Original source of publication: An Empirical Analysis on the Discontinuous Use of Pair Programming. XP 2003, p. 205-214, 2003
DOI: http://dx.doi.org/10.1007/3-540-44870-5_26
The original publication is available at http://link.springer.com/chapter/10.1007/3-540-44870-5_26

An Empirical Analysis on the Discontinuous Use of Pair Programming

Andrea Janes, Barbara Russo, Paolo Zuliani, Giancarlo Succi

Center for Applied Software Engineering, Faculty of Computer Science
Free University of Bolzano-Bozen
Dominikanerplatz 3 Piazza Domenicani
I-39100 Bozen, Italy
e-mail: Giancarlo.Succi@unibz.it

Abstract. Pair Programming has been shown to increase communication and teamwork skills and to provide better code. The aim of this work is to show the efficacy of Pair Programming in transferring knowledge and skills over an environment where people met only occasionally. In a quasi experiment, we find that Pair Programming is effective in sharing knowledge among 15 students who met once a week for a half day, and did their internship individually or in couple for the remaining 4 half days.

1 Introduction

Pair programming has usually considered non effective for distributed teams, not working most of the time together ([3], [4], and [8]). In this paper we discuss the effectiveness of Pair Programming at transferring knowledge and skills among students that met only occasionally and worked mostly independently.

The effect of geographical distance between pair programmers has been already addressed by Baheti *et al.* ([5]). They performed an experiment on a graduate class to assess whether it is feasible to use distributed PP to develop software. It turned out that distributed (i.e., geographically distant) pair programming teams can effectively develop software, that is, with productivity (in terms of LOC/hr) and code quality (in terms of grade awarded to the project developed) comparable to those of close-knit teams.

Kircher *et al.* ([2]) identify the aspects of XP which require co-located programming teams. The authors analyze these aspects in the distributed development of software for collaborative productivity. They found that the effectiveness warranted by physical proximity could not be completely substituted by any communication tool, though various combinations turned out to be quite effective. However, their findings are based only on the personal opinions of the participants, the authors themselves, and no empirical evidence is provided.

We report on the experience of a group of fifteen students doing a summer internship experience at the end of their first year of a first degree in Applied Computer Science at the Free University of Bolzano-Bozen (Italy). For three months students worked either in companies or research centers the whole week but Friday afternoons, when

they met altogether in a university laboratory. Here, they worked on a different project using PP. Our aim was to monitor the knowledge they acquired from such a structured context. Even if such an environment is not distributed in the genuine sense of the term, similar factors may affect the success of the project. Indeed problems with non-continuous use of the same software practices, difference of environments and requests and geographic distance can be equally experienced.
This paper is organized as follows. In Section 2 there is a discussion of the structure of the experiment, including a GQM schema, the questionnaire used to gather the data, the description of the sample and of the environment. In Section 3 there is a presentation of the results, with specific analysis of the effectiveness of communication tools, of pair programming as a vehicle for knowledge transfer. In Section 4 we draw some conclusions.

2 Structure of the experiment

As mentioned, this research deals with a group of fifteen (volunteer) students doing a three-month summer internship.

Eleven students worked in local companies for all the working days but Friday afternoons, when all of them met in a university laboratory for four hours to share their experience. A group of four students worked for a research center of the Faculty - joining the others on Friday afternoons.

The environment was distributed in the sense that the students had the chance to work together only one afternoon per week, spending the rest of the week working in geographically distant places.

In the Friday afternoon meetings all the students had the possibility to share their knowledge and skills by developing software using Pair Programming. This work was completely independent from what they were doing over the rest of the week. In all the companies there were no special indications to use Extreme Programming practices except for students working for the university lab, where XP was continuously adopted.

Altogether, the use of Pair Programming was non-continuous – only on Friday afternoons – and alternated with other coding styles.

At the end of their experience students answered to a questionnaire.

