

**Copyright: EuroSPI 2005, Will be presented at 9-11 November,  
Budapest, Hungary**

## **Deploying Agile Practices in Organizations: A Case Study**

Minna Pikkarainen<sup>1</sup>, Outi Salo<sup>1</sup>, and Jari Still<sup>2</sup>

<sup>1</sup>VTT Technical Research Centre of Finland, P.O. Box 1100, FIN-90571 Oulu, Finland

<sup>2</sup>F-Secure Corporation, Elektriikkatie 3, 90570 Oulu, Finland

<sup>1</sup>[Minna.Pikkarainen@vtt.fi](mailto:Minna.Pikkarainen@vtt.fi), <sup>1</sup>[Outi.Salo@vtt.fi](mailto:Outi.Salo@vtt.fi), <sup>2</sup>[Jari.Still@f-secure.com](mailto:Jari.Still@f-secure.com)

**Abstract.** Currently, software development organizations are increasingly interested in adopting agile processes and practices. The organizations, however, need procedures and methods for supporting a systematic selection and deployment of new agile practices and for tailoring them to suit the organizational context. In this paper, an agile deployment framework is proposed. It is compatible with the ideology of continuous improvement of organizational practices (QIP), while it also integrates it with the opportunities provided by short iterations of agile process model. The suggested framework includes the procedures and methods needed for selecting suitable new agile practices in an organization. It also embodies the means for iteratively tailoring and validating the deployed practices within agile projects and gaining feedback rapidly from projects to the organization. The paper presents the empirical experiences of a case study where the F-Secure Corporation deployed a new agile software development process (Mobile-D) in a pilot project in order to utilize its experiences in developing an organization specific agile process model alongside their traditional F-Secure product realization process.

## **1 Introduction**

Over the past years, there has been increasing interest towards agile software development methods and practices. Agile software development attaches weight to, for example, rapid responding to constant changes and increasing customer collaboration ([agilemanifesto.org](http://agilemanifesto.org)). In spite of the promising experience reports of applying agile practices [1, 2], their deployment is a challenging task demanding a great deal of adjustment from all the stakeholders involved in the software development process (e.g., software developers, testers, management, and customers) [1, 2]. Thus, organizations need agile specific guidelines and methods to support systematic selection, deployment and tailoring of agile practices to fit the organization's software development context. In this paper, an agile deployment framework is proposed in order to provide organizations with procedures for adopting and improving practices in the agile software development context. The suggested framework and its steps are designed to comply with the continuous improvement ideology of the Quality Improvement Paradigm (QIP) [3]. However, since the existing software process improvement (SPI) approaches, such as QIP, have

originally been developed for the context of the traditional software development, they do not necessarily include all the elements and possibilities provided for the deployment by the agile software development process. For example, the iterative process adaptation within agile project teams is addressed in the principles of agile software development ([www.agilemanifesto.org/principles.html](http://www.agilemanifesto.org/principles.html)). This provides project teams with a means of iterative tailoring the deployed practices in a validated manner and offers organizations rapid feedback from the deployment [14].

The traditional SPI methods can be utilized in the deployment of agile practices, e.g. the Goal-Question-Metric method for identifying feedback metrics [4]. However, the agile deployment framework identifies the agile specific methods that support the various tasks of deploying agile practices (i.e., agile assessment [5] used for setting goals and identifying suitable agile practices, and post-iteration workshops [6] for iteratively improving, validating and packaging feedback in projects).

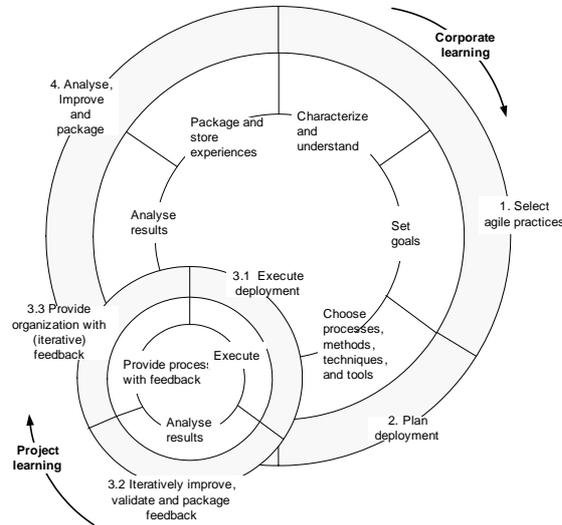
This paper presents the empirical experiences of a case study where the F-Secure Corporation adopted an entire agile software development process (i.e., Mobile-D [7]) in order to evolve a agile approach alongside the traditional F-Secure product realization process. Thus, the aim of this study is to evaluate the proposed framework and to present how the steps of the agile deployment framework provide a loop for continuously improving organizational software development practices. The paper is composed as follows: Section 2 presents the agile deployment framework; Section 3 contains the research goals and context; and Section 4 the empirical evidence from the case study. The last section concludes the paper with final remarks.

