Publication II

Jari Vanhanen, Casper Lassenius, and Mika V. Mäntylä. 2007. Issues and tactics when adopting pair programming: A longitudinal case study. In: Bob Werner (editor). Proceedings of the Second International Conference on Software Engineering Advances (ICSEA 2007). Cap Esterel, France. 25-31 August 2007. Los Alamitos, California, USA. IEEE Computer Society. 70, 7 pages. ISBN 978-0-7695-2937-0.

Issues and Tactics when Adopting Pair Programming: A Longitudinal Case Study

Jari Vanhanen, Casper Lassenius and Mika V. Mäntylä
Helsinki University of Technology, Software Business and Engineering Institute
P.O. BOX 9210, 02015 TKK, Finland
{firstname.lastname}@tkk.fi

Abstract

We present experiences from a two-year study of adopting pair programming (PP) in a Finnish software product company. When adopting PP, the company used five tactics: the creation of simple PP guidelines, the use of a PP champion, making the use of PP voluntary, creating a positive atmosphere for PP, and instituting a separate PP room. By the end of the study the feelings of PP considerably surpassed developers' preconceptions of PP, and even the feelings of solo programming. Issues identified in the infrastructure for PP were solved through the adoption of the PP room. In the end of the study, a majority of the developers thought that PP should be utilized more than the reached ca. 10% of development effort. Unresolved issues in resourcing PP probably hindered reaching the desired level for the use of PP.

1. Introduction

In pair programming (PP) two persons design, code and test software together at one computer actively communicating with each other [1]. The driver controls the keyboard, and the navigator observes the work, thinking at a more strategic level [1]. PP can have a positive effect on, e.g., quality of code and design, knowledge transfer, learning, team work, and work satisfaction [2,3,4]. The negative effects can be increased effort especially on the level of individual tasks [2,3,5,6,7] and mental exhaustiveness of work [1,8].

PP has become well-known since the popular agile development methodology extreme programming (XP) advocates its use for all development work [9]. Such an extreme approach may be difficult to adopt, but nothing hinders using PP only for those tasks for which it is found most beneficial. Actually, the only studies of time spent with a pair in XP projects report figures as low as 30% [8] and 50% [10].

Various problems and solutions on adopting PP in industry have been reported [7,8,10,11,12,13]. Many details need consideration when applying PP in industry, and the degrees of freedom even increase if the XP style use of PP "for everything by everyone" is not applied. The current lack of understanding of the various details on practicing PP [14] may hinder the successful adoption of PP or decrease its usefulness, thus meriting further studies of the subject.

We report on the adoption of PP in an organization during a two-year time span. We discuss the progress of the adoption, challenges faced, and some solutions to the challenges. This case study was mainly based on four repeated surveys, but we also observed PP sessions and discussed with the employees of the organization about their use of PP. Chapter 2 summarizes the previously published experiences of adopting PP from the industry. Chapter 3 presents the research methodology and introduces our case organization. Chapter 4 presents our results, and Chapter 5 concludes the paper.

2. Related work

Williams and Kessler propose that most people resist transitioning to PP, but after trying it decide it is better than working alone [1]. They mention that it may take a couple of weeks before PP becomes efficient. They recommend having the following things in place when adopting PP: individual freedom to decide on applying PP, some level of consistent organizational standards and procedures for PP, and management support without enforcement. They also remind of the need for changing the values of the organization away from individual and toward team recognition and goals.

Studies from industry report issues that have slowed down the adoption of PP and some solutions to these issues. The use of PP may suffer if the expected effort increase is not included in the effort estimates [12] or there are deadline pressures [13]. Developers may not be able to work with everyone [7], and an organization's culture may prevent increases in the use of PP [10]. Successful ways for increasing the use of PP have included explicitly reserving time for PP [13], persuasion by management [12], keeping two developers responsible for the quality of a task [12], rewarding use of PP by avoiding formal reviews [11], and dropping the ownership of work stations [7].