2.1 GQM of the experiment

To properly structure the experiment, we use the well known Goal-Questions-Metrics (GQM) paradigm ([6]) according to the guidelines of [7]:

Goal:

- Monitoring skills acquired in using PP in order to investigate:
- Knowledge transfer
- Effectiveness of a non continuous PP practice - alternated with a different programming methodology
- Integration of XP skills learned at University and practices acquired during an industrial internship

Questions:

- How much effective is the use of PP in transferring knowledge in a distributed environment?
- How much effective is PP in a non temporary continuous work alternated with other practices?
- How much effective is the use of PP in integrating University studies and applicative practices of a company of an industrial environment?

Metrics:

- Final questionnaire

2.2 Structure of the questionnaire

The final questionnaire was developed according to standard questionnaire-writing styles. It consisted of three main parts: the first described the student's status – work experience and skills, the second dealt with the Internship experience and the third reported the students' opinion on the PP style. The questionnaire was structured by several multi-choice questions alternated with some rating and free-style questions. It covered topics listed in Table 1.

Table 1: Main subjects of the questionnaire

Topics	
1. General work experience	7. Internship: Communication tools
2. Skills in Computer Science	8. Internship: PP Best Aspect
3. Skills in some PP features	9. Internship: Benefits
4. Internship: Project knowledge	10. Evaluation PP: Hardest Thing
5. Internship: Project structure	11. Evaluation PP: Non Effectiveness
6. Internship: Project support	12. Evaluation PP: Most Important Aspect

In the first three points of Table 1, the student's work experience in computer science is evaluated. It was measured by common questions on work experience and on some aspects of team working.

Points 4 to 9 of Table 1 describe the environment of the internship. Point 4 focuses on what of the project was known before the Internship experience, such as tools (NetBeans, …), languages (Java, PHP, …) and approach to the problem - how to translate requirements into code.

To evaluate the students' degree of comprehension of the project, point 5, 6 and 7 asked students to describe the project – structure, and technical and human support – and the communication tools they used during the experience.

Points 8 and 9 measured the PP practice rooted in the students' experience, while points 10 to 12 asked students to give an opinion on the PP style independently from the project.

In Table 2 we reproduce the acronyms of the measures. Besides each acronym we put the reference number of Table 1. Points 8, 10 and 12 were in the form of free-style questions, so they are not included in Table 2.

Table 2: Acronyms of measures

1-2	**WS**	Working student
	WE	Work experience
	WECS	Work Experience in Computer Science
	WTE	Experience in Working in Team
	WPE	Experience in Working in Pair
	PPW	If WPE: Is Pair Programming worth?
3	**PSC**	Experience in working in pair sharing the same computer
	SC	Experience in working on the same code
	WD	Experience in work division
	SE	Experience in sharing experience
4	**TL**	Project Tools knowledge
	PA	Knowledge on how to translate requirements in code – Problem Approach
5	**SP**	Switched partner more than two times
6	**CP**	Customer's physical presence
	PS	Reference Instructor's technical support
7	**T**	Use of Telephone
	NM	Use of NetMeeting
	IM	Use of Instant Messenger
	EM	Use of e-mail
9	**LC**	Increasing Learning and Comprehension
	CT	Increasing Communication and Team working
	TM	Increasing Time Management
	OE	Increasing Opportunity of experimentations
	SR	Increasing Self-Reliance
	PSST	Increasing Problem Solving and Strategy Thinking
11	**SAFYC**	The use of PP is not effective Soon After a First Year Course
	STE	The use of PP is not effective for a Short Experience
	BPEC	The use of PP is not effective if Both Partners are not Equally Competent
	PU	The use of PP is not effective if the Project is Unknown

2.3 Details on the sample

In this section we characterize our sample by studying the answers to the first part of the questionnaire and the cross-correlations among them.

Fifteen students volunteered for this project. Eleven were full-time students with some previous work experience, while four were part-time students (with part-time jobs). In Table 3 we report the frequencies for the answers of the questionnaire regarding the students' previous skills and knowledge. The frequencies are based on a sample of size fourteen, as one questionnaire was not returned. Frequencies, Pearson's cross-correlation coefficients and p-significance (as usual, we consider $\alpha<0.05$) are calculated using SPSS, a well-known statistical tool.

We also note that the sample of the PPW variable has size five, that is, the number of students who answered "yes" to the "Experience in Working in Pair" (WPE) question.

