFES Improvement Methodology for use in an industrial, agile process context, to ensure more predictable product quality.

Method: An explorative case study at BMW Car IT, which included several structured interviews with stakeholders such as customers and developers.

Result: Adapted PROFES methodology with regard to agility and initial product-process dependencies, which partially confirm some of the original PROFES findings.

Conclusion: The cost-value ratio of applying PROFES as an improvement methodology in an agile environment has to be carefully considered.

## 1 Introduction

Agile methods have a reputation for being faster, more customer-related, and more flexible in the case of unknown or changing requirements. As a result, agile methods are being established not only in new business organizations, but also in organizations dealing with innovation and early product development in more traditional branches, like automotive industry. Nevertheless, customers of those organizations demand a specified quality of the delivered product. It is here that software process improvement (SPI) promises to contribute to an organization's process maturity and, therefore, to obtaining stable and predictable product quality. Today, it is widely accepted that SPI, similarly to software development, has to be performed in a systematic and managed way. The PROFES (ESPRIT project no. 23239: "PROduct Focused improvement of Embedded Software processes") improvement methodology [1] provides a framework of methods and tools that supports industries in their product-driven process improvement. In contrast to traditional process-driven improvement

approaches that are mostly based on process capability and maturity models (e.g., [2]), PROFES starts with the product and especially with product quality to identify improvement potential. The quality attributes are preferably defined by the customer and used to drive improvement activities. This does not mean that, for example, process assessments are not used, but they are not the only source for improvement initiatives. Another important aspect of the PROFES approach, and especially of the idea of systematically managing experience with technology application, is to base decisions on a comprehensible and documented basis, or, in other words, on empirical evidence [3], [4], [5].

On the other hand, agile software development approaches have also become more and more important today (e.g., [6]). Two of the main characteristics of agile development approaches are short cycle times and less documentation in usually small teams. Nevertheless, certain development activities including necessary documents are explicitly demanded and set up.

So far, not much experience exists in improving agile software development processes and maturity. Basically, two scenarios are possible for coming up with a software process improvement (SPI) approach for agile software development: a) define an agile-specific SPI approach from scratch, or b) use an existing SPI approach and adapt it to the specific needs of an agile approach.

For this work we have chosen the second scenario, since the aim of the company, BMW Car IT, was to get a pragmatic solution with early results. The main reason to do so was the expectation to come up with initial results in a shorter period of time. Therefore, an existing software process improvement approach was chosen as the starting point, namely PROFES, since it is especially designed for and used in the embedded domain (due to the automotive background of the company). Before PROFES could be used for an existing agile development process, PROFES had to be adapted.

*Research question:* Is PROFES, although developed as an improvement method for more traditional software development processes, transferable (after the typical, organization-specific adaptations) to an agile process environment?

To answer this question, we started with the identification of major quality issues and product-process dependencies across eight projects. With regard to the quality issues, we documented the shortcomings mentioned by the customers and provided related Goal-Question-Metric-based quality models. To identify product-process-dependencies (PPD), we performed structured interviews. The findings from the quality models and the PPD interviews were combined with information we were able to obtain from a previous CMMi assessment. This led us to the areas with major improvement potential. Finally, we extended the agile process framework with the necessary concepts to keep the PROFES continuous improvement cycle running.

The research question can be answered positively. In addition, we were able to confirm that PPDs found in traditional processes are also valid in agile processes.

The remainder of the paper is structured as follows. First, in chapter two, we give the necessary background to understand the case, a brief insight into the company, the development process at hand, and a sketch of the PROFES methodology. Chapter three gives a summary of related work. The methods used during the study are described in chapter four. The steps performed to adapt PROFES to the agile environ-

ment are shown in chapter five. In chapter six we summarize the main findings and present some lessons learned. The paper is concluded in chapter seven.

## 2 Background

Software has become a fast-growing element in the modern automotive industry (c.f., Fig. 1). During the next few years, the effort for research and development (R&D) of software-based functions will surpass the one for pure electronic R&D. Searching for a better position to meet the challenges of the next decades, the BMW Group founded BMW Car IT in 2002 as a competence center for automotive-specific IT know-how. The task of BMW Car IT is to identify, evaluate, and integrate software-based technologies and methods for and in the BMW Group development process. Therefore, three subgroups are concentrating on different aspects of information technology in the automobile. The Man-Machine-Interface (MMI) group is researching methods and processes enabling the efficient development of MMIs. The second group works on integrated data management to support abstract communication and interaction of car functions. The research on concepts to modify or upgrade software after the cars have left the manufacturing base is done in the software transfer group. Important aspects are coordinated authorized access in different areas of security as well as the guarantee for consistency after changes take place. These fields of research at BMW Car IT are the prerequisite to enable software as a product in the automotive industry.

