Improving Software Process in Agile Software Development Projects: Results from Two XP Case Studies

Outi Salo
VTT Technical Research Centre of Finland
P.O. Box 1100, FIN-90571 Oulu, Finland
Outi.Salo@vtt.fi

Abstract

One of the Agile principles is that software development teams should regularly reflect on how to improve their practices to become more effective. Some systematic approaches have been proposed on how to conduct such a self-reflection process, but little empirical evidence yet exits. In this paper, the empirical results are reported from two XP (Extreme Programming) projects where the project teams conducted "post-iteration workshops" after all process iterations in order to improve and optimize working methods. Both qualitative and quantitative data from the total of eight post-iteration workshops is presented in order to evaluate and compare the findings of the two projects. The results show the decline of both positive and negative findings, as well as the narrower variation of negative findings and process improvement actions towards the end of both projects. In both projects, the data from post-iteration workshops indicate increased satisfaction and learning of project teams.

1. Introduction

AGILE methodologies and principles[1] place emphasis on incremental software development with short iterations, adaptation to changing requirements, close communication, and simplicity, for example. One specific agile principle closely relates to the software process improvement (SPI): "regular reflection of teams in how and where to improve". Furthermore, agile proponents have noted that "each situation calls for a different methodology" [1, p. 184]. Thus, when using any of the agile approaches, continuous improvement, tuning and adjusting of the software development process is required.

Although individuals and interactions are placed over processes and tools in the Agile Manifesto[2], at least Extreme Programming (XP) claims to be a disciplined process [2] and may actually be characterized as such from a Capability Maturity Model for Software (The Software CMM) [3] viewpoint, for example.

In the CMMI staged model, only the reaching of maturity level 2 (Managed) [4] includes, implementation of PPQA process area (Process and Product Quality Assurance), amongst six other process areas. For example, it includes the evaluation of performed processes and identification of lessons learned that could improve processes. In XP, some references can be found that display such activities. For example, in the death phase of XP it is instructed to "Imagine with the team how they would run things differently next time" [5, p. 137]. However, the XP practices [5], do not include detailed procedures on how to actually carry out such activities to improve the software development process.

Recently, some systematic approaches have been proposed on how to improve the software development process in agile context for an individual project. Cockburn suggests a methodology-growing technique for "on-the-fly methodology construction and tuning" [1, p. 185] that embodies a reflection workshop technique for the mid- or post-increment workshops. Also, Dingsøyr and Hanssen [6] have suggested a workshop technique called postmortem reviews to be used as an extension for agile software development methods. It pursues on making good use of the experiences of project participants at the end of the short iterations to enhance the development process and also over the project boundaries.

However, only a very limited amount of empirical evidence can be found on applying team reflection

[1] www.agilealliance.org

[2] www.agilemanifesto.org

workshops, lightweight postmortem reviews [6] or any other SPI efforts from agile software development projects. This paper presents a comparison of empirical results from two case studies (eXpert and zOmbie) conducted at the Technical Research Centre of Finland. Two consecutive projects adopted XP and conducted post-iteration workshops systematically after each iteration to improve their software development process. The post-iteration workshops included elements from both the lightweight postmortem review technique [6] and the methodology-growing technique [1] and focus on the project level SPI.

This paper presents the analysis of the post-iteration workshop data (i.e. number and content of negative and positive findings, and number of SPI actions) from the two case studies. The aim is to present the consistencies and deviations in the data of two somewhat similar, yet also divergent projects, and the underlying causes for such findings are also suggested. Thus, in this paper, the findings of the post-iteration workshops are discussed in the light of the different characteristics of the two case projects. Also, one goal is also to either support or evert the early conclusions previously drawn based on the eXpert case study [7].

The focus of this paper is to analyze the quantitative and qualitative findings generated during the post-iteration workshops and the quantity of the implemented process improvement actions. However, these two case studies do not offer extensive enough data for firm generalizations, but some conclusions can be brought forward for further evaluation.

