T. Hirashima et al. (Eds.) (2011). Proceedings of the 19th International Conference on Computers in Education. Chiang Mai, Thailand: Asia-Pacific Society for Computers in Education

The Effects of Pair Programming in an Introductory Programming Course in Thailand

Alisa SONGSRIWITTAYA
Department of Computer and Information Technology,
Faculty of Industrial Education and Technology,
King Mongkut's University of Technology Thonburi (KMUTT),
126 Pracha Uthit Rd., Bang Mod, Thung Khru, Bangkok, Thailand, 10140
Alisa.son@kmutt.ac.th

Abstract: This experimental study aimed to assess the effects of pair programming pedagogy in an introductory programming course at a public university in Thailand. There were total 65 undergraduate students participated in two sections; 31students in solo section and 34 students in paired section. Results indicated that pair programming students performed better on lab assignments and examinations than their solo counterparts. Moreover, pair programming created a laboratory environment conductive to more active learning and enjoyable than traditional labs. Students and lab assistant reported labs to be more productive and less frustrating.

Keywords: Pair programming, Introductory Programming Course, Undergraduates

Introduction

Teaching programming is a major challenge for IT instructors. Much research has reported the success of pair programming as an application of collaborative learning [1, 2, 3]. When used in undergraduate courses, pair programming has been shown to be very effective way for teaching students how to program [4, 5, 6, 7]. Pair programming involves two students working together at a computer to code one program. One member of the pair called the 'driver' takes control of the mouse and keyboard to develop the program. The other member called the 'navigator' watches for defects and thinks of alternatives [1, 2]. Pair programming is very different from a two-person team project that would be use divide and conquer process. With pair programming all code is developed at a single computer with both students working together [1, 2]. The published studies on pair programming in the classroom have identified a number of reasons why instructors should allow their students to use pair programming. These benefits include more students passing the course, higher quality programs, less time to complete programming projects, increased student satisfaction, increased numbers of students continuing with a computer related major, and possibly better exam scores [4, 5, 6, 7, 8, 9]. For example, the research held at the University of Utah in a senior-level Software Engineering course observed that the benefits for using pair programming included superior results on graded assignments, increased satisfaction/reduced frustration from the students, increased confidence from the students on their project results, and reduced workload of the teaching staff [8, 9]. These observations inspired further research directed by educators at the University of California-Santa Cruz and North Carolina State University to assess the efficacy of pair

programming in an introductory Computer Science classroom. They found that pair programming improved retention rates and performance on programming assignments [4, 5, 6, 7].

In an Introductory Programming Course in Thailand, most programs written by college students are written individually. This practice is based on the belief that the students must write the programs on their own in order to learn how to program. This belief includes the assumption that if allowed to work with a partner one student might do all of the work and the learning while the other student does neither. However, recent studies have begun questioned the practice of necessitating students to complete programming assignments individually [9]. This solitary programming approach has begun to change in recent years with growing numbers of instructors requiring or allowing students to use pair programming. However, little research has been done for assessing the effects of pair programming in an introductory programming course in Thai universities. Therefore, this experimental study is carried out at King Mongkut's University of Technology Thonburi (KMUTT), Bangkok, Thailand. The aim is to assess the effects of using pair programming in an introductory programming course in Thai context.

1. Research Questions

1.1 Are there any differences of the test performances between solo and paired group?
1.2 What are the students' experiences with pair programming?
1.3 How are the laboratory environment based on lab assistant's observations and students focus group of using pair programming?

2. Research Methodology

2.1 Course Setting

An introductory programming course at the Faculty of Industrial Education and Technology, KMUTT is the "Computer Programming Language I" course. This course employs C++ to provide students with a foundation in computer programming. The course is taught with two 50-minute lectures and two 50-minute laboratory sessions each week. Students attend labs in groups of not more than thirty-five with others in their own lecture section. The lab period is run as a closed lab where students are given a weekly assignment to complete during the allotted time. Student grades are based on one midterm exam (30%), one final exam (30%), and eight lab assignments (40%).