Williams and Kessler [1] give the following suggestions on good infrastructure for PP. It is necessary to be

International Conference on Software Engineering Advances(ICSEA 2007)
0-7695-2937-2/07 $25.00 © 2007 IEEE

able to sit comfortably side-by-side with a fairly large monitor between the partners. A six-by-three foot table is ok, and a convex table works very well. Many pair programmers prefer having two displays, keyboards and mice connected to the same computer. Wireless keyboards and mice could also be considered. Two displays can be used as a one ultra-wide display. Standardized development tools for all developers is ideal. Other pairs can easily ignore the noise generated by PP but it may disturb solo programmers.

Similar opinions about desks for PP have been reported by others [15,16]. However, there are no studies on the effects of infrastructure for PP or on the infrastructure used by pair programmers in industry.

3. Research method

We made an exploratory longitudinal case study [17] of PP in a department of a Finnish software product company. Below we describe the research goals and questions along with the data collection and analysis methods, and introduce the case organization.

3.1. Research goal and questions

Our research goal was *to observe the adoption of PP in a realistic, industrial context*. The focus was on the issues involved in adopting PP, and on tactics for solving them. We explicitly searched for issues and tactics related to the developers' attitudes to PP, infrastructure for PP such as computers and work place, and organizing for PP such as resourcing PP and pair formation. The research questions were:

1. What issues are faced during PP adoption?
2. What tactics can be used to solve the issues?

In addition, we studied the effects of PP on learning, quality, effort, and human factors. They results of that part of the study are reported in another paper and only briefly summarized in this paper [18].

3.2. Case description

The case organization was a department responsible for the development of a large and very successful software product, with a development history of over ten years. The development languages were C/C++, and the most used development environment was Visual Studio. The amount of developers in the case organization increased from about 25 to about 35 during the study. They were divided into four independent development teams, each having a senior developer as a team leader. All developers were sitting in the same large open office containing many cubicles and desks.

Before the study, some developers in the department had informally used a little PP, and one team had done a small pilot on PP with promising results. Therefore, they decided to spread the practice to all teams, the primary goals being to improve software quality and increase knowledge transfer among developers. They did not aim for an XP style full scale use of PP, but for using it only when it would be most beneficial. At this time, the developers' experience on PP varied a lot, if including PP done in their previous work and studies. Most developers had a few dozens of hours PP experience, a few had no or almost no PP experience, and a few had used it for hundreds of hours.

The team leader behind the piloting became the PP champion, who took the responsibility for the department's PP guidelines and for promoting PP in the whole department. The first author got involved after the piloting as an external observer whose main goal was to study the effects of PP, but he also co-operated in a small role in introducing PP to the developers, and in creating PP guidelines for them.

The company used several tactics to launch the adoption of PP. They had a PP champion, who introduced the initial guidelines for using PP in the department in order to distribute the experiences from the pilot and to increase its use. The guidelines included a tactic, i.e. making the use of PP voluntary. The guidelines are summarized below:

- PP should be used in particular when new developers join a team, when knowledge needs sharing, or when doing difficult or error-prone tasks.
- Teams should select the features that would benefit most from the use of PP. PP should be used in particular for the specification and design activities for these features. A pair should work together for 10–100% of a feature's development effort. Solo development should not be done more than 10h between two PP sessions
- Each feature still has only one responsible person.
- Developers select their partners themselves.
- PP sessions should last 0.5–3h, and be allocated in advance.

3.3. Data collection and analysis

The study started with informal discussions with the PP champion. The first author also made a semi-structured interview with the pilot team about how they had applied PP and what effects they had perceived. Thereafter, the PP champion and the first author introduced PP and the initial guidelines to the developers.

Our main data collection method was four surveys (S1–S4) made between 3/2005 and 11/2006. The first author developed the questionnaire together with the PP champion and the head of the department based on the research goals. We improved the questionnaire slightly for each survey based on what we had learned or wanted to know in more detail, and before each survey 1–2 developers from the target population tested it.