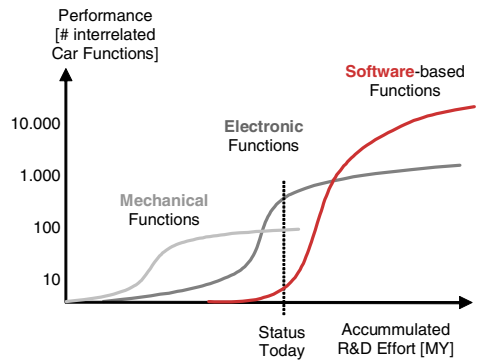


Fig. 1. Potential of Software Technologies

### 2.1 The Process Context

Influenced by traditional software process improvement teaching, the need for stable and predictable processes was identified. The mission for innovation and the vagueness of customer requirements were accepted as main influencing factors for the targeted development process. To establish such a process, a separate project was started to identify possibilities and assess them for the specific demands. Therefore, and to support the business process, different approaches

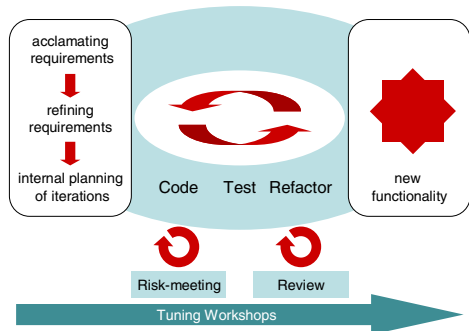


Fig. 2. The agile process at BMW CarIT

ranging from agile methods like “XP” to traditional approaches like the “V-Model” were taken into account. It was found that due to the need for flexibility with regard to customer requirements, fast delivery of first product versions, and involvement of the customer, an agile model will fit the needs best. Nevertheless, it was not the aim to instantiate the Agile Manifesto [7] as is, but a combination of techniques from different agile approaches (mainly XP, but also FDD and Scrum) is needed. The process, as depicted in Fig. 2, uses iterative development with preparation similar to the planning game, selective pair programming, unit tests, refactoring, and acceptance tests.

### 2.2 The SPI Context: PROFES Methodology in a Nutshell

After having applied the process for half a year, a CMMi assessment found some improvement potential. The company's goal to achieve higher quality complies with the target to fulfill CMMi requirements, but the question was how to find an integrated and proper approach for this company to apply, monitor, and assess the improvement actions. Therefore, a flexible but comprehensive improvement approach was needed. The decision for an existing approach was made between the IDEAL (Initiating, Diagnosing, Establishing, Acting, Learning) approach [8], and the PROFES approach [1]. Since the IDEAL approach was not explicitly defined for the embedded domain and defines SPI activities mainly based on the results of a CMM-based software process assessment, the PROFES approach was chosen as the basis for the agile SPI approach.

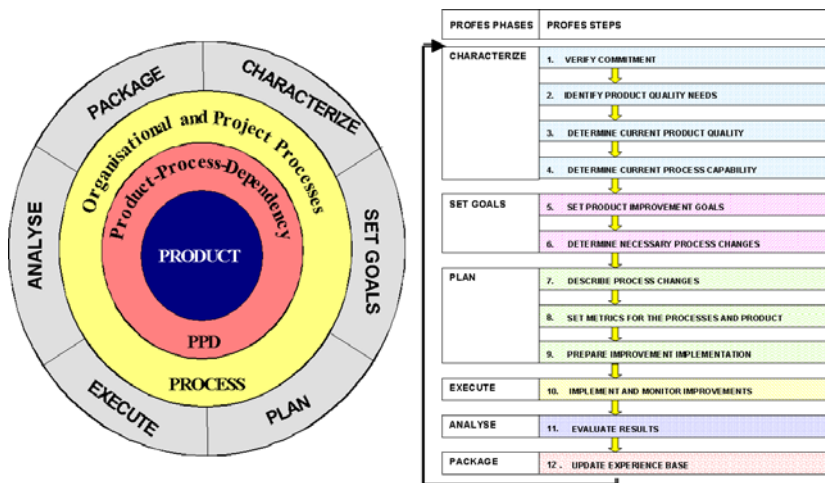


Fig. 3. The PROFES Improvement Methodology

The PROFES improvement methodology [1], [9], [10] uses a modified version of the Quality Improvement Paradigm (QIP) [11]. To illustrate and emphasize the importance of the product as a driver for process improvement, it is placed into the center of the PROFES improvement circle (see Fig.3). The product is the starting point for any improvement activities, starting with the identification of the product quality

needs and the determination of the preliminary product quality goals. Product-Process-Dependencies (PPD) [12], [13] form the linking element between the product and the product development processes. PPD models are used to find and determine the required process changes such that stated product quality improvement goals are achieved. The PROFES improvement methodology consists of six phases, which are further refined into twelve steps. The phases are depicted in Fig. 3. A detailed description can be found in [1].