## **2 The Agile Deployment Framework**

There are many different SPI approaches addressing continuous and systematical improvement of software development processes in organizations, such as the QIP [8]. The existing approaches include the aspect of deploying new practices if these are required to meet the organizational improvement goals. In QIP, two cycles of improvement are identified: 1) the organizational learning cycle in which, for example, the improvement goals and improvements are executed, and 2) project learning cycle which is used, for example, for piloting and for collecting feedback needed for finding problems and validating improvements. Many of the existing SPI approaches are goal-oriented and address the utilization of metrics data from software development projects in selecting and evaluating process improvements.

In this paper, an agile deployment framework is proposed. It is designed to integrate the iterative cycles of agile software development with the continuous improvement of organizational practices. Its focus is on deploying agile practices in organizations and it addresses the importance of utilizing the experiences of the software developers an important source of input to SPI. In Fig. 1, the original cycle of QIP (white) is mapped with the steps of the Agile Deployment Framework (grey): select agile practices, plan deployment, execute deployment, analyze and package results, and improve. The main difference of the proposed approach compared to traditional approaches is in its iterative execution of deployment, which provides

feedback from the iterative improvement and from the validation of the deployed practices in software development projects.



**Fig. 1.** QIP cycle (from [9]) and Agile Deployment Framework

Table 1. defines the steps of the QIP approach [8] and maps them with the steps of the agile deployment framework. The main activities of the deployment steps as well as the suggested agile specific methods to support deployment (i.e., agile assessment [5] and post-iteration workshops [10] (hereafter referred as PIWs)) are also included. The PIW method was evolved based on two existing agile reflection techniques, namely the reflection workshop technique by Cockburn [11] and the postmortem reviews by Dingsøy et al., [12]. The PIW method, however, has been complemented with systematic planning, follow-up, and validation of SPI actions [10].

**Table 1.** Mapping the Agile Deployment Framework with QIP

QIP Steps	Main Activities	Agile Deployment Steps	Main Activities	Agile Methods
1.Characterize and understand	Gather knowledge of projects	1.Select agile practices	Set goals for deployment	Agile assessment
2.Set goals	Set goals for improvement		Identify suitable practices	Agile assessment
			Select practices to deploy	-
3.Choose processes, methods, techniques, tools	Define models needed by a project to achieve the goals	2.Plan deployment	Plan deployment	-
			Prepare deployment	-
4.Execute	Implement the plans, collect measurement data and provide feedback to project	3.Execute deployment	Execute deployment	-
			Iteratively improve, validate and package feedback in projects	PIW
5.Analyze results	Analyze project practices, problems, findings and recommendations	4.Analyse, improve and package	Analyze project feedback to identify improvements	Agile Assessment
6.Package	Package experiences and ensure their use in future projects		Improve the organizational processes	-
			Package	PIW

In the following, the agile deployment steps are defined in more detail.

## 2.1 Select Agile Practices

An organization should first *set goals for deployment* and consequently, *identify* the potential agile practices. The existing ways to discover the agile methods to deploy are unstructured; for example, one may study the current agile literature or gain knowledge from partners who have already applied certain agile practices. The agile assessment [5], however, provides systematic and goal-driven mechanisms for *identifying* and *selecting* suitable agile practices for the organization specific context.

The steps of agile assessment are: 1) focus definition, 2) agility evaluation, 3) data collection planning, 4) data collection, 5) analysis and 6) final workshops. In the first step, the goals are set for adopting agile methods. The second step provides a better understanding on how suitable and effective the various agile methods would be in specific projects. The agile assessment data can be collected using interviews, agile assessment workshops, and from the recorded iterative SPI actions (from PIWs) and improvement opportunities (from project postmortems). In addition, various metrics data can be utilized in the analysis. Agile assessment workshops are conducted in order to identify the strengths and weaknesses of the software development process and to discuss the possibilities of increasing the agility of the development process together with the project stakeholders. The assessment workshops support project and organizational learning between different projects and also the development of an organizational level agile software development model. The agile assessor should be well aware of the available agile methods as well as the agile assessment method.