Questionnaires 1 and 2 contained mostly closed questions. The developers attitudes to PP were inquired using the question "*What are your feelings towards the following topics?*". The topics were "*PP*", "*PP before*

trying it", "*solo programming*", and "*acting as the more/less skillful person in a pair*". Two topics, "*organizing for PP*", and "*infrastructure for PP*", were added to survey 3. The answers were given on a 7-point ordinal scale: "*1-negative*", "*4-neutral*", "*7-positive*". Beginning with survey 3 we also asked "*What do you think about the amount of PP in your team during X?*", where X was 2005 for survey 3 and 2006 for survey 4. The answers were given on a 7-point ordinal scale: "*1-we should use less PP*", "*4-current amount ok*", "*7-we should use more PP*".

The results of survey 2 showed that use of PP increased slowly. Therefore, the first author observed three PP sessions in order to find out how PP really was applied. Based on the findings from the observations and discussions with the PP champion we added open questions to questionnaire 3. They focused on developers' recommendations on how PP should be used and on problems related to the organizing and infrastructure for PP. For questionnaire 4, we added questions about the recently introduced PP room.

The head of the department sent the questionnaires to all developers. Answering was voluntary. Developers answered by e-mail to the first author, and their identities were not revealed to the company. The first author presented research results after three surveys to the team leaders and twice to all employees. During these sessions the developers gave comments and feedback which helped to understand their answers better. Figure 1 shows the research timeline.

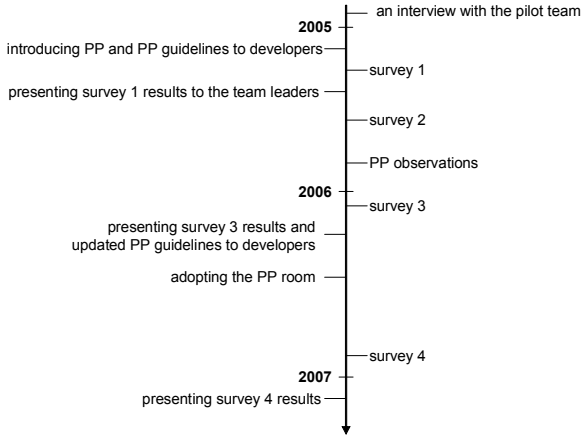

Figure 1. Research timeline

Thirty-two developers answered at least one survey, and eight of them answered all four surveys. The response rates are listed in Table 1. The statistical significance tests were done with the SPSS 14.0 software using the Mann-Whitney test due to the non-normal distribution and ordinal scale of the data. The chosen level of significance was 0.05. The qualitative data from each open question in the questionnaires was grouped thematically and synthesized.

Table 1. Response rates for the surveys

	S1	S2	S3	S4
Date	3/05	6/05	1/06	11/06
Respondents	19	14	21	21
Developers (total)	25	24	24	35
Response rate	76%	58%	88%	60%
Years in the company (median)	4.5	1.5[1]	4.0	3.0

[1]Many senior developers did not answer e.g. due to vacations.

4. Results

Below we present the results of our study. First we analyze the amount of PP. Then we present the results of developers' attitudes to PP, organizing for PP and infrastructure for PP. Finally, the experiences of the dedicated PP room are presented in more detail.

4.1. Amount of pair programming

In order to get a rough estimate of the amount of PP, we asked the respondents to estimate how many hours they had used PP since the previous survey. Accurate data could not be collected because the department's time reporting system did not support reporting PP effort separately. The amount of PP remained quite low even though it doubled between surveys 3 and 4. During 2006 the use was on the order of 10% of all development work, and individual developers used PP from zero to a few hours per week on the average.