This paper is composed as follows. The next section presents the research design including the method, the research target and settings, with the presentation of the similar and divergent characteristics of the two case studies. The paper continues by presenting the results and analysis of post-iteration workshops, and ends with a discussion and acknowledgements.

2. Research Design

In this section, the research method, data collection, and the research setting are described for both the eXpert and zOmbie projects.

2.1. Research Method and Data Collection

The research method used in this study was action research [8] that can be seen as one form of case study. The focus in action research is more on what practitioners do rather than what they say they do [9]. The resulting knowledge should guide the practice [10]. In action research, the modification of reality requires the possibility to intervene [11]. In the post-iteration workshops the researchers' acted in a role of a moderator and participated in the generation of positive and negative findings, and enhanced the process with the project team. Also, one role for researchers' was to provide the boundaries in which the project team was allowed to enhance the process.

In both projects, quantitative and qualitative research data was collected on a) effort used on workshops, b) quantity of findings and c) their content and, d) quantity of suggested and actual process enhancements (i.e. action points) and e) their content. Furthermore, the developers maintained diaries to record their negative and positive perceptions. Also, a post-project workshop and a group interview were held for the project team at the end of both projects.

2.2. Research Target: Post-Iteration Workshop technique

This research aims to study how a short iterative reflection session suits for self-adapting and improving the practices during an Agile software development project. Thus, the focus is SPI on project level.

The post-iteration workshop technique applied in both of the case studies was evolved by combining attributes from both of the existing reflection techniques (i.e., lightweight postmortem review and team reflection workshop techniques) as described in more detail in [7]. In short, as suggested in postmortem review technique, the problem-solving brainstorming method called the KJ method [12] was adopted in the post-iteration workshops for *generating, collecting and structuring positive and negative experiences*. The project team recorded positive and negative issues concerning the previous iteration on post-it notes. These notes were then grouped, and negative issues were discussed to generate process enhancements.

Both existing techniques suggested prioritizing the negative findings and analyzing only the most important ones. However, in post-iteration workshops all the negative findings were considered as equal and all of them were included for further discussion. Then, the actual software process improvement actions (hereafter referred as SPI actions) were decided together with the project team and researchers. This data was collected in action point lists by the project team member during the post-iteration workshops. Finally, the previous action point list was revised to find out what improvements had actually taken place and which ones were not implemented for one reason or another.

The quantitative as well as qualitative data from the post-iteration workshops is the central research data presented in this paper. This includes the positive and negative findings, and the implemented SPI actions.

The results from the first case study (eXpert) were earlier reported in [7]. It was suggested that post-iteration workshops concretely help to improve and optimize practices, and enhance the learning and satisfaction of the project team. This argument is evaluated in this paper by strengthening the case with the comparative analysis of eXpert and zOmbie case studies. Another target of this research is to seek out any consistencies and deviations between the two projects, and to find underlying factors behind them.

2.3. Research setting

The two case studies presented in this paper are the first ones in the ongoing series of Agile software development case studies at VTT Electronics. As this paper presents results from two case studies, i.e. eXpert and zOmbie, their characteristics need to be addressed in order to offer a framework for the interpretation of results. Thus, the similarities and divergences of the two projects are described in this sub-section.

Similarities of eXpert and zOmbie case studies

Both case studies were conducted in a co-located development environment. In fact, the projects worked in exactly the same open office space. Also, the common tools that were not dependent on the developed application type were identical in the two projects. These included configuration management, data collection, and documentation tools.

Intensive two-day training was given to both of the teams including XP practices, configuration management and data collection issues. The teams were advised to follow XP process as suggested by Beck [5] including planning game, small releases, metaphor, simple design, testing practices, refactoring, pair programming, collective ownership, continuous integration, 40-hour week, and coding standards. However, also other practices, such as SPI activities, were employed to support software development.