## 2.2 Plan Deployment

Organizations have different approaches to the deployment of new practices. An organization can, for example, select a pilot project or even embody the new practices directly in its organizational software development processes. Whether an organization plans to experiment with the new practices in a pilot project or to deploy the new practices in a larger scale, it should also *plan* how empirical feedback is provided for a continuous improvement of organizational practices. For example, it should be defined how the suitability of each adopted method will be evaluated during the piloting, and how the feedback from the (pilot) projects is stored and analyzed. Thus, in this step of agile deployment, it should be ensured that there are mechanisms available for the project teams to collect and store the relevant feedback in an appropriate format from projects to the organizational level.

The deployment phase also includes the *preparation* of projects involving changes to the daily software development practices. The preparation includes, for example, training, tailoring the deployed practices to fit the existing process, and preparing the tools considering the used practices. The deployment, thus, includes all the preparations needed for using the selected new practices in the selected projects.

### 2.3 Execute Deployment

Unlike the other steps of the agile deployment model, the execute step is conducted at the project level. Its focus, from the organizational viewpoint, is to gain feedback from the deployed practices in order to enhance the organizational processes. The execution of deployment consists of three steps: 1) execute deployment, 2) iteratively improve, validate, package feedback, and 3) provide the feedback to an organization. In the QIP, the execute step is defined as the project learning cycle (Fig. 1).

The projects selected for deploying agile practices can be regarded as pilot projects providing the organizational level with feedback on applying new agile practices. The short development cycles of agile software development provide rapid loops, which allow project teams to *iteratively improve* and adapt their daily working practices in a validated manner [13] based on their own experiences and domain knowledge. From the viewpoint of deploying new practices in an organization, this kind of iterative adaptation and *improvement* also provides a means for organizations to gain on-time feedback on how the project teams have adapted and improved their practices.

In agile deployment, the PIW method can be used for two purposes: 1) to provide project teams with a mechanism to tailor the deployed and the existing software development practices during the ongoing project in a validated manner, and 2) to provide the organizational level with mechanisms for gaining systematic and rapid feedback from the process improvement of (pilot) projects. The validation is done by implementing process improvements in the ongoing project and iteratively evaluating their usefulness with available metrics and experience data.

At the end of the software development project, the last PIW can be conducted as a traditional project postmortem [14]. As the project team will no longer be able to implement or validate the improvements at this point, the goal of the project postmortem is to harvest process knowledge from the stakeholders of the project teams solely for organizational improvement purposes. The postmortems, thus, provide another experience based feedback mechanism from projects to organization.

The PIW method offers mechanisms to *provide the organization with iterative feedback* from individual projects. A structured action point list [13] suggests how the SPI actions may be iteratively documented in a project in order to support SPI in an ongoing project and to provide validated SPI knowledge from projects to organizational improvement activities. Thus, the action point list includes the identification of the following issues for each improvement action: 1) the exact problem that the action point aims at solving, 2) the specific action to be taken, 3) the responsibilities for implementing the action and schedule, 4) the means to validate the usefulness of an action point, and 5) the results (qualitative or quantitative) of validation (updated in the following PIW after piloting). Another output of PIWs are the flap-sheets containing grouped experiences of the project team, which form the basis for the improvement actions of the project team (see more in [6, 15]).

### 2.4 Analyze, Improve, and Package

The key purpose of the *analyze, improve and package* step is to make sure that the deployed practices that have been found useful in the pilot projects are identified and

employed in the organization. In the agile deployment framework, the organizational level can gain process knowledge from two sources: 1) agile assessments and 2) individual projects. More specifically, the projects can provide the organizational level with experience based process knowledge (validated improvements from PIW's and improvement opportunities from project postmortems). As suggested in QIP, the projects may have collected metrics data defined at the organizational level. The feedback from projects is analyzed, the improvement actions planned and implemented, and the results stored and packaged for later SPI purposes.

### **3 Research Context**

In this section, the goals, context, and methods of this research are presented.