In survey 3, 60% of the respondents wanted more PP, and 40% were pleased with the amount. Promoting PP as a voluntary practice probably explains the lack of respondents wanting less PP. The questions about problems revealed issues related to organizing and infrastructure for PP. This explained why many developers had not used PP as much as they desired. Based on the findings, a new tactic, i.e., a PP room was taken into use, and the PP guidelines were slightly changed. Despite of the doubled use of PP, 57% of the respondents in survey 4 still wanted more PP, and 43% were pleased with the amount. The broad-based desire of increasing use of PP indicates that some problems were remaining with respect to the possibilities to use PP.

4.2. Attitudes to pair programming

Negative attitudes to PP can complicate its adoption, because PP requires quite a radical change in working methods. In order to assure the viability of the adoption, we inquired the developers' attitudes to PP.

Starting with survey 1, the feelings of PP were mainly positive (Figure 2). Initially three respondents had negative feelings, but even their feelings improved later (2→6, 2→6, and 3→5). One of them worked mostly remotely at the time of survey 1, and therefore did not feel positive about PP. The median of the feelings improved from 5.0 to 6.0 over time but the difference between the first and last survey was not statistically significant (Mann-Whitney p=0.13).

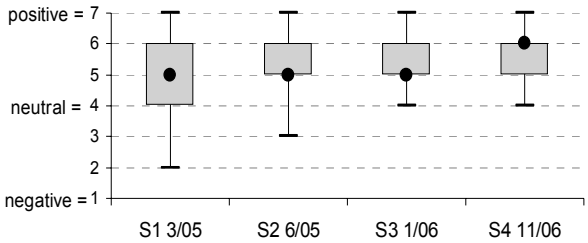

Figure 2. Feelings of pair programming (N_{S1}=19; N_{S2}=14; N_{S3}=21; N_{S4}=21)

The developers' feelings of PP before they had tried it were quite neutral, the median being 4.0. The difference between the preconceptions of PP and the feelings of PP in the last survey is statistically significant (Mann-Whitney p=0.001).

The feelings of solo programming (SP) were slightly above those of PP in surveys 1-3, but fell slightly below SP in survey 4 (Figure 3). The decrease was mainly due to the somewhat low feelings of SP by a few new employees answering the survey for the first time, but there was also some decrease among the other respondents. The differences between the feelings of SP and PP were not statistically significant in any of the surveys (Mann-Whitney p>0.19).

According to the latest responses from each respondent (N=30), 50% of the developers felt the same about both PP and SP, 20% preferred SP, and 30% preferred PP. We assumed that the preference for PP might correlate negatively with the number of work years in the company, because juniors may benefit more from PP, and on the other hand seniors may have higher resistance to changes. Contrary to our assumptions, there was no correlation (Spearman rho=0.1).

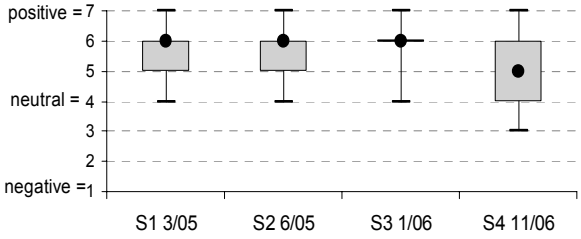

Figure 3. Feelings of solo programming (N_{S1}=19; N_{S2}=14; N_{S3}=21; N_{S4}=21)

We assumed that some developers might find it boring to act as the more skillful partner especially if they were not driving and were not eager teachers. On the other hand, the less skillful partner might be afraid of showing his/her weaknesses to the more skillful partner. Therefore we studied if there was resistance to either of the situations. In general the feelings of both situations were positive and improved over time as did the general feelings of PP. The latest responses (Figure 4) indicated slightly higher feelings for acting as the less skillful person in a pair, but the difference was not statistically significant (Mann-Whitney p=0.46).