The two projects had an identical calendar time (nine weeks) and length of working week (4-day week of 24-hours). As proposed by the 40-hour week rule, no overtime was recommended. The possible overtime was compensated in the following iteration.

Differences of eXpert and zOmbie case studies

Table 1. presents the central differences between the eXpert and zOmbie projects that should be taken into consideration when interpreting the data from post-iteration workshops.

Table 1. Central differences of eXpert and zOmbie

Characteristic	eXpert	zOmbie
Team size	4 developers	4 developers 1 project manager
Type of end product	Intranet application	Mobile application
Experience in XP/Agile	4 novice	1 experienced 4 novice
Experience in the end product development	1 experienced 3 novice	5 novice
Experience in coding	2 experienced 2 novice	4 experienced 1 novice
Iterations	3 x two weeks 3 x one week	1 x one week 3 x two weeks 2 x one week
Size of end product	10 000 LOC	7 000 LOC
Software development tools	Eclipse Apache Tomcat MySQL Java +JSP	Eclipse Apache Tomcat MySQL J2ME
XP practices	On-site customer	Off-site customer

Firstly, the team size was slightly different. Both project teams included 4 software developers all being university students at the final stage of their information processing science studies. However, zOmbie employed also a project manager from the previous eXpert team to provide expertise on the Agile development approach and XP. The project manager worked a shorter week than the rest of the project team, i.e. 2/3 of their effort. Despite this, he participated in all the post-iteration workshops and thus is to be noticed in the quantity of the findings.

Also, the end product being developed was divergent in the two projects. The eXpert team implemented an intranet application for managing research data of a Finnish research institute. The zOmbie team had a task to implement a mobile application for managing transactions in a stock exchange. Naturally, the distinct application types also caused some differences in the software development tools and languages (see table 1). Furthermore, the size of the end product varied from the eXpert's 10000 LOC to zOmbie's 7000 LOC.

Furthermore, experiences of the team members differed between the two projects. In eXpert, all the team was novice on using agile software development methods. The zOmbie project had the advantage that the project manager was one of the developers from the previous eXpert project. As such, he was a valuable inboard "coach" [5] for the zOmbie team and also may have influenced some of the project practices based on his experiences from eXpert. However, as mentioned earlier, all the actual SPI decisions were to be made solely in the post-iteration workshops.

The experiences of project teams varied in the end product development, as well as coding skills. In eXpert, one team member was experienced on the development of intranet applications, whereas the zOmbie team totally lacked knowledge on mobile software development. Vice versa, two of the four eXpert team members were experienced coders but in zOmbie, three of the developers plus the project manager could be regarded as experienced coders, and only one novice. In this context, novice on coding is regarded as "no industrial experience". Thus, zOmbie project had more advanced coders, but their experience in the end product development was lacking.

The length of iterations was different in the two projects. Both projects consisted of six iterations. In eXpert, the project started with three two week iterations and finished with three one week iterations, the last one being a corrective iteration. In zOmbie, the first and last two iterations lasted one week and the 2nd, 3rd, and 4th iterations for two weeks each. As this paper involves analysis on the first four iterations of both projects, the difference in the length of first and fourth iterations should be noted in interpretations.

Probably the biggest difference strictly related to XP practices was the role of the customer. The eXpert project included an on-site customer working in the same office space with the software developers, as suggested by XP [5]. In the zOmbie project, however, one of the research targets was to study how an on off-site customer would suit an XP project. This matter is out of the scope of this paper, but it is clearly one of the factors to be noted when comparing the findings of the case studies reported in this paper.

3. Case Study Results

In this section, the analysis of the post-iteration workshop findings is presented with interpretations. Only the first four post-iteration workshops are included in this analysis. The fifth workshops in both projects concentrated on the experiences from the entire project instead on the previous iteration and, thus, are not comparable. However, these post-project workshops are valuable from the viewpoint of organizational level SPI and will be reported elsewhere in the near future.