Table 3: Frequencies of the previous skills and knowledge of the sample

(%)	General working experience (1-2)					
	WS	**WE**	**WECS**	**WTE**	**WPE**	**PPW**
no	71.4%	14.3%	64.3%	42.9%	64.3%	20%
yes	28.6%	85.7%	35.7%	57.1%	35.7%	80%
n/a	0%	0%	0%	0%	0%	0%
(%)	**PP aspects experience (3)**				**Project Knowledge (4)**	
	PSC	**SC**	**WD**	**SE**	**TL**	**PA**
no	50%	57.1%	14.3%	28.6%	28.6%	21.4%
yes	35.7%	28.6%	71.4%	57.1%	71.4%	78.6%
n/a	14.3%	14.3%	14.3%	14.3%	0%	0%

From Table 3, we can infer that the majority of the students had a previous work experience (WE), a few of them in Computer Science (WECS). More than 70% had a good knowledge of the project they were going to start (see TL, PA in Table 3). Some students (WPE 35.7%) had already practiced PP in the past, and most of them found it worth (PPW 80%). Students with work experience (WE) have more experience in teamwork (WTE) than working in pair (WPE).

The cross-correlations resulting from the first part of the questionnaire (Table 4) confirm the students' curricula. Again, we see that students' work experience is mainly in computer science. General team working has a good correlation with work experience.

Table 4: Correlations between different aspects of students' know how

	WTE	**WECS**	**CP**	**PSC**	**SC**	**WD**
WS		0.85 p=0.000				
WE	0.57 p=0.032					
PSC			0.60 p=0.023		0.92 p=0.000	
WD				0.77 p=0.001	0.74 p=0.002	
SE				0.69 p=0.007	0.64 p=0.015	0.88 p=0.000
WPE				0.65 p=0.012	0.53 p=0.050	

From Table 4 we may infer that students who experienced a general work in pair, know and appreciate the PP practice in some of its aspects – "Experience in working in pair sharing the same computer" (PSC) and "Experience in working on the same code" (SC).

Table 4 also shows that the four different aspects of PP are each other correlated. This might mean that students had a somehow "homogeneous" experience of PP (i.e. they did not practice just one aspect).

2.4 Details on the environment

The companies selected for the internship were mainly local businesses. Some were software houses, others non-IT organizations with an EDP department. Students selected the companies on a First-In-First-Out basis.

To take full advantage of the internship, students were introduced to the project with several seminars related to the experience they were about to begin. Different subjects were presented: legal rights and duties, role of the unions, importance of, and techniques to communicate within corporate organizations, how to secure funds to create a start-up and so on. They were also introduced to team working by role play. They were taught time and stress management, how to support a talk and how to give priorities.

At the beginning of the internship, each company assigned a task to the student. Most of the time company assignments were part of a big project already started. Since the students had attended a course on Java during the previous semesters, all of them were not only able to use Java, but also to learn new languages and tools.

A company-internal reference person was selected to act as internal "tutor" of the student. Additionally, some selected members of the Computer Science Faculty provided technical and social support to students and monitored the overall experience.

So, almost every week a member of the University staff visited the student in the company and reported on the student's situation. Reports were published on an internal web site, so each instructor could access them. Students and companies were aware of the dates of the visits in advance, so that the internal tutor could be present to the visit.

In the Friday afternoon meetings all the students gathered in a university laboratory and worked, using Pair Programming, on a project different from what they were working on in the rest of the week. Therefore, in such meetings all the students had the chance to communicate, to compare and to analyze their weekly experience, evidencing similarities and differences. In this way they had the possibility of increasing their skills by knowledge transfer.

An instructor and a "virtual customer," i.e., a faculty member acting as the customer, were always present in the room.

The Friday afternoon project was divided into independent subprojects, each assigned to a group of four students experiencing PP. They periodically switched partners in the team.

In each of the four teams there was a member of the group of students working for the CASE, i.e., a student who was experiencing PP the whole week.

2 Results

In this section we present a summary of the results of the questionnaire. We analyze the results in two parts. First, we study how communication tools were used. Second, we report on how PP was effective in transferring knowledge and skills among participants.