### 2.3 Project Context

Projects at BMW Car IT deliver mainly textual documents, containing concepts, feasibility studies, and proof of concept. The evidence is shown by developing (throw-away) prototypes that will, as such, not be used in production. To better understand the context, we give a brief categorization of the projects. In contrast to pure software development, the customer of BMW Car IT is mainly interested in the transfer of knowledge, which is typical for R&D organizations. Four different types of projects occur:

- *Feasibility study*, which is an evaluation of solutions regarding realizability
- *Specification*, e.g., of software to be delivered from externals
- *Prototype*, which is a proof of concept of the previous deliverables
- *Concept*, which is a detailed description of a solution for a given problem with alternatives and evaluation. Concept includes the three project types listed above.

The size of projects ranges from one person up to seven, whereas the majority is small to medium-sized; only a few projects consist of a number of six or more developers.

With regard to the agile process framework of techniques, appropriate techniques can be chosen with regard to the type of project. For example, the role of the customer varies. In projects initiated by customers without IT background, many assumptions must be made without the customer being on-site. This changes if the customer directly participates in the project, for example with his own developers. Thus, in the first case a customer proxy can be installed.

## 3 Related Work

Generic frameworks for Software Process Improvement (SPI) are the Software Engineering Institute's (SEI) Capability Maturity Model<sup>®</sup> (CMM)<sup>1</sup> [14], or more recently, the SEI Capability Maturity Model<sup>®</sup> Integrated (CMMI)<sup>2</sup> [15], ISO9000:2000<sup>3</sup>, and the Software Process Improvement and Capability Determination (SPICE)<sup>4</sup> [16]. These frameworks are standards for assessing organizational and software process maturity. They can be used for benchmarking against an ideal set of requirements. But they do not propose concrete SE techniques to be used in specific project situations.

<sup>1</sup> <http://www.sei.cmu.edu/cmm/cmm.html>

<sup>2</sup> <http://www.sei.cmu.edu/cmmi/>

<sup>3</sup> <http://www.iso.ch/iso/en/iso9000-14000/iso9000/iso9000index.html>

<sup>4</sup> <http://www.sqi.gu.edu.au/spice/>

Traditional SPI approaches are thought of as being related to plan-driven development processes.

The most prominent continuous SPI approaches are SEI's IDEAL Model [17] and the QIP, which can be seen as the software engineering equivalent of Total Quality Management [18]. The aim for continuity is reflected through the cyclic nature of the above mentioned improvement approaches. The relation between PROFES and QIP was mentioned before.

Recently, "revolutionary" approaches are coming up, especially in the area of the Internet. They tackle what they call the "bureaucracy" by moving the human factor (communication) into the main focus. To facilitate reading, we summarize all those approaches under the name "agile methods". A recent survey is given in [19]. Nevertheless, according to Boehm [6] a liberal interpretation of CMMI includes agile methods.

Traditional as well as agile methods have to overcome issues like short time to market, frequently changing requirements, low budgets, and high quality demands. Addressing desires such as speeding up software development (more software parts with fewer staff in shorter time) and dealing with vague requirements (late changes) Manhart & Schneider [20] added single agile practices (e.g., test first) to their QIP-based process improvement toolbox.

In addition, there is a lot of research going on to replace or enrich traditional techniques with those compatible with agile development. Especially in the area of measurements [21], [22], [23], [24] and [25] have given much attention to retrospective-like practices.

The main difference between Manhart & Schneider's work and the work presented here is the starting point. They started with the traditional software development process, including process improvement, and brought agile aspects into that framework to improve the development process. Our work started from a given agile process framework and the task was to show how a "traditional" process improvement method (PROFES) can support higher product quality and learning from experience.

## 4 Research Method

The research approach can be summarized as follows. First, we performed a survey among eight experts on the customer site of BMW Car IT (*customer interviews*). Each expert is a customer's representative responsible for one project at the customer site. These experts are the most important stakeholders and have the broadest quality overview regarding requirements and issues. Out of the eight customer projects, six had already finished the starting phase. One project had just been started, whereas another one was just finished. The aim of this survey was to identify relevant product qualities and related characteristics from a customer perspective. The interviews were structured as follows: After some initial questions, we asked for the expected deliverables. For each deliverable, we asked for positive as well as negative experience related to quality. We concluded with questions related to the customer's quality requirements.