3.1. Post-Iteration Workshop Findings

Table 1 presents the costs of holding post-iteration workshops in terms of duration per workshop and the percentual effort spent on the workshops calculated from the total iteration effort.

Table 2. Cost of post-iteration workshops

	Duration		**Effort %**	
Iteration	**eXpert**	**zOmbie**	**eXpert**	**zOmbie**
1	2.68 h	2.18 h	5,5 %	7,8 %
2	1.83 h	2.35 h	3,8 %	5,5 %
3	1.0 h	1.63 h	2,1 %	3,0 %
4	0.93 h	1.13 h	3,3 %	2,2 %
Avg	**1.6 h**	**1.82 h**	**3.7 %**	**4.1 %**

Results show the tendency of the duration, as well as the percentual effort lowering from iteration to iteration in both of the projects. To explain the seemingly high effort percentage in both cases, several explanatory factors can be found. First, it should be noted that each software developer worked a 24-hour week instead of 'normal' 40-hour week. In the latter case, the corresponding figures would be substantially lower. However, this requires an assumption that a 40-hour week does not increase the duration of post-iteration workshops. Also, in the zOmbie project, the project manager worked about a 16 hour-week, which also increases the percentual effort spent on post-iteration workshops. Secondly, it should be noted that in both cases the shorter iteration (4th in eXpert and 1st in zOmbie) causes the proportion of effort to rise even though the actual effort spent may even be lower.

In Dingsøyr and Hanssen's [6] study the effort spent on lightweight postmortem reviews was around 4.7% and the duration of one workshop was roughly 1.4 hours (calculated from their data). Cockburn [1] estimates a minimal duration from two to four hours. In these two cases the average effort percentage was 3.7% in eXpert and 4.1 % in zOmbie whereas the average duration in eXpert was 1.6 hours and 1.8 hours in zOmbie. One interesting observation from table 2 is that in both projects the duration of the workshop has halved from the 2nd iteration to the 4th iteration.

Furthermore, it can be presumed that the learning of the post-iteration workshop technique took some time during the first workshops and also affects the decline trend in effort and duration data. However, it can be seen in Table 2, that the duration of all but the first workshop was longer in the zOmbie project compared to eXpert. One reason for this is the clearly larger amount of negative findings (Figure 2), as well as the topics behind these findings (Figure 3). Thus, the discussion and decision-making during the post-iteration workshops obviously took more time in zOmbie than in eXpert. The long duration of the first post-iteration workshop in eXpert project can be explained by the fact that the technique was applied for the first time and, thus, took some time for the moderator (i.e., researcher) to learn as well.

Quantitative data from the four post-iteration workshops is presented in the figures 1 and 2.

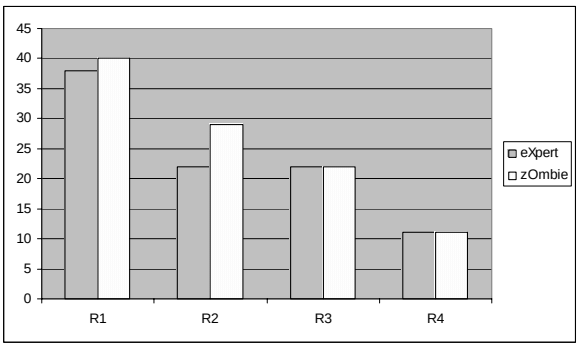

Fig. 1. Number of positive findings from eXpert and zOmbie post-iteration workshops

The first four iterations of eXpert produced a total of 93 positive findings whereas the corresponding number in zOmbie was 102 (Figure 1). This is a total of 23.3 positive findings per person in eXpert and 20.4 in zOmbie. In both cases the trend seems to be a decline in positive findings towards the end of the project.