As usual, we only consider Pearson's correlation coefficients whose p-significance is less than 0.05.

3.1 Use of communication tools

The two tables below provide some understanding on the use of communication tools. The most used communication tool has been e-mail, but telephone and instant messenger were also adopted - Table 5.

The use of telephone is negatively correlated with the tools used for coding during the Internship - Table 5. This means that students used telephone when they had troubles with the software tools. In the same way we may say that NetMeeting was used by students who initially knew little on how to approach the project. The use of Instant Messenger is negatively correlated with "Self-Reliance" (SR) - Table 6. These three facts might indicate that the more students think they have increased skills, the less they use "synchronous" communication tools.

On the other hand, "Use of E-Mail" (EM) has a good correlation with the initial ability in approaching the project (PA). From this we may instead infer that students with increased skills preferred to use "asynchronous" communication tools.

Table 5: Use of Communication Tools

(%)	Communication Tools (7)			
	T	NM	EM	IM
no	57.1%	85.7%	21.4%	57.1%
yes	42.9%	14.3%	78.6%	42.9%
n/a	0%	0%	0%	0%

Table 6: Cross correlations between Use of Communication Tools and Knowledge of the Project

	Cross Correlation		
	TL	PA	SR
T	-0.73 p=0.003		
NM		-0.78 p=0.001	
EM		0.58 p=0.031	
IM			-0.58 p=0.031

To summarize, the results of this part of the questionnaire indicate that students preferred synchronous, real-time communication tools when they knew little about coding tools or problem approach, otherwise, e-mail was the most used communication tool.

3.2 Knowledge transfer and effectiveness of PP

In this section we assess the effectiveness of PP in transferring knowledge and skills.

We omit frequencies On Internship benefits and non effectiveness of PP, 80% of the students answered positively on each benefit listed in Table 7.a (except for the last two items, for which slightly less than 50% gave a positive answer). This entails that students actually experienced a transfer of knowledge and skills. In Table 7.b less then 50% of the students considered Pair Programming non effective whether the kind of project is unknown and even less considered Pair Programming unfeasible for a short time experience.

Table 7: Ranking internship benefits (a) and conditions for non-effectiveness of PP (b)

	Benefits of Internship
Low------->High	Communication (CT)
	Problem solving (PSST)
	Learning and comprehension (LC)
	Time management (TM)
	Self-reliance (SR)
	Opportunity to experiment (OE)

a

	Conditions for Non-Effectiveness of PP
Low------->High	Unknown project (PU)
	Experience soon after a first year course (SAFYC)
	Member of pair not equally competent (BPEC)
	Short experience (STE)

b

Table 8: Cross-correlations with Internship Benefits

	PSST	**OE**	**SP**	**CP**	**EM**	**IM**	**STE**
LC	0.78 p=0.001		0.59 P=0.026				
CT				0.68 p=0.008			
TM							-0.74 p=0.002
SR						-0.58 p=0.031	
PSST		0.57 p=0.032	0.57 p=0.032				
OE					-0.55 p=0.042		

From Table 8, the two abilities “Increasing Learning and Comprehension” (LC) and “Increasing Problem Solving and Strategy Thinking” (PSST) are both correlated to each other and with “Switch partner more then two times” (SP). This might mean that switching partner more than two times during the Friday afternoon PP sessions had a good influence in increasing global comprehension of the project and maturity of the students.

“Communication and Team working” (CT) is positively related with the “Virtual Customer’s physical Presence” (CP). From this we might infer that the on-site presence of the customer (one of the XP practices) influenced favorably the communication and teamwork skills of the students. This also suggests that PP should always be practiced with a strong presence of the customer.

The ability to manage time is highly and significantly negatively correlated with the non-effectiveness of a brief PP experience. By the frequency of the positive answers (72%) to "Increasing Time Management" (TM) we may infer that students think that PP helps to manage time better.

The last part of the questionnaire provided the students with the possibility of giving a personal opinion about PP independently from the project. In Table 9 we report the most significant cross-correlations between variables that we have extracted.