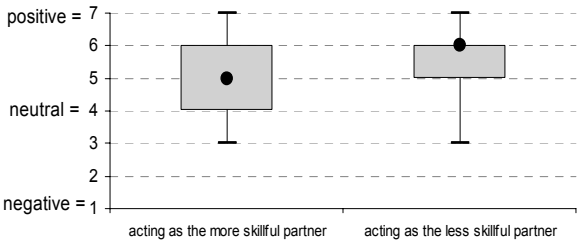

Figure 4. Feelings of acting as the more vs. less skillful partner (N=30)

The analysis of the individual answers revealed that 53% of the developers felt the same about both situations, 30% preferred being the less skillful partner, and 17% the more skillful partner. Three developers felt negatively about either of the situations, but all of them felt positively about the opposite situation.

4.3. Organizing for pair programming

Several developers reported problems in resourcing PP, especially in survey 3, but also in survey 4. The problems included being too busy to use PP related to the other developers' tasks, difficulties in finding common time, lack of encouragement from team leaders, and not considering PP in project planning. Some developers reported that they have not used PP because their tasks have not been suitable for PP, i.e. their tasks have been simple and/or independent.

"*The problem is getting PP organized. Everyone on the team agrees it would be beneficial, yet it is not done.*"

"*Finding common time is not easy: both have also own tasks that must be done. If common time is not found, nothing is done and then late in the development pair task is done mostly alone (in a hurry) because someone must implement the task anyway. The result is no PP at all.*"

"*I have had no opportunities to do PP. It doesn't seem to be a factor in project management and resource planning.*"

"*Team leaders are not encouraging team members enough to organize PP, and therefore PP sessions are not organized often enough. Personally I feel that other team members are so busy doing their own tasks that they don't wish to waste their time on anyone else's tasks.*"

It seems that the developers had found suitable ways for pair formation, because no problems were reported related to it. However, there were some contradictory opinions on the best ways to do it. Several developers considered the developers should have the final decision on the use of PP and on the partner.

"*Pair forming should be done as needed by the participants, but one should avoid the same pair all the time.*"

"*PP can be encouraged by management but it will have positive results only if both persons feel positive about it.*"

However, some other developers recommended planning the use of PP in weekly team meetings, and team leaders should be more active in ensuring that PP gets done. We think this may be due to problems of getting more senior developers to do PP with the juniors even if the juniors would like to use PP.

"*Every week, decisions should be made on the task, number of hours, and paired people in the team meetings. And in the team meeting next week, it should be checked that the PP tasks were accomplished.*"

"*PP should be encouraged but not enforced. The team leader can suggest pairs.*"

Based on the results from survey 3, the PP guidelines were updated in order to create a more encouraging and more positive atmosphere for PP. The updated guidelines explicitly mentioned that both the team leader and any developer may propose the use of PP. The proposals about the features for which PP would be used, should preferably be made when a four-month long iteration's work is planned and assigned to developers. The proposal should identify the partners, and roughly how much they should use PP for the different activities related to the development of the feature, e.g. specification and coding. In addition, it was emphasized that team leaders should encourage using the planned amount of PP for the selected features.

The feelings of the organizing for PP improved slightly after survey 3 (Figure 5). All four negative answers disappeared, but the median remained at 5.0. The difference was not statistically significant (Mann-Whitney p=0.101). Some comments in survey 4 indicated that the same problems still remained.

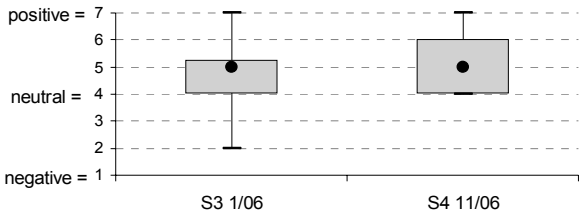

Figure 5. Feelings of the organizing for pair programming (N_{S3}=20; N_{S4}=21)

4.4. Infrastructure for pair programming

The first author observed three PP sessions and noticed problems in the infrastructure for PP. Two of the PP sessions took place in a meeting room using a laptop with a small display. The third PP session was arranged in an ordinary work space, which was quite cramped, and the navigator had to sit behind the shoulder of the driver. Due to these observations, the recommendations and problems related to the infrastructure for PP were inquired in surveys 3 and 4.