One reason for this is the frequent occurrence of post-iteration workshops (one to two weeks apart). Thus, the project team may not always find it necessary to repeat neither the positive nor negative findings - even though they still may be valid. In fact, one comment made by a software developer during the 3rd post-iteration workshop in zOmbie was: "The charm of novelty is gone. Trifles don't make one so happy anymore". At the time, he found it hard to think of any (new) positive findings. Thus, as the team gets more accustomed to the adopted practices, their weaknesses and rewards may be taken for granted.

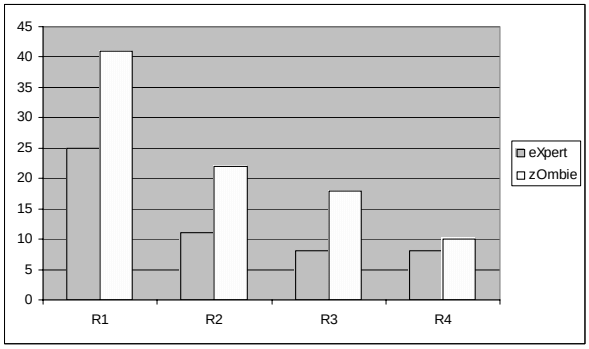

Fig. 2. Number of negative findings from eXpert and zOmbie post-iteration workshops

The first four iterations of eXpert produced a total of 52 negative findings. The corresponding number in zOmbie was 91 (Figure 2). It should be noted, that the zOmbie team had one "extra" team member yet the number of negative findings per person is still noticeably higher (13 in eXpert and 18.2 in zOmbie).

Also, the negative findings in both projects seem to follow the declining trend as in positive findings. This supports the argument of Cockburn [1], that the changes needed in the process will be much smaller after the second and subsequent increments. Also, the trend lines in both positive and negative findings indicate that the duration of the iteration does not affect the amount of findings that are generated in post-iteration workshops.

The closer examination of the research data reveals that as the topics causing negative findings became fewer during both projects they also drew closer to each other (Figure 3). In other words, the criticism of the project team became more focused.

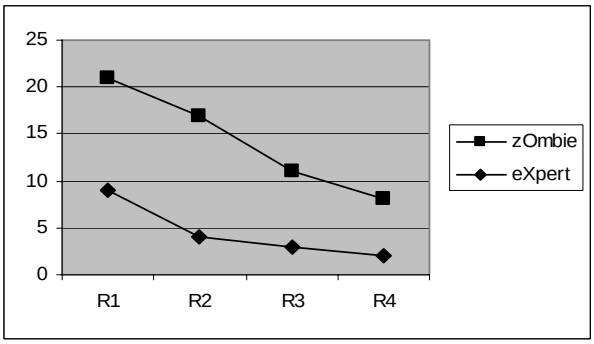

Fig. 3. Number of topics behind negative findings

The above zOmbie data clearly supports the earlier reported eXpert case study results [7]. The declining trends of negative findings and topics behind them (Figure 3) support the argument that the process actually adapted to the needs of the project team, and increased their satisfaction for the process [7].

Table 3 illustrates the central topics behind the positive and negative findings in eXpert and zOmbie.

Table 3. Top five of the positive and negative findings

	Top 5 positive findings		**Top 5 negative findings**	
	eXpert	**zOmbie**	**eXpert**	**zOmbie**
1	Pair programming	Team spirit	Time tracking	Testing
2	Short iterations	Working environment	Testing	Time/task estimation
3	Continuous integration	Technical environment	Code commenting	Technical environment
4	On-site customer	Planning game	Effort estimation	Time tracking
5	Refactoring	Pair programming	Test-first development	Off-site customer

Interestingly, all the top five positive topics in eXpert focus on the practices of XP. In zOmbie, the

positive findings concentrated more on human and environmental aspects.