2.2 Participants

There were total 65 students enrolled in this course in the second semester of academic year 2010. All of them were the sophomores majoring in Printing Technology who did not have experience about programming. Participants were divided into two sections based on their enrollment, 31 students were in the solo section and 34 students were in the paired section.

2.3 Instruments

There were three instruments used in this study as following:

2.3.1 Midterm and final exams

The midterm and final exams were performed individually on paper-pencil format. Students were asked to complete numbers of small programs. If a student completed all the functions correctly, he/she would be awarded a total score of 30 for each exam. The scores from midterm exam, final exam, and lab assignments would be analyzed using independent samples t-test to compare between solo and paired sections.

2.3.2 A survey questionnaire

The questionnaire used in this study was adapted from [4, 5]. It consisted of two parts. The first part had five demographic information questions such as age, sex, GPA. The second part had four groups of statements described students' experiences with pair programming. For example, the effect of pair on understanding course material, the effect of pair on individual test performance, helpfulness in laboratories, and enjoyment. Respondents rated each statement on three scales: positive effect, neutral effect, and negative effect. The data would be analyzed using percentage.

2.3.3 Focus group report

In this study, two focus groups were held, one with a randomly selected group of five students in paired section and the other one with one lab assistant from both solo and paired section. Analysis of qualitative data from focus groups used to explain laboratory environment in both solo and paired lab.

2.4 Procedure

The experiment was run in two sections of the course; the same instructor taught both sections. Additionally, the midterm exam and the final exam were identical in both sections. One section had traditional, solo programming labs. In the other section, students were required to complete their lab assignments utilizing the pair programming practice. In the pair programming labs, students were assigned partners based on their grade-point-average (GPA) in previous semester. The highest GPA score student would be paired with the lowest GPA score student. Students worked with the same partner throughout the entire semester. We employed closed lab in this experiment because it was easy for controlling the usage of pair programming. The lab assistant could ensure that students were working in pairs at one computer and rotated the roles of driver and navigator periodically. At the end of the semester, data was collected in three ways to assess the effects of pair programming: students' scores from exams and lab assignments; questionnaires of students' experiences with pair programming; and focus groups reports.

3. Results

3.1 Performance on Examinations

Table 1: Midterm Examination Scores

Type	Section	n	Mean	S.D.	t-value	t-prob.
Midterm	Solo	31	22.39	2.591	-2.698	.009*
	Pair	34	24.26	3.018		
Final	Solo	31	22.48	2.014	-2.899	.005*
	Pair	34	24.09	2.442		
LAB assignment	Solo	31	31.81	2.104	-6.024	.000*
	Pair	34	35.09	2.288		

*p-value < .05

From Table 1, the mean scores of midterm exam, final exam, and lab assignments from paired section were higher than solo section. These differences were statistically significant at $p<.05$.

3.2 Experiences with Pair Programming

Table 2: Experiences with Pair Programming

Experience areas	Positive Effect	Neutral Effect	Negative Effect
Understanding of Course	65%	30%	5%
Helpfulness in Laboratories	70%	22%	8%
Enjoyment	80%	13%	7%
Individual Test Performance	30%	60%	10%

From Table 2, the majority of students believed that pair programming had a positive effect on their understanding of the course material (65%), helpful for them to complete tasks (70%), and enjoyable (80%). However, for the effect on individual test performance, the majority of students (60%) presented a neutral view toward to this statement.

3.3 Focus Groups

3.3.1 Students

During the focus group discussion, students stressed the advantages of pairing. Primarily, students brought up the benefits of having their questions answered immediately by their partner rather than having to wait for an instructor or lab assistant. Having someone there while working on problems also seemed to help them pick up on minor errors and to focus on understanding conceptual knowledge. The main concern of pair programming was the imbalance in effort. Many students identified problems that occurred when one of the pair was not as well prepared as the other. Compatibility between members in the pair was also seen as a problem by many students. Some students claimed they could not understand the program written by their partner.