A frequently mentioned problem was that the noise from PP disturbs others. Therefore, PP was sometimes done in a meeting room using a developer's laptop. Several respondents suggested having a separate well-prepared room for PP, which was also considered to allow the pair to work without disturbance from others.

"*Coding could be done in a normal work space, but specification and other tasks requiring more talking should be done in meeting rooms.*"

"*A separate place where you could do PP would be good, as then nobody would come and disturb the session, and talking much would also not disturb anyone else.*"

The developers proposed several improvements: big screens, wireless mice and keyboards, more suitable desks for PP, and whiteboards on the walls.

"*A big screen would also be good so it's easy for both programmers to see what's being produced.*"

"*If the computer is in a corner of a table, it is impossible for both to be equally involved in the session. The best place for the computer is in the middle of a long enough straight table located in a separate room. The lack of such PP infrastructure is a current problem.*"

"*There should be more space, because the personal work space is currently too small for two people.*"

There were also some software related comments.

"*Problems might arise when different tools are used. Then the desktop owner has to be on the keyboard all the time and no role change can be done.*"

"*The driver's desktop environment should be available.*"

The feelings of the infrastructure for PP improved a lot between surveys 3 and 4 (Figure 6). The median rose from 4.0 to 6.0, and the difference is statistically significant (Mann-Whitney p=0.004).

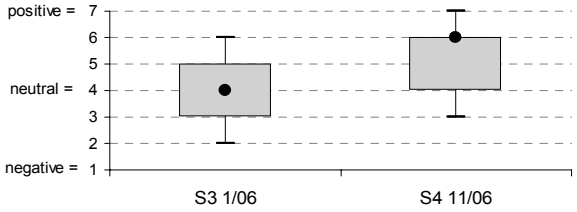

Figure 6. Feelings of the infrastructure for pair programming (N_{S3}=21; N_{S4}=20)

4.5. Pair programming room

A PP room was adopted in the spring of 2006. It contained a desk with *two* computers with large displays, a long, straight table, rolling chairs, and a whiteboard. Developers worked over a remote connection to their own desktop computer in order to have a familiar development environment. The room could be reserved using the company's meeting room reservation system.

In survey 4, we asked about the preferred place for PP: 1) own work space, 2) PP room, or 3) ordinary meeting room. The PP room was preferred by 89% of the respondents. The simplicity of doing ad-hoc PP sessions, and being closer to the others if help was needed, were the only benefits of the own workspace.

The answers to our question of the benefits of the PP room compared to the own work space focused on the avoidance of noise and better infrastructure for PP.

"*In the PP room you can be by yourselves and don't make it hard for other people to concentrate.*"

"*Time is not wasted on looking for e.g. an extra chair.*"

"*Two machines and enough space for both.*"

We asked about possible problems with the PP room. There were only a few small complaints about

poor air conditioning, chairs, mouse, and keyboard. The most common "complaint" was the need for more PP rooms:

"*It would be nice to get the room ad hoc — this is why a backup in for example meeting room could be good.*"

"*There should be another such room. The only one is quite often occupied.*"

Usually pair programmers work at the same computer, but in the PP room there were two computers side-by-side. We asked how the developers divided work activities between the two computers. The work was typically done on one computer and the other was used only sometimes for browsing specifications or code, finding information, testing and debugging.

"*One computer is for development and the other one is for checking things and finding information.*"

"*Coding with one and maybe reading specs on the other or it's just idling.*"

We asked the developers to estimate how much they worked together with one computer, and separately with two computers during their latest PP session. The answers differed among the developers but the averages were 85% together and 15% separately. The situation can thus well be called PP, but the additional computer may allow more efficient work when some things need to be clarified e.g. by finding information, or when trivial code is written and the partner can do something else during that.