The closer examination of the negative findings in the two projects discloses some project specific problem areas especially regarding the zOmbie project. One clearly emerging issue is the off-site customer that was applied in only the zOmbie project. 20% of the negative findings in the first iteration related to the off-site customer that was too busy. Changing the off-site customer - which often may be impossible - solved much of the problem in this case. Also, the communication practices with customers were improved throughout the project. Another clearly project specific problem area in the zOmbie project was testing in the mobile software development environment. 12% of the 92 negative findings throughout the project related to this problem. Specifically, this problem related to testing (test-first) in the client side (i.e., simulator) environment and was not solved during the project. Also, the set-up of the technical environment was found more complicated in mobile software than in intranet application development as it required fire wall configurations and setting-up of IP server, for example. In zOmbie, 8.7% of the negative findings related to environmental problems where as this topic resulted zero findings in eXpert.

Common in negative findings for both projects was the lack of clear exit-criteria for tasks, i.e. criteria to verify if the task is actually done. Also, data collection was found problematic. Due to the research character of the project, the collection of measurement data was heavy and time tracking detailed. However, both of these findings appeared strongly after the first iteration and sharply lowered towards the end of the project. Also, the importance of coding standards as well as their proper use was an issue that came up in both of the projects. Furthermore, the estimation of tasks was found as clearly problematic in both of the projects. In zOmbie, 15% of all the negative findings related to this topic and 17% in eXpert.

Two topics that were reported in only the eXpert project were test-first and short iterations. The test-first approach [13] was clearly problematic due to the fact that there was no expert available on this approach. In the latter project, however, the project manager was a member of the previous eXpert team and thus had some experience and knowledge on this topic. In eXpert, the negative findings related to testing refer to the decrease of motivation of the external testing team that reflected on the development team as lack of testing results and feedback.

Figure 4 presents the number of actual process improvements that were carried on after iterations.

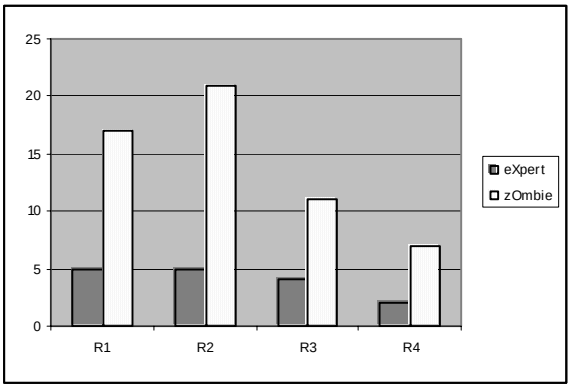

Fig. 4. Quantity of implemented software process improvements

As it can be seen, the post-iteration workshops of the zOmbie project implemented 56 improvement actions whereas eXpert only 16. The thorough analysis of improvement actions and their effect is out of the scope of this paper. One reason for this is that the existing project level SPI techniques lack a detailed procedure for follow-up of SPI actions as well as their support with, for example, measurement data. However, quantitative data of SPI actions is interesting in order to evaluate how the number of negative findings in each workshop relates to the number of process improvements (Figure 5).

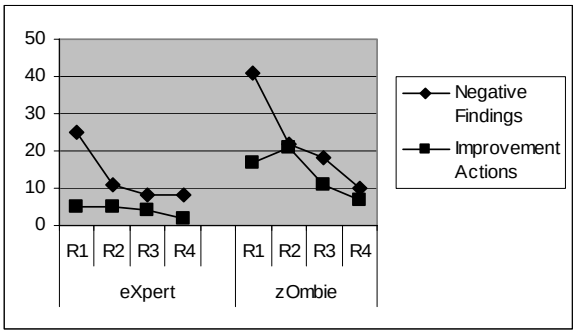

Fig. 5. Negative findings vs. quantity of implemented software process improvements

Figure 5 illustrates how the variation between the amount of negative findings and process improvements actions is widest after the first iteration in both of the case studies. For one, this data indicates that the project teams were cautious on making any process changes at the beginning of the project. One reason for this was the novelty of the methods and techniques used which made it impossible to evaluate whether the negativity was caused by the method itself or the lack of ability of the project team to apply it. For example,

in eXpert, the test-first approach caused negative findings during the first two iterations but they actually turned into positive findings towards the end of the project as the project team learned to apply the technique.