#### **3.1 Research Goals and Methods**

The goal of this research is to evaluate the proposed agile deployment framework in an industrial context. In particular, the usefulness of the agile specific methods integrated in the agile deployment framework is assessed, i.e., agile assessment [5] for *selecting* suitable agile practices in individual projects and within an organization and PIWs [6] for a continuous *adaptation and improvement* of these practices. In other words, the goal of this study is to evaluate if iterative software development model provides added value to the deployment of new practices and how it bonds with the loop of continuous improvement of organizational software development.

This research can be characterized as constructive research, in which a case study forms the basis for further development and evaluation of the proposed agile deployment model and the methods integrated in it. As a researcher was acting as a facilitator in the PIWs and project postmortem, an action research approach (e.g., [16]) was applied especially in activities concerning project level SPI. Both the agile assessor and the facilitator participated in the improvement activities at the end of the project. A participative approach enabled an effective way to “integrate theory with practice through an iterative process of problem diagnosis, action intervention, and reflective learning” [17] throughout the case study. Both quantitative and qualitative data was collected from project and organizational SPI activities. In addition, a questionnaire was prepared to collect the developers' perceptions of the PIWs.

#### **3.2 Research Context**

The case study of this research was conducted at F-Secure Corporation, an organization developing products to protect individuals and businesses against computer viruses and other threats spreading through the Internet and mobile networks. At F-Secure, a project named Phantom was set up to pilot an agile software development process (i.e., Mobile-D [7]) that had earlier been developed at VTT.

The goal of the Phantom project was to develop a mobile security application. The core of the case project team consisted of four software developers and one tester who were working in an open office space. The Phantom team conducted five software development iterations in all (1x1 week, 3x2 weeks, 1x1 week) and completed a total of 7.2 person months of effort. The team leader of the project provided by the research organization was an expert in the Mobile-D process. Thus, the team had constant support and coaching available on adopting the new agile practices. Other stakeholders of the project were the organizational management, a project manager, two customers and quality engineers, and an exterior facilitator. The customers were available on-site in the same department, but not constantly working in the Phantom office-space as suggested in Extreme Programming (XP) [18].

## **4. Case Study**

In this section, the most important empirical results are presented concerning how the case organization conducted the deployment of Mobile-D in the Phantom project.

### **4.1 Select Agile Practices**

The goal at F-Secure was to deploy an agile software development model (i.e., Mobile-D) in a pilot project in order to utilize its experiences in evolving an organization specific agile process model. Prior to launching the Phantom project, the Scrum method had already been introduced in a few projects. The Mobile-D process itself contained the methods for gaining feedback from projects to the organization (i.e. PIWs, project postmortem, and defined metrics). These methods were systematically used in the case organization for iterative adaptation of the used practices in the project and in order to provide the organization with validated improvements and improvement opportunities from the case project.

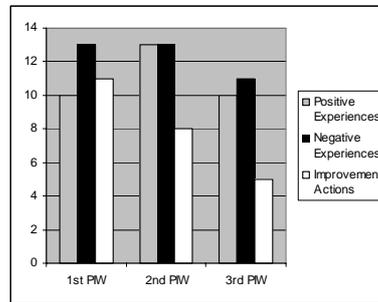
### **4.2 Plan the Deployment**

At F-Secure, various activities were needed for setting up the pilot. Firstly, in order to ensure a successful deployment of Mobile-D, the project team of F-Secure was complemented by developers from VTT, who were experts in Mobile-D and could thus provide on-line coaching for the in-house developers. Many of the agile practices and tools included in the Mobile-D process were new at F-Secure. Thus, a software development (e.g., unit testing tools) and working environment (e.g., open-office space) was set-up, and the project team was trained to use the new procedures. In the case project, however, no tailoring of the deployed practices to the existing organizational processes was needed as Mobile-D was adopted as such.

### 4.3 Execute Deployment

The iterative *improvement, validation and packaging* tasks were ensured by adopting the PIW method and by conducting a project postmortem. The Phantom project team collected a fair amount of metrics data, as suggested by Mobile-D. The data was used, for example, for validating the iterative process improvements in PIW's.

In Phantom, a total of three PIWs were held after the first three iterations. The workshops were attended by the project team and also by one of the customers and some quality assurance team members. The participants first collected positive and then negative experiences from the previous iteration on a flap-sheet. The facilitator (expert in the Mobile-D process) led the discussion using the negative experiences as a basis to define process improvements for the next iteration.