4.6. Perceived effects of pair programming

A detailed analysis of the perceived effects of PP is reported in another paper [18]. Briefly summarized, the perceived effects of PP were positive in all of the surveys. The effects were most positive for learning, schedule compliance of tasks, making the acquaintance of other developers, and team spirit. A small but clearly positive effect was perceived for various quality aspects, discipline in following work practices, and enjoyment of work. On the negative side, the development effort of individual features was higher. First, the exhaustiveness of work was higher, but over time it decreased to the level of solo programming.

5. Discussion

Below we discuss our results and compare them to previous studies. Then we summarize the lessons learned, and discuss the limitations of this work.

5.1. Summary and comparison to other studies

In our case the developers' attitudes to PP should not have hindered the use of PP. The preconceptions of PP were neutral and the feelings of PP continuously improved even slightly surpassing the feelings of SP. This is in accordance with Williams and Kessler [1], who propose that most people resist PP, but after trying it decide it is better than working alone. Everyone's last response regarding their feelings of PP and SP were positive, but half of the developers preferred either PP or SP. The last responses regarding acting as the more or less skillful partner were also mostly positive for both situations, but again about half of the developers preferred either situation. A few developers had negative feelings of either situation, but even they had positive feelings of the opposite situation. The positive perceptions of the effects of PP probably had a positive effect on the attitudes.

There were clear problems related to the resourcing of PP, as also reported by others [12,13]. It seems that the identification of the problems and the changes to the guidelines after survey had a small positive effect on the feelings of organizing for PP in survey 4, but the problems were not completely solved.

First there were problems in the infrastructure for PP, but by the end of the study the feelings of it were so positive that the problems should no more hinder the use of PP. The only major change in the infrastructure was the adoption of the PP room, and it seems to have solved the problems. It may be that the commonly proposed agile practice of a co-located team did not work well in an environment where there were several teams in the same office. In this case, the background noise is often not related to one's own work and can disturb concentration without providing valuable knowledge transfer. In addition, due to the low proportion of PP of all work, most of the developers in the office were typically working alone, which makes them more prone to be disturbed by the noise as proposed in [1].

The use of PP increased slowly, first due to at least issues related to resource allocation and infrastructure for PP. Improvements were made after survey 3, and the use of PP doubled, but was still only ca. 10% of development work. In survey 4, 57% of the developers still desired more PP, which indicates there were problems remaining. We believe they were still related to resourcing PP, because attitudes to PP and infrastructure for PP should not hinder the use of PP anymore.

5.2. Lessons learned

The company employed five tactics that we think helped the adoption of PP: the use of guidelines for PP, having a champion for PP, making the use of PP voluntary, creating a positive atmosphere for PP, and instituting a dedicated PP room.

When starting the adoption of PP, the company created a set of simple guidelines for its use. The guidelines were later revised to tackle, e.g. problems with lacking resource allocation for PP.

Since the use of PP was voluntary, its use varied among developers. However, after a slow start the use of PP doubled. At the end of the study, the majority of the developers reported wanting to use it even more. Encouraging the use of PP instead of enforcing it

probably helped lower the barrier to trying PP and created a positive atmosphere for PP.

The positive atmosphere was seen in the developers' positive attitudes to PP at all points of the study, and regardless of whether they were the more or less skillful partner in a pair. However, there were individual preferences of SP vs. PP and acting as the less vs. more skillful partner, which should be taken into account when forming pairs. We did not find any significant difference in preferences for PP vs. SP between senior and junior employees.

Initially, developers complained about inappropriate infrastructure for PP. This problem was alleviated by instituting a dedicated PP room that provided both the needed space and technical infrastructure to support PP, as well as helped shield other developers from the noise created by the pair programmers.

5.3. Limitations and further work

Using surveys as the main data collection method has the strength of getting lots of data from a large group of respondents, but may restrict the findings to what is expected. However, in addition to the surveys, we discussed several times with the PP champion and with the developers when presenting intermediate study results, and observed three PP sessions. Thereby, we got a reasonable understanding of the overall PP situation and were able to improve the questionnaires.