Learning was something that also took place during the post-iteration workshops. For example, some of the negative findings could be identified as misunderstandings or problems in communication. These issues needed no specific actions but were solved by discussion within the team. Also, it can be interpreted from Figure 5 that the implemented SPI actions influence in decreasing the amount of negative findings after the following iteration. This, again, implicates to the increased satisfaction of the project team to the enhanced process. For example, as the data collection tools were improved both in eXpert and zOmbie, the negative findings this topic were dried up. On the contrary, the SPI actions, though relatively small at times, seemed to produce positive findings on even the annoying topics such as data collection.

4. Conclusions and Further Work

Agile principles suggest that the software development team should regularly reflect on how to become more effective and tune and adjust its behaviour accordingly. Some systematic approaches have been proposed on how to execute this self-reflection process effectively but little empirical evidence yet exits. This paper presents a comparison of empirical results of two case studies where two known self-reflection approaches were combined [1, 6] and four post-iteration workshops were held in two XP projects.

The goal was to study how the post-iteration workshop results from two affinite, yet divergent projects vary in order to strengthen the conclusions of earlier reported eXpert results [7], and to broaden the study to find coherences and deviations from the research data. The data includes the quantity and quality of positive and negative findings from the post-iteration workshops as well as quantity of the actual SPI actions made based on the findings. Though these two case studies do not offer extensive enough data to draw any generalizations, some conclusions can be brought forward for further evaluation.

Firstly, several consistencies could be found on the data of the two projects. For one, the amount of both positive and negative findings decreased towards the end of the projects quite rapidly. This indicates the customization of the project team to the new tools and practices and especially the decrease in negative findings refer to increased satisfaction of the project team towards the end of the project. In other words, the findings support the assumption that post-iteration workshops were effective on improving the process to suit the development team. Secondly, the correlation between the number of negative findings and the number of SPI actions clearly drew closer towards the end of the projects as the amount of needed improvements clearly lowered. This data also speaks for the successful adaptation of the software process and the effectiveness of post-iteration workshops.

Thirdly, the data from both projects show that the effort needed on post-iteration workshops clearly decreases from iteration to iteration. This somewhat indicates the learning of using the technique, but it also correlates to the increased satisfaction of the project team. The lower amount of negative findings in consecutive iterations shortened the time spent on discussion and decision-making concerning the SPI actions for next iteration.

Also some deviations could be found when comparing the research data from the two projects. The zOmbie case study produced clearly a larger amount of negative post-iteration workshop findings. The underlying causes for this were found in several factors. One of these was the complexity of a project. In the mobile software development project (zOmbie) few clearly complex factors, such as environment setup and testing in mobile device were found to increase the amount of negative findings compared to eXpert. Also, one clear factor to increase the amount of negative findings was the off-site customer in zOmbie. Naturally, the larger amount of negative findings can also be seen in the longer duration of workshops in zOmbie. In other words, factors such as the complexity of project and suitability of the used software process for the specific team effect on the time spent on post-iteration workshops and the amount of changes needed in the process.

The effort used on post-iteration workshops decreases towards the end of the project in both case studies. It could be calculated to be as low as 2.2% (4th zOmbie workshop). When taking into consideration the shorter working week (i.e., 24 hours per week), the effort needed on post-iteration workshops is quite tolerable especially when considering the immeasurable value of increased satisfaction and learning of project team. Still, the effectiveness of post-iteration workshops in regard to effort and duration is something that should be further increased especially during the first iterations. On the research data presented, some normative base can be found to estimate how much effort should be allocated in organizations for holding post-iteration workshops.