**Fig. 2.** Quantity of Phantom Post-Iteration Workshop Results

Fig. 2 illustrates the number of positive and negative experiences, as well as the implemented improvements resulting from the three subsequent PIWs. It should be noted that there were five participants in the 1<sup>st</sup> PIW and seven in the last two. Thus, the declining trends in all the categories presented in Fig. 2 would be even more distinct if relative numbers were presented for the findings of the 1<sup>st</sup> PIW.

Each PIW resulted in a structured action point list (see more in [13]), which was put on the wall of the open-office space and also iteratively e-mailed to project management for monitoring and organizational improvement purposes. Thus, the PIW data was iteratively packaged in the project and delivered to the organizational level. For each process improvement, the specific improvement action, the reasons for it, the means of validation and its effectiveness were documented. After the validation (i.e., after the improvement had been experimented in project iteration) the proven usefulness or non-usefulness of the process enhancement was also documented.

**Table 2.** Most Important Improvement Categories in the Phantom Project

Improvement Category	Improvement Actions	Negative Experiences
Quality Assurance	8	3
Pair-Programming	4	7
Project Monitoring & Management	4	1

The PIWs revealed several problems and produced a number of improvement solutions. The top three improvement categories are illustrated in Table 2 along with the number of resulting improvement actions and the amount of negative experiences on each topic. As it can be seen in Table 1, several improvements were needed on the

Quality Assurance (QA) category, which includes issues related to unit testing, verification of tasks, and system testing. The Pair-Programming (PP) practice was also found highly controversial throughout the project. Some project members (4/7 of negative experiences) wished to increase the use of PP in the project, whereas the others (3/7 experiences) found it mostly unnecessary. For solving this problem, the team agreed to iteratively identify the tasks that would require PP. However, due to the resistance of a proportion of the project team, none of the tasks were identified as such and in the second iteration, for example, only two out of a total of 12 tasks were partially implemented using PP. Thus, the team failed to reach an agreement during the project on how extensively the PP practice should be adopted. The third most active improvement category was project management, which was mainly concerned with the improvement of the templates used for defining tasks and improving the usefulness of the information radiator [11] for project monitoring.

In addition, a project postmortem was held after the Phantom project together with the Phantom project team and its stakeholders. The aim was to distinguish the most suitable and unsuitable agile practices for the F-Secure specific agile process. Because of the Agile Assessment purposes, the Phantom postmortem was organized together with the PIW facilitator and the agile assessor. In the postmortem, the project stakeholders identified the most suitable and unsuitable practices of Mobile-D process. The best practices identified were unit testing, the incremental process model, and iterative planning of tasks with the customer. The most unsuitable practices were the PP practice, open office space, and the procedures of QA. In the postmortem, improvements for the three top unsuitable agile practices and the key benefits of the best agile practices compared to their traditional plan driven software development approach were also identified. On the basis of the project experiences, a number of problems and solutions were revealed. These were summarized by the facilitators and reported to the F-Secure management for further analysis.

At F-Secure, the PIWs were found a useful method of improving the practices at project and organizational levels. In the Phantom postmortem, the management and the customer reported PIWs as one of the positive practices of Mobile-D. Likewise, the questionnaire filled in by the project stakeholders revealed that they either strongly or somewhat agreed (other options being neutral, somewhat disagree, and strongly disagree) on the claim that “PIWs were useful in finding improvements in software development practices during the project”. They also strongly or somewhat agreed that “it would be useful to carry out PIWs also in future agile projects”. However, both the project team and the management requested that in future PIWs the project team would need to be able to suggest action points iteratively directly to the organizational level also as some of the action points could not be implemented by the project team on its own. They might have required, for example, organizational participation or decision making. The management was willing to consider and implement such process changes already during the pilot project.

#### **4.4 Analyse, Improve and Package**

At F-Secure, the *organizational improvement* of the used agile practices was done immediately after the Phantom postmortem. The F-Secure management, Phantom

project team and its stakeholders participated in the organizational improvement workshop, which focused on elaborating the used agile practices for the organizational agile software development process. The external facilitator (of PIWs and the Phantom postmortem) was present to provide information on the SPI actions during the project as well as on the Mobile-D when needed. The agile assessor observed the workshop and gathered information for the ongoing agile assessment.