Table 9: Cross-correlation - Condition for non-effectiveness of Pair Programming

	SAFYC	STE	TM
SAFYC		0.64 p=0.014	
STE			-0.745 p=0.002
BPEC	0.54 p=0.046		

The correlation of "Considering the use of PP not effective soon after a first year course" (SAFYC) with "Considering the use of PP not effective for a short experience" STE and "Considering the use of PP not effective if both partners are not equally competent" (BPEC) confirm well known results on the XP practices.

By the students' answers to the free-style questions and by the individual meeting with a faculty member we inferred that at the end of the experience the students were conscious of the limitations and benefits of PP. In particular, conflict of personalities and difference in skills caused most of the problems in Pair Programming. The most common answer to the best aspect of PP has been – as students said – "two minds working on the same code". This might mean that although PP attracts students, they realize that this coding style is really involving.

To summarize, we saw that the vast majority (80%) of the students benefited from the experience in four ways: "Learning and Comprehension" (LC), "Communication and Teamwork" (CT), "Time Management" (TM) and "Problem Solving and Strategy Thinking" (PSST). We saw from Table 9 that these four benefits are correlated with PP aspects, namely "Switched partner more than twice" (SP), "Virtual Customer Presence" (CP) and "Use of PP is not effective for a short experience" (STE). From this we may infer that the benefits which participants received came from experiencing PP.

3 Conclusions

This paper aimed at investigating the effectiveness of Pair Programming as a tool for experience exchange. We performed a first analysis of the experience of a summer internship program run on a group of fifteen students. The goal was to assess the transfer of knowledge and skills when using Pair Programming (PP).

The peculiarity of this case study consisted in the kind of distributed environment and in a methodology approach in which PP was alternated with other programming styles. Most of the students worked in separate companies the whole week but Friday afternoons, when they met in a university laboratory to work on a different project using Pair Programming (there were no special indications to use PP when working for the companies).

Increased communication ability was the benefit that 92% of all the students felt to have gained. Also, the vast majority of students found their problem-solving, time management and learning abilities improved. These benefits are correlated with the practice of PP. Therefore, Pair Programming was effective at transferring knowledge and skills. We also found that the students' levels of self-reliance and project knowledge affect the use of communication tools: the more students become conscious of their abilities the less they use communication tools (and the more they think that meeting the partner once a week is enough).

Our results confirm previous empirical evidence about the benefits and the good resistance to distance hampering factors of Pair Programming. We gathered new empirical evidence which shows that PP keeps its effectiveness also when alternated with other coding styles. Our findings might be of help to people involved in the distributed development of software projects (*e.g.*, open source software), as well as to educators for planning and running programming projects with teams composed of distance-learning students.

Acknowledgements

The authors would like to thank the ESF office of the Province of South Tyrol (Italy) and its chairman, Dr. Barbara Repetto, for providing full support to the experience. The authors would also like to thank all the participating students and companies.

References

1. Kircher, M., Jain, P., Corsaro, A., Levine, D.: Distributed eXtreme Programming. Proceedings of XP2001 May (2001). Available at http://www.xp2001.org
2. Cockburn, A., Williams, L.: The Costs and Benefits of Pair Programming. In: Succi G., Marchesi M. (eds.): Extreme Programming Examined. Addison Wesley Professional (2001) 223-248
3. Williams, L., Kessler, R., Cunningham, W., Jeffries, R.: Strengthening the Case for Pair Programming. IEEE Software. 17 (2000) 19-25
4. Baheti, P., Williams, L., Gehringer, E., Stotts, D., Smith J.: Distributed Pair Programming Empirical Studies and Supporting Environments. Technical Report TR02-010 Dep. Computer Science Univ. of North Carolina at Chapel Hill 15 (2002)
5. Basili, V., Rombach, D.: The TAME Project: Towards Improvement-Oriented Software Environments. IEEE Trans. of Software Eng. 14 (6) (1988) 758-773

6. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B, Wesslén A.: Experimentation in Software Engineering: An Introduction. Kluwer Academic Publishers, December (2000)
7. Succi, G., Marchesi, M., Pedrycz, W., Williams, L.: Preliminary Analysis of the Effects of Pair Programming on Job Satisfaction. Proceedings of XP2002 May 2002. Available at http://www.xp2002.org