According to the software developers of eXpert and zOmbie, the rapid visibility of the SPI actions and the

concrete possibility to influence the working practices increase the satisfaction of the project team. These strong implications of the benefit of the post-iteration workshops were found in the positive remarks made by both development teams in the final interviews.

This paper does not report the quality of the actual SPI actions made during the two case studies nor their effect on, for example, the quality of the end product. One reason for this is that the existing project level SPI techniques lack a detailed procedure for the follow-up of SPI actions, as well as their support with, for example, measurement data. Also, the existing techniques lack important aspects in enhancing the extensive learning in the future projects. Based on this observation, the post-iteration workshop technique has been evolved and is currently being applied for further evaluation in the two following case studies (bAmbie and uniCorn) at the VTT Technical Research Centre of Finland.

Overall, based on the data presented in this paper, the second case study (zOmbie) is in line with the results of eXpert and, thus supports the early conclusions presented in [7]. Accordingly, the iterative workshop gathering is a concrete way to improve and adapt Agile software processes during the iterative cycles of software development. Thus, post-iteration workshops should be regarded as a useful method to be included in Agile software development projects, especially if supplemented with follow-up and validation of process improvements.

5. Acknowledgements

The research work presented in this paper has been carried on in the ICAROS project funded by TEKES (National Technology Agency of Finland). Acknowledgement also to the eXpert and zOmbie teams for their thorough participation in the process improvement activities. Also, thanks to Dr. Pekka Abrahamsson for his co-operation and support, and Annukka Mäntyniemi for her valuable comments.

6. References

[1] **A. Cockburn, *Agile Software Development.* Boston: Addison-Wesley, 2002.**

[2] **K. Beck and B. Boehm, *Agility through Discipline: A Debate.* Computer, pp. 8, 2003.**

[3] **M. C. Paulk, *Extreme Programming from a CMM Perspective*. Software, vol. 18, issue 6, pp. 19-26, 2001.**

[4] **S. E. I. (SEI), CMMISM SE/SW/IPPD/SS Model, Version 1.1, Staged Representation (Technical Report CMU/SEI-2002-TR-012), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 2002.**

[5] **K. Beck, *Extreme Programming Explained: Embrace Change*. Addison Wesley Longman, Inc., 2000.**

[6] **T. Dingsøyr and G. K. Hanssen, *Extending Agile Methods: Postmortem Reviews as Extended Feedback*. 4th International Workshop on Learning Software Organizations (LSO'02)), Chicago, Illinois, USA, 2002.**

[7] **O. Salo, K. Kolehmainen, P. Kyllönen, J. Löthman, S. Salmijärvi, and P. Abrahamsson, *Self-Adaptability of Agile Software Processes: A Case Study on Post-Iteration Workshops*. To be published in the proceedings of the 5th International Conference on Extreme Programming and Agile Processes in Software Engineering (XP 2004), Garmisch-Partenkirchen, Germany, 2004.**

[8] **J. B. Cunningham, *Case study principles for different types of cases*. Quality and quantity, vol. 31, pp. 401-423, 1997.**

[9] **D. Avison, F. Lau, M. Myers, and P. A. Nielsen, *Action Research*. Communications of the ACM, vol. 42, pp. 94-97, 1999.**

[10] **P. Oquist, *The epistemology of action research*. Acta Sociologica, vol. 21, pp. 143-163, 1978.**

[11] **G. I. Susman and R. D. Evered, *An Assessment of the Scientific Merits of Action Research*. Administrative Science Quarterly, vol. 23, pp. 582-603, 1978.**

[12] **R. Scupin, *The KJ Method: A Technique for Analyzing Data Derived from Japanese Ethnology*. Human Organization, vol. 56, pp. 233-237, 1997.**

[13] **K. Beck, *Test-Driven Development By Example*. 1st ed: Addison-Wesley, 2003.**