In the organizational improvement workshop, the recommendations of project stakeholders were collected and discussed on each Mobile-D phase. Prior to the workshop, the F-Secure management had made the necessary preparations and provided feedback from the PIWs and the Phantom postmortem. Thus, the sheet that was used for collecting the opinions of the workshop participants was pre-filled by the management to include the evident improvements that had already been identified (e.g. separated office space, and exclusion of the PP practice). Table 3 illustrates the organizational SPI decisions on QA practices made in the improvement workshop.

**Table 3.** Organizational Improvements on the QA Practices

Practice	Improvement	Cause	Origin
QA	Established collaboration of test and development teams	Lack of external test team activities in the used process	PIWs
	Iteratively updated documentation to support an external testing team	Unclear test focus due to lack of design documentation	Organization Improvement Workshop
	Daily wrap-up meetings	Development team in separate rooms	Postmortem
	Defined code review practices	PP excluded from the process	Postmortem

The *agile assessment* [5] was held after the first organizational improvement workshop. It was conducted by assessors who were familiar with the existing agile software development practices and with the agile assessment method. The goal was to *analyse* the suitability of the agile practices based on the feedback from the agile pilot projects and also from more traditional software development projects in order to evolve an F-Secure specific agile software development process. Two earlier projects had piloted the Scrum method while only the Phantom project used Mobile-D that included PIWs and postmortems providing agile assessments with validated process knowledge and improvement opportunities. In addition to the action points lists, reports and flapsheets of PIWs and postmortem, assessment data was collected using interviews, agile assessment workshops, and by observing the organizational improvement workshop. The available metrics (e.g. effort data) were utilized.

In the Scrum projects, agile assessment workshops were conducted to analyse the used agile practices together with the development team. The key problem, however, was the validity of the workshop results. The team members could not necessarily remember exactly what had happened in the project two or three months earlier during the project iterations. Instead, the validated PIW data (flap sheets, action point lists) provided new opportunities to analyse the advancement of the used agile practices (i.e., different solutions that had been experimented and evaluated) between the project iterations and to compare experiences of the different projects for finding the relevant agile based solutions for improving future software development processes. As an example, PP was one of the most problematic practices used in all the projects. In the Scrum projects, PP had been used in an unsystematic manner in complex coding situations. PP was also one of the most controversial practices in the Phantom project. The validated PIW data proved that PP was problematic throughout

the project. It was defined in the first PIW that PP should be used but only in complex tasks and for knowledge dissemination purposes. In spite of this, one of the key negative findings in the second and third PIWs was the use of PP in the project. Due to the resistance of a few persons, a decision was made not to use PP systematically in organizational practices at that point. The analysis of the postmortem data in the agile assessment, however, revealed that dropping PP from the software development process would demand additional QA practices such as code reviews.

## 5. Conclusions and Further Work

Currently, the agile software development methods provide an attractive alternative to the traditional plan-driven software development approaches. Specific procedures are, however, needed to support a systematic selection and deployment of new agile practices as well as for tailoring them to suit individual organizations. Thus, this paper proposes an agile deployment framework for software development organizations, designed for deploying and adapting agile practices in an iterative and agile specific manner. The framework puts emphasis on how the deployment can be carried out in the iterative life cycle of agile software development and how it integrates with the continuous improvement of organizational practices.

In this paper, the empirical results from a case study are presented in order to illustrate how an agile development method (Mobile-D) was deployed in a pilot project in F-Secure Corporation. The organizational goal was to utilize the experiences from the pilot project in establishing organizational agile process. The pilot project applied a post-iteration workshop method [6] (i.e. PIWs) for iterative *adaptation and improvement* of agile practices. Some more traditional mechanisms were also used for collecting the experience based feedback from the project for the needs of the organization (i.e., project postmortems). In addition, agile assessment [5] was conducted, utilizing the validated knowledge from PIWs.

The key point of this paper is to empirically evaluate the efficiency of PIWs in agile SPI, and the usefulness of systematically collected and validated PIW results in agile assessments. Furthermore, it is defined how these two agile specific SPI methods can be used to build an agile deployment framework, i.e. compatible and appropriate mechanisms for adopting and adapting agile methods, which also provide for continuous SPI in software development organizations. The qualitative results of deploying the different agile methods and practices of Mobile-D in the case project, however, are organization specific and not generalizable without further empirical evidence. Thus, the focus of this paper is on describing how the agile deployment was conducted in an industrial environment as suggested by the agile deployment framework and not so much on any detailed analysis of the qualitative findings of different agile practices adopted in the case organization.