For the purpose of finding initial PPDs we used goal-oriented interviews based on the well-known Goal-Question-Metric (GQM) method [26] (*PPD interviews*). We interviewed four senior employees of BMW Car IT (two project leaders, one technical

coordinator, and one developer). For each goal we interviewed each person; additionally, we interviewed two persons each for the project types *prototype* and *concept*.

To find further details and to evaluate the initial findings, we conducted five additional interviews. The interviews were comparable to retrospectives. We confronted the project members with the quality requirements and issues of the customer (our findings from the *customer interviews*). Thus, we asked, e.g., how did you achieve customer satisfaction, or how will you avoid this problem in the future. This led to concrete techniques and more detailed PPDs. Table 1 gives an overview of the sequence of the interviews.

**Table 1.** Sequence of interviews

Step	# Interviews	Interviewees	Purpose of the Interview
1	8	Experts on customer site	Identify product qualities and related characteristics
2	4+4	Senior employees	Identify initial PPDs
3	5	Project members	Evaluate the initial findings

With regard to the acquisition and evaluation of the PPDs, a more quantitative analysis of the projects was not possible, since a measurement program was not in place before.

## 5 Adapting PROFES for Use in an Agile Environment

Before describing what has been done, we give some restrictions. Some are due to practical reasons, whereas others are due to time restrictions. The first step was to decide whether PROFES should be applied in a specific project (bottom-up) or whether it should be applied across projects (top-down) right from the beginning. Taking into account the inhomogeneity of the projects in that company, yielding transferability issues, it was decided to start top-down. Due to time restrictions we skipped phases four (execute) and five (analyze). In phase three, we suggested how to adapt the development process to support PROFES in future.

### 5.1 Product Quality Criteria

For the analysis of the first series of the *customer interviews*, we distinguish between the *concept* and *prototype* project types and categorized the quality criteria accordingly. We found that some requirements and issues are valid for both types of projects, so they were assigned to the category *generic*. Since the company is working in the area of software, we used ISO 9126 as a starting point and extended the given criteria with company-specific criteria like “development time” and “cooperation”. The first one was already addressed in PROFES, whereas the latter one is a new but very important one in this context. Both of them are valid across the company’s

products (projects). Other criteria, like functionality and usability, are more specifically related to the type of project.

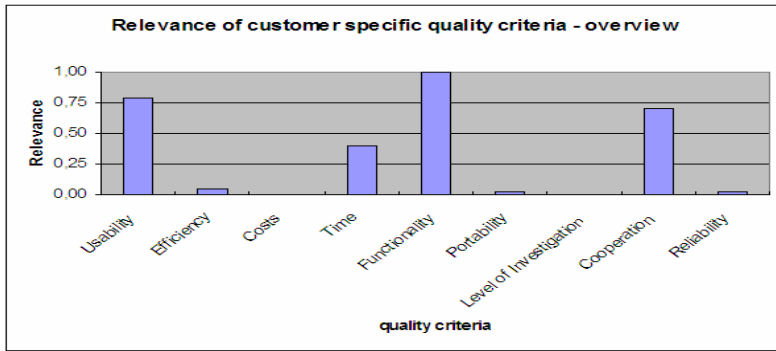


Fig. 4. Overview - distribution of the quality criteria relevance

Fig.4 gives a summary of the analysis of the customer survey. The x-axis depicts the extended list of quality criteria and the y-axis shows the relevance. The relevance is the standardization of the weighted customer statements consisting of number and kind (fulfilled, unfulfilled, wished, and unimportant).

In contrast to the original PROFES approach, where for each project the quality attributes have to be acquired in advance, in the context of R&D, quality and project goals are a matter of evolution, and thus difficult to obtain in advance. In addition, because of the innovative character of R&D projects, the knowledge about what will happen might have a lower level than in more traditional software development.

Table 2. Quality characteristics chosen for further investigation and number of related PPDs found during the interviews

Quality characteristic	Operational problem area	# PPDs
Collaboration	Communication	8
	Infrastructure	
	Controlling	
	Risk management	
Time	Time divergences	9
	Customer overview	
Functionality & Usability	Innovation / Feasibility	7
	Suitability / Understandability	

Within this work, we did not explicitly perform a process assessment as described in PROFES step 4 to identify further process improvement potential. Instead, we used information from a CMMi assessment that was performed during the year before our



work started. The identified weaknesses in addition to the findings from the customer interviews were used as a basis for the definition of the improvement goals.

Based on the customer interviews and the results from the CMMi assessment, we chose the quality goals shown in Table 2.

The reason for the combination of functionality and usability is due to the special context of BMW Car IT in the early development. This also reflects the fact that in requirement documents, functionality and usability cannot really be separated, so an explicit assignment of customer statements was not possible.