3.3.2 Lab Assistant

In the focus group, lab assistant agreed that implementing the paired programming gave him flexibility and time to give students equal opportunities for questions, discussions, and other

support. As a result of having more time for meaningful exchanges with students, lab assistant found his jobs more satisfying and rewarding when teaching in paired labs. Lab assistant also noted that students in paired labs displayed more active participation in their learning than students in the unpaired labs. Students in paired labs engaged in extensive discussion throughout the entire lab session, and students seemed to help each other resolve questions. Alternatively, solo lab sessions were quiet and appeared to be very frustrating for the students. Frequently, a student needed to wait to ask a simple question. During this waiting period, students were often very unproductive.

4. Discussion

This study provides strong results of the following findings: students in paired labs have a more positive attitude toward working in collaborative environments; this should ultimately help the student in his/her professional life [Table 2]. I believe that previous studies and the above data indicate that students should be allowed to use pair programming. Previous studies have shown that the benefits of using pair programming in an introductory programming course may ensue to faculty and students when pair programming is done in a closed lab and when pairs are assigned [5]. The vast majority of students will learn more when working with a partner to create a working program than they would struggle on their own to create a non-working program. Our data clearly show that the lab assignments produced by students working in pairs are significantly better than the lab assignments produced by individuals [Table 1].

Further study is needed to understand how much additional benefit is accrued from some of the more aspects of pair programming suggested by others, such as effects of gender, partner evaluations, and changing partners versus working with the same partner. In the mean time, we hope more instructors will take the first step and at least let their students voluntarily pair.

References

[1] Preston, D. (2005). Pair Programming as a model of collaborative learning: a review of the research, *The Journal of Computing Sciences in Colleges,* 20(4), 39-45.

[2] Williams, L., & Kessler, R. (2000). All I Ever Needed to Know about Pair Programming I Learned in Kindergarten. In *Communication of the ACM,* 43(5), 108-114.

[3] Williams, L., & Kessler, R. (2000). The Effects of "Pair-Pressure" and "Pair-Learning" on Software Engineering Education. In *13th Conference on Software Engineering Education and Training*, Austin, TX, USA, IEEE Computer Society, 59-65.

[4] McDowell, C., Werner, L., Bullock, H., & Fernald, J. (2002). The Effect of Pair Programming on Performance in an Introductory Programming Course. In *ACM special Interest Group of Computer Science Educators,* Covington, K. Y., 38-42.

[5] McDowell, C., Hanks, B., & Werner, L. (2003). Experimenting with pair programming in the classroom. *Proceeding of the 8th Annual Conference on Innovation and Technology in Computer Science Education,* ACM SIGCSE BULLETIN, 35(3), 60-64.

[6] McDowell, C., Werner, L., Bullock, H., & Fernald, J. (2003). The Impact of Pair Programming on Student Performance, Perception, and Persistence. In *International Conference on Software Engineering 2003,* Portland, Oregon, 602-607.

[7] McDowell, C., Werner, L., Bullock, H., & Fernald, J. (2006). Pair Programming Improves Student Retention, Confidence, and Program Quality. In *Communication of the ACM,* 49(8), 90-95.

[8] Williams, L., Wiebe, E., Yang, K., Ferzli, M., & Miller, C. (2002). In Support of Pair Programming in the Introductory Computer Science Course. *Computer Science Education,* 12(3), 197-212.

[9] Nagappan, N., Williams, L., Ferzli, M., Yang, K., Wiebe, E., Miller, C., & Balik, S. (2003). Improving the CS1 Experience with Pair Programming, in *ACM Special Interest Group Computer Science Education (SIGCSE) 2003,* Reno, 359-362.