The differences between the sets of respondents in the surveys, e.g. the proportion of new employees, could have had an effect on the overall results of a survey. Therefore we analyzed, in addition to all respondents, the subset of respondents who answered all surveys. The changes in the answers between the surveys were similar between this subset and all respondents of each survey, which indicates that the different sets of respondents did not affect the results.

While we think that our study can provide useful lessons regarding the adoption of PP, it is inherently limited by the fact that it is a single case study, which must be taken into account when generalizing the results. Several tactics used by the company are likely to be useful for others, such as the use of PP guidelines and having a PP champion, but other practices might not be that widely generalizable. For example, the need for a dedicated PP room can be explained by the use of an open office space for the whole development organization, and the lack of team collocation.

We think that the study makes a worthwhile contribution despite its limitations, as current literature contains little empirical information on the adoption and practicalities of PP in industry. We hope that our study helps encourage more empirical research into PP adoption. In the future, we hope to be able to extend the study and make the results more generalizable by studying other organizations adopting PP.

References

[1] L. Williams and R. Kessler, *Pair Programming Illuminated*, Addison-Wesley, 2002.

[2] L. Williams, *The Collaborative Software Process*, Ph.D. dissertation, University of Utah, 2000.

[3] J. Nosek, "The Case for Collaborative Programming", *Communications of the ACM*, 41(3), 1998, pp. 105–108.

[4] J. Vanhanen and C. Lassenius, "Effects of Pair Programming at the Development Team Level: An Experiment", *In Proceedings of International Symposium of Empirical Software Engineering (ISESE2005)*, 2005.

[5] K. Lui and K. Chan, "Pair programming productivity: Novice–novice vs. expert–expert", *International Journal of Human-Computer Studies*, 64(9), 2006, pp. 915–925.

[6] J. Nawrocki and A. Wojciechowski, "Experimental Evaluation of Pair Programming", *In Proceedings of the 12th European Software Control and Metrics Conference*, 2001.

[7] M. Rostaher and M. Hericko, "Tracking Test First Pair Programming – An Experiment", *In Extreme Programming and Agile Methods - XP/Agile Universe*, 2002.

[8] R. Gittins, S. Hope, and I Williams, "Qualitative Studies of XP in a Medium Sized Business", *In Proceedings of the XP 2001 Conference*, 2001.

[9] K. Beck, *Extreme Programming Explained*, Addison-Wesley, 2000.

[10] L. Williams, "Extreme Programming Practices: What's on Top?", *Agile Project Management, Executive Report*, 12(5), Cutter Consortium, 2004.

[11] A. Pandey, N. Kameli and A. Eapen, "Application of Tightly Coupled Engineering Team for Development of Test Automation Software – A Real World Experience", *In Proceedings of the 27th Annual International Computer Software and Applications Conference (COMPSAC'03)*, 2003.

[12] G. Luck, "Subclassing XP: Breaking its rules the right way", *In Proceedings of the Agile Development Conference (ADC'04)*, 2004.

[13] B. Greene, "Agile Methods Applied to Embedded Firmware Development", *In Proceedings of the Agile Development Conference (ADC'04)*, 2004.

[14] H. Gallis, E. Arisholm, and T. Dybå, "An Initial Framework for Research on Pair Programming", *In Proceedings of International Symposium of Empirical Software Engineering (ISESE2003)*, 2003.

[15] M. Ally, F. Darroch, and M. Toleman, "A Framework for Understanding the Factors Influencing Pair Programming success", *In Proceedings of the XP 2005 Conference*, 2005.

[16] H. Sharp and H. Robinson, "An Ethnographic Study of XP Practice", *Empirical Software Engineering*, 9(1–2), 2004, pp. 353–375.

[17] R. K. Yin, *Case Study Research: Design and Methods, 2nd ed.*, Sage Publications, 1994.

[18] J. Vanhanen and C. Lassenius, "Perceived Effects of Pair Programming in an Industrial Context", *In Proceedings of the 33rd EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO-SEAA'07)*, forthcoming.