## 5.2 Product-Process-Dependencies

Hamann et al. [12] and Birk [27] propose a process for the construction of PPDs. Ideally, the process starts with a literature survey to acquire the state of the art and the state of the practice. In parallel, expert interviews and analysis of existing data are used to get organization-specific information. The theoretical and practical information is combined into initial PPD models. During its life cycle, the PPD is continuously evolved and evaluated (if applied).

For the purpose of this project, we used the analysis of the *PPD interviews*, which yielded initial quality models and more concrete insights with regard to problems. We found eight PPDs for *collaboration*, nine PPDs for *time*, and seven PPDs for *functionality & usability*. These PPDs are already in use and have shown their applicability in different projects (up to four) (c.f. Table 2). Additionally, we found detailed information with regard to metrics.

Further on, we analyzed the PPDs found during the PROFES project with respect to their transferability into this organization. We found four PPDs (candidates), from which two have already been successfully applied, among them a PPD related to development time, which is positively influenced (shortened) by having project and iteration planning installed.

Further investigation of the PPDs provided by the PROFES project yielded few operational results; in most cases, the context was not provided in a way that would have allowed a goal-oriented selection. A comparison with the PPDs developed from the interviews showed that they can be seen as operationalization of the abstract PROFES PPDs that were candidates for this environment. For those PPDs that could not be operationalized, detailed investigation will be done in future projects. In the case of design and requirements PPDs, work has already started. That was the reason for separating the relevant PPDs into two groups: those that are applicable directly, and those that have to be detailed further.

### 5.2.1 The Influence of Context Factors on PPD Evolution

Many authors have already discussed the need of context factors. Birk describes a two-layer model with different abstraction levels based on taxonomies, e.g., for processes [27]. In many cases, information about the company, like domain, size, etc. as well as project and team characteristics are used to describe the context in which a certain technology has been applied in.

Since this work was performed in a project organization, we started with the characterization of the projects. On this level the more general context factors describe the kind of project, (1) the definition of work, (2) the number of team members, (3) the

number of stakeholders, (4) the responsible group, and (5) the proximity to production. The context factor definition of work is a four-dimensional vector, whose axes are related to the proportions of conceptual work, coding, theoretical investigation, and demonstration. Each axis has four possible values: none, low, medium, and high. These factors were provided by the interviewees for their specific project. Later on, the more technology-specific factors were acquired, for example, what Jedlitschka et al. [4] call pre-condition for the application of the technology.

The interviews with experts, such as project managers and leading developers, yielded context factors, technology experience, and best practices. Due to the clarity of the process, the relationship between PPD and process area was implicit.

PPDs found in this company differ from the PROFES PPDs [1], especially in the handling of context factors. To allow us to also learn from negative experience, we extended the status of context factors. A context factor can also be “negatively validated”. Additionally, we do not only have a single context model for one PPD, but a separate context model for each single application. This gives us the opportunity to provide “historic” information and assures that we do not loose information through the process of aggregation. Nevertheless, we provide aggregated context information to the user, thus not overloading him with unnecessary information. Additionally, we do have descriptions of the contexts in which a PPD has been applied, and so we are able to validate them on different levels.

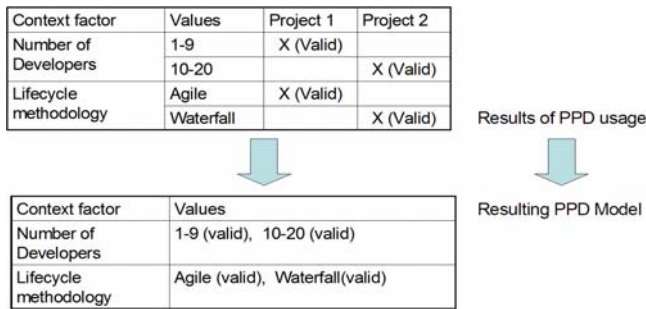


Fig. 5. Resulting problems of simple context aggregation

The example given in Fig.5 depicts the problem that would arise with “simple” aggregation of context factors. Assume we would have applied a certain technique in two different projects, *Project 1* and *Project 2*. The technique was applied successfully in *Project 1* using an agile life cycle method and eight developers. The same technique was also successfully applied in a waterfall-driven project with 19 developers. “Simple” aggregation would lead to the assumption that the technique is also applicable in a project within an agile environment and with a team consisting of 18 persons, or vice-versa. Although validated in their specific combination, the new combination is only a hypothesis and not validated yet. By taking the “history” into account, we help the decision maker not to be trapped by misleading assumptions.