The empirical evidence from the case study illustrates how the case organization was able to employ and benefit from the deployment mechanisms suggested in the agile deployment framework. Both the customer and the project team found the PIW method a useful mechanism in iteratively improving the daily working practices. The management also found the iterative and validated feedback from PIWs as well as the

results of agile assessment useful in monitoring the deployment process and evolving an organization specific agile process model alongside with their plan-driven product development process. However, in future projects both the software developers and the management would like to increase the on-time collaboration of project team and management already during the ongoing projects. This would allow the process improvements that the project team finds useful but can not implement all by itself to be experimented already in the ongoing project.

The agile deployment framework, as a whole, is primarily designed for the *iterative* software development model. Thus, it does not directly support the changing of process model type from traditional to agile. Yet, some of its individual methods, such as agile assessment, can also be applied in the traditional mode of development.

**Acknowledgement** to all the employees of VTT and the F-Secure Corporation who have participated in the Phantom project. The research was conducted within the Agile ITEA project funded by the National Technology Agency of Finland (TEKES).

## References

1. M. Cohn and D. Ford, "Introducing an Agile Process to an Organization," *IEEE Computer Society*, pp. 74-78, 2003.
2. H. Svensson and M. Höst, "Introducing an Agile Process in a Software Maintenance and Evolution Organization," 9<sup>th</sup> European Conference on Software Maintenance and Reengineering, 2005.
3. V. R. Basili, "Software Development: A Paradigm for the Future," COMPSAC '89, Orlando, 1989.
4. V. R. Basili, "The Goal Question Metric Approach," in *Encyclopedia of Software Engineering*, vol. 2: John Wiley & Sons, Inc., 1994, pp. 528-532.
5. M. Pikkariainen and U. Passoja, "An Approach for Assessing Suitability of Agile Solutions: A Case Study," 6<sup>th</sup> International Conference on Extreme Programming and Agile Processes in Software Engineering, Sheffield University, UK, 2005.
6. O. Salo, "Improving Software Process in Agile Software Development Projects: Results from Two XP Case Studies," EUROMICRO 2004, Rennes, France, 2004.
7. P. Abrahamsson, A. Hanhineva, et al., "Mobile-D: An Agile Approach for Mobile Application Development," 19th Annual ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'04), Vancouver, British Columbia, Canada, 2004.
8. V. R. Basili and D. Weiss, "A Methodology for Collecting Valid Software Engineering Data," *IEEE Transactions on Software Engineering*, vol. SE-10, pp. 728-738, 1984.
9. V. R. Basili and G. Caldiera, "Improve Software Quality by Reusing Knowledge and Experience," *Sloan Management Review*, pp. 55-64, 1995.
10. O. Salo and P. Abrahamsson, "A Post-Iteration Workshop Approach for Agile Software Process Improvement: Implications from a Multiple Case Study," *Under Review*, 2005.
11. A. Cockburn, *Crystal Clear: a Human-Powered Methodology for Small Teams*: Addison-Wesley, 2005.
12. T. Dingsøy, Moe, N.B., Nytrø, Ø. "Augmenting Experience Reports with Lightweight Postmortem Reviews," 3rd Int'l Conference on Product Focused Software Improvement (Profes 01), Kaiserslautern, Germany, 2001.
13. O. Salo, "Systematical Validation of Learning in Agile Software Development Environment," 7th International Workshop on Learning Software Organizations, Kaiserslautern, Germany, 2005.
14. N. L. Kerth, *Project Retrospectives: A Handbook for Team Reviews*: Dorset House Publishing, 2001.
15. O. Salo, K. Kolehmainen, et al., "Self-Adaptability of Agile Software Processes: A Case Study on Post-Iteration Workshops," 5th International Conference on Extreme Programming and Agile Processes in Software Engineering (XP 2004), Garmisch-Partenkirchen, Germany, 2004.
16. J. B. Cunningham, "Case study principles for different types of cases," *Quality and quantity*, vol. 31, pp. 401-423, 1997.
17. F. Lau, "Toward a framework for action research in information systems studies," *Information, Technology & People*, vol. 12, pp. 148-175, 1999.
18. K. Beck, *Extreme Programming Explained: Embrace Change*: Addison Wesley Longman, Inc., 2000.