

Utilizing Pair Programming Promotion in the Computer Science Classroom

Wanwipa Titthasiri

College of Digital Innovation Technology: Department of Computer Science
Rangsit University, RSU
Pathumthani, Thailand

Abstract— Perception of CS is as a “difficult” subject, especially in programming. Research has suggested that pair programming has many pedagogical benefits. In this study, pair programming and group of pairs programming were utilized through OOP class in practical laboratories and software development project team. Pair programming tests and written midterm-final exams, as well as survey were conducted. Results showed that pair programming creates an environment conducive to success favorable, helps to weaker student frustration free with programming problems, and increases programming self-confidence. There was increase in final exam scores, suggesting pair programming and group of pairs programming have individual programming skill benefits. It is expected that it would influence to help increase retention in CS in the future.

Keywords—Collaborative learning; pair programming; extreme programming; group of pair programming; object-oriented programming; computer science;

I. INTRODUCTION

The shortage of Computer Science (CS) graduates in Thai workforce has been appeared. It was reported that “CS enrollment decline as well as students’ drop outs are problems, facing in every Thai universities [5].” The top reasons for rejecting the CS study, were its’ difficulties in programming and technical work. In addition, programming skill of CS graduates is today poor [4]. Laurie and Robert confirmed that students were very frustrating in an introductory CS courses, in particular programming courses. They were frustrating that “... typically, one-quarter of the students drop out of the classes and many others perform poorly [6].” However, programming skill as the importance of CS basis provides the best possible CS graduates in self-evidence. Thus, there are intense efforts to find the appropriate teaching methods as well as an opportunity for practice in programming classes.

Much of the evidence from several sources indicates that a collaborative approach on the same design, algorithm, code and test yields better than working alone [1][2][3][6][7]. The use of this technique as called pair programming where two programmers work on one program. Pair programming was employed for the in-class activities [7]. “It is a novel collaborative paradigm which is growing in popularity in the CS industry [1].”

In this study, pair programming as a pedagogical technique was focused on the teaching of programming to a second year CS course in Object-Oriented Programming (OOP), although pair programming originated in industry. However, it is focused here on, using in educational settings.

II. OBJECTIVE

While there is a growing body of research in the area of pair programming, few studies (possibly none) have focused on groups of pairs as project software team.

The primary contribution of this study is a direct assessment of the effects of pair programming and groups of pair programming on the development of individual programming skills.

III. LITERATURE REVIEWS

The perception of CS as a “difficult” subject, discourage students from choosing study it in the first place [3][4]. However, it is reported that the benefits of pair programming pedagogies, appear to help increase retention in CS [8]. Thus, pair programming is firstly reviewed.

A. Pair Programming

Pair programming is a collaborative paradigm, originated in industry. Pair programming was employed for the in-class activities where two programmers work on one program. Keith and Michael [7] explained that with pair programming, all code is developed at a single workstation with both students working together. One student assumes the role of the “driver” who writes the code which the other student assumes the role of the “navigator” who makes suggestions, and corrects coding and logic errors as they occur. Laurie and Robert [6] clarified that driver has control of the pencil/mouse/keyboard and is writing the design or code. Navigator as observer continuously and actively examines the work of the driver—watching for defects, thinking of alternatives, looking up resources, and considering strategic implications of the work at hand. The observer identifies error, syntax, misspelling or smaller logic mistakes. Much of researches on pair programming have shown that programmers working in pairs produce shorter programs with better design and fewer bugs than those working alone. Weaker students can learn more from stronger students [3]. The workload of the teaching staff is reduced because the students more often look to each other for technical support and advise [7]. Working in pairs performs a continuous code review because the navigator is constantly reviewing the work of the driver, leading to a higher quality product [6]. Pair programming creates an environment conducive to active learning and collaboration, helps to lower student frustration with challenging problems, and increases programming self-confidence and interest in CS [8]. However, it is found that students have poorer learning

outcomes when paired randomly, rather by ability [3]. Students were required to switch role for each step [7].

B. Object-Oriented Programming (OOP)

Computer programming I and II courses are introductory courses, teaching to the first year CS students (CS1). They are procedural programming, but OOP is a new programming paradigm. Lecture is still and efficient way explains new concepts and construct. However, programming is best learned through practice and experience [3]. Thus, it is indicated that the second year CS students (CS2) have faced two problems in the same time- their programming skill and knowledge of new programming paradigm (OOP).

OOP is a programming model that organizes program around objects, rather than function and logic. An object has unique attributes and behavior. The organization of an object-oriented program consists of objects, a programmer wants to manipulate and identify how they relate to each other. OOP focuses on the objects, but procedural programming focusses on data as input that are processed by functions and logic to produced output. The difference among them is that software design in OOP focusses around data or objects but procedural software design, focusses on functions and logic.

This OOP approach is well-suited for collaborative development, where projects are divided into groups. Thus, all production code could be written collaboratively with a partner. This indicates pair programming is highly beneficial in OOP class.

Thus, both lecturing and collaborative in-class activities are necessarily used in OOP class. Lecturing is needed to provide an overview of OOP concepts and construct-what is it and what to do. In-class activity enables students work with OOP-how to do- and technical aspects of the new content in a complex, local environment such as a laboratory [7]. The techniques of pair programming were implemented for this part of collaborative in-class activity.

IV. METHODOLOGY

This study was carried out with a second year CS class (CS2) in Rangsit University, Thailand, taking in object-oriented programming (OOP). There were a total number of 28 students enrolled in fall semester, 2020

A. Design

This study is based on both OOP concepts and programming skill. The researcher designed the study, shown as timeline in Fig. 1.

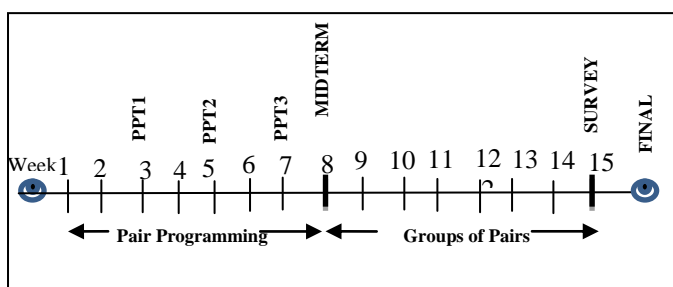


Fig.1 shows that the study was divided into 4 parts: pair programming, groups of pairs (groups work), end-of-course survey, and individual OOP concepts and construct.

The study was employed with pair programming and group of pairs. Students' performance was assessed through pair programming tests (PPTs) and written examinations. Measures of attendance programming confidence and