The downside of this approach is that we have to explicitly deal with contradictory findings. One reason, especially for contradictions over a longer time-span, might lay

in the evolution of the organization. If there are contradictions, the reason is in the application itself or within the relationships of the context factors; the discriminating one is missing. In this case we suggest a careful investigation of the context factors, e.g., through technology reviews, followed by aggregation supported by the technology experts. Returning to the given example, the PPD could be separated considering the lifecycle as a separating context factor. The conclusion is that only those PPDs can be aggregated (automatically) that vary in the instances for only one context factor. Additionally, we need to differentiate between the levels of validity; a PPD that has been applied successfully, e.g., eight times, has a greater validity than one that has only been applied once.

### 5.2.2 PPD Repository

In contrast to the original PROFES approach where the PPDs are stored in a database-driven repository, the PPDs are stored in a *Wiki*<sup>5</sup>. This allows distributed “content (PPD) management”, since everyone is allowed to provide information. In general, the process team hosts the *Wiki* and supports people with their PPD-related work. Especially in case of contradictions, the process team acts as “mediator”, thus is responsible for resolving the contradiction as mentioned above. This very open solution might only fit an organization with smoothly cooperating developers where security is not a big issue. We are well aware that for bigger organizations and those with more separated projects, security and PPD management are an issue.

## 5.3 Adaptation of PROFES

Finally, we suggest how to use and evolve the PROFES methodology in the agile process environment.

The main issue, after having shown that PROFES is, in general, usable in an agile environment, and after having collected initial organization-specific PPDs, was to extend the agile process with aspects that are necessary in order to be able to benefit from PROFES. One conflicting requirement with regard to the adaptation of the agile process was that the agility of the process environment should remain. This required some adaptations of the PROFES approach.

The first issue is that because of the agile environment, the goals and therefore the quality attributes are not fixed from the beginning. The goals are to be defined during the project itself. The original PROFES approach requires the goals right from the beginning, since they are used to perform the goal-oriented tailoring, that is, selection of the right technology taking the quality goals into account.

The second issue is that the iterations are very short, i.e., on average six weeks, which makes it very hard to fulfill all PROFES steps. Nevertheless, we found that many projects have a rough plan that spans more than one iteration, especially if the project has already been running for some time.

---

<sup>5</sup> *Wiki* is in Ward's original description: The simplest online database that could possibly work. *Wiki* is a piece of server software that allows users to freely create and edit Web page content using any Web browser. *Wiki* supports hyperlinks and has a simple text syntax for creating new pages and cross links between internal pages on the fly [from <http://wiki.org/wiki.cgi?WhatIsWiki>].

The third issue is that the projects do not have the resources to deal with formal PPD construction. In some cases, when, for instance, a new technique was applied, PPD construction is implicitly done, but the formal description has to be supported by the organization, i.e., the process team.

Fig.6 depicts the extension of PROFES with regard to the agile environment. Whereas the standard PROFES methods can be applied on the organizational level and on the project level, we found that on iteration or group level, some more “light-weighted” methods are to be used.

Based on these circumstances, we suggest to adapt PROFES as follows:

- In the phase *start of project* after the initial requirements elicitation, the initial goals for the project are available. Based on these goals, it is possible to apply PROFES. In this phase we propose to use “heavy-weighted” activities like assessments, but also the preparation and enactment of a measurement plan.
- In the subsequent iterations, the application of PROFES is useful if important aspects have been changed, e.g., quality goals. In the beginning of the projects, this occurs quite often, so the suggestion is to use only “light-weighted” PROFES aspects, like interviews with customers and usage of an online PPD repository. For now, the company has decided to use regular requirements meetings and resulting lists of requirements and issues to track product quality from a customer perspective. These lists are also used to select appropriate techniques. After having applied the technique, the application experience has to be stored in the *Wiki*.
- The explicit PPD construction also needs to be done, as described in this work, on an organizational level. This is in slight accordance with the QIP paradigm where we do have an organizational level and a project level.

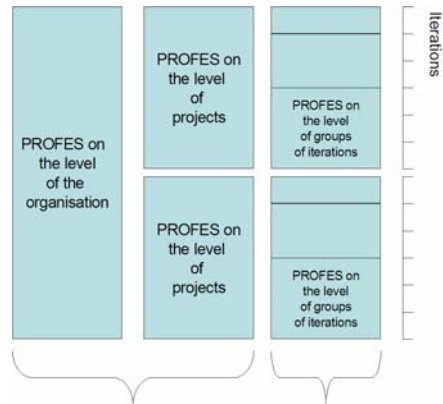


Fig. 6. Aggregating PROFES and agility

## 6 Main Achievements and Lessons Learned

Based on the work presented here, we present results for four perspectives.

For the PROFES perspective, we have provided evidence that PROFES is highly adaptable because of its flexibility in using the underlying methods and technologies. A more comprehensive guideline with references to known alternatives would be helpful, but could not replace an introduction-/adaptation phase of PROFES in a new organization. Ideally, the target process is also adapted to improve the integration of PROFES. This includes, for instance, a higher documentation effort for the developer, or the extension of the project cycle by an additional iteration cycle.

For improving PROFES, we suggest to

- consider negative experience as valuable for being stored in the PPDs
- extend the customer-oriented view of PROFES beyond the product quality horizon with aspects like quality of cooperation if there is an influence on the product quality

For the agile perspective, we found that the proposed approach supports learning. Traditionally, agile environments heavily depend on interpersonal communication (socialization [28]) to transfer knowledge. The approach proposed here extends this kind of communication by active knowledge elicitation, i.e., by using interviews, and the publication of this knowledge within PPDs. In cases where “light-weight” is not the key issue, continuous assessments [29], in comparison to traditional assessments, reduce the effort (time and resources) needed for assessments [30] and thus contribute to the agile idea.

For the organization perspective, we found that:

- the primary benefit is the exploration of the customer quality requirements, and the integrated way of achieving their fulfillment.
- PROFES assists in the development of a learning organization, which is a critical success factor for the evolution of an organization, especially in R&D.
- the need for achieving higher capability or maturity levels in a process becomes a more sophisticated aspect. This helps to invest available resources in product quality-related processes in a goal-oriented manner with respect to the organizational needs.

From a customer perspective, we found that:

- the integration of the customer in product quality improvement gives him the chance to articulate his quality requirements explicitly, but he also assumes the risk of failure if he does not do so.
- through the methodological assistance, it becomes easier to formulate product quality requirements that are easier to understand and realize by developers.

## 7 Summary

With this work we have shown that PROFES as an improvement method can be applied in an agile environment. The adaptation of PROFES to the organization-specific requirements did not exceed the expected effort. Besides this finding and the acquisition of the initial PPDs, some of the original PROFES PPDs were revalidated. So we have at least initial indicators that techniques are transferable between traditional and agile environments.

Adaptations with regard to the agile process framework were necessary in the area of measurement and documentation. We did not start a dedicated measurement program (although measurement is also a topic in agile environments); instead, we proposed retrospectives using the quality models to track progress and quality, and to discuss the usability of PPDs. To overcome the issue of documentation in agile environments, we proposed a *Wiki*-based PPD infrastructure.

The experience we made at BMW Car IT show that combining PROFES with an agile approach is possible and useful, although, it requires some effort, in particular, when the issue is to keep the resulting process possible as “light-weighted” as possible. At BMW Car IT, we use process review meetings at the end of each iteration to reflect upon and document the context factors under which certain best practices did or did not work. In summary, the cost-value ratio of applying PROFES in an agile environment has to be carefully considered.

## Acknowledgements

We would like to thank all interviewees at the customer site who have contributed to the success of this work by providing their expert knowledge. Without their support, this work would not have been possible. We would also like to thank Dietmar Pfahl for fruitful discussions and the anonymous reviewers for giving valuable feedback, thus helping to improve the paper. Furthermore, we are grateful to Sonnhild Nam- ingha from the Fraunhofer Institute for Experimental Software Engineering for re- viewing a previous version of this paper.

## References

- [1] The PROFES Consortium. The PROFES User Manual. Fraunhofer IRB Verlag, Stuttgart, Germany, 1999.
- [2] CMMI Product Team. Capability Maturity Model Integration CMMI, Version 1.1 CMMI for Systems Engineering and Software Engineering (CMMI-SE/SW, V1.1) Staged Representation. Technical Report CMU/SEI-2002-TR-002, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, December 2001.
- [3] Kitchenham, B.A.; Dybå, T.; Jørgensen, M.; “Evidence-based Software Engineering”, in Proc. Intern. Conf. on Software Engineering, May 2004 Edinburgh, Scotland, UK, pp. 273-281
- [4] A. Jedlitschka; D. Pfahl, and F. Bomarius: “A Framework for Comprehensive Experience-based Decision Support for Software Engineering Technology Selection”; In Proc. of Intern. Conf. SEKE 2004. Banff, Canada, 2004, pp. 342-345
- [5] A. Jedlitschka; M. Ciolkowski: Towards Evidence in Software Engineering; In Proc. of ACM/IEEE ISESE 2004, Redondo Beach, California, August 2004, IEEE CS, 2004
- [6] B.Boehm; Get Ready For Agile Methods With Care; Computer, Vol. 35, 2002, pp.64-69,
- [7] Agile Alliance. Manifesto for Agile Software Development, <http://www.agilemanifesto.org>, 2001
- [8] B. McFeeley. IDEAL: A User’s Guide for Software Process Improvement. Handbook CMU/SEI-96-HB-001, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, February 1996.
- [9] A. Birk, J. Järvinen, S. Komi-Sirviö, P. Kuvaja, M. Oivo, and D. Pfahl. PROFES – A Product Driven Process Improvement Methodology. In Proceedings of the Fourth European Conference on Software Process Improvement (SPI ’98), Monte Carlo, Monaco, Dec 1998.
- [10] M. Oivo, A. Birk, S. Komi-Sirviö, P. Kuvaja, and R. v. Solingen. Establishing Product Process Dependencies in SPI. In Proceedings of the European Software Engineering Process Group Conference 1999 – European SEPG99, Amsterdam, The Netherlands, June 1999.

- [11] V.R. Basili, G. Caldiera, and H.D. Rombach: Experience Factory; in: Marciniak JJ (ed.), *Encyclopedia of Software Engineering*, Vol. 1, pp. 511-519, John Wiley & Sons, 2001.
- [12] D. Hamann, J. Järvinen, A. Birk, and D. Pfahl. A Product-Process Dependency Definition Method. In *Proceedings of the 24th EUROMICRO Conference: Workshop on Software Process and Product Improvement*, Volume II, pages 898-904, Västerås, Sweden, August 1998. IEEE Computer Society Press.
- [13] D. Hamann, J. Järvinen, M. Oivo, and D. Pfahl. Experience with explicit modelling of relationships between process and product quality. In *Proceedings of the Fourth European Conference on Software Process Improvement (SPI '98)*, Monte Carlo, Monaco, Dec.1998.
- [14] Jalote, P.; "CMM in Practice. Processes for Executing Software Projects at Infosys" Addison-Wesley, Reading; 1999.
- [15] Chrissis, M. B.; Konrad, M.; Shrum, S.; "CMMI. Guidelines for Process Integration and Product Improvement"; Addison-Wesley, Boston; 2003,
- [16] El Emam,K.; Drouin,J.-N.; Melo,W. (Eds.); "SPICE. The Theory and Practice of Software Process Improvement and Capability Determination"; IEEE CS, Los Alamitos, 1998.
- [17] Gremba, J.; Myers, C.: "The IDEALSM Model: A Practical Guide for Improvement" in *Software Engineering Institute (SEI) publication*, Bridge, issue three, 1997.
- [18] S.K.M. Ho; *TQM – An integrated approach*; School of Business, Hongkong Baptist University; 1999
- [19] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta. *Agile Software Development Methods. Review and Analysis*. VTT publications 478, VTT Electronics, Espoo, Finland, 2002.
- [20] P. Manhart, K. Schneider: *Breaking the Ice for Agile Development of Embedded Software: An Industry Experience Report*; IN Proc. Of the 26th Intern. Conf. on Software Engineering (ICSE'04), Mai 2004, Edinburgh, Scotland, pp 378-386
- [21] M. Visconti and C. R. Cook. *An Ideal Process Model for Agile Methods*. PROFES 2004, pages 431-441.
- [22] A. Birk, T. Dingsøyr, T. Stalhane; *Postmortem: never leave a project without it*; IEEE Software, Vol. 19 , Issue: 3 , May-June 2002, pp. 43 - 45
- [23] J. Kerievsky. *Continuous Learning*. Proc XP. 2001.
- [24] C. T. Collins and R. W. Miller. *Adaption: Xp style*, Proc XP. 2001.
- [25] T. Dingsøyr and G. K. Hanssen, *Extending Agile Methods: Postmortem Reviews as Extended Feedback*, Learning Software Organizations Workshop, Chicago, Illinois, USA, 2002
- [26] V.R. Basili, G. Caldiera, and H.D. Rombach: *Goal Question Metric Paradigm*; in: Marciniak JJ (ed.), *Encyclopedia of Software Engineering*, Vol. 1, pp. 528-532, John Wiley & Sons, 2001.
- [27] A. Birk: *A Knowledge Management Infrastructure for Systematic Improvement in Software Engineering*; Ph. D. diss., Dept. of Computer Science, University of Kaiserslautern, Germany; Stuttgart: Fraunhofer IRB Verlag; 2000.
- [28] I. Nonaka, H. Takeuchi. *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press, New York 1995.
- [29] J. Järvinen. *Measurement based continuous assessment for software engineering processes*. PhD thesis, VTT Publications, Technical Research Centre of Finland, 2000.
- [30] J. Järvinen and R. v. Solingen. *Establishing continuous assessment using measurements*, in *Proc. of the 1<sup>st</sup> International Conference on Product Focused Software Process Improvement 8PROFES'99*, Oulu, Finland, June 22-24. 1999, pp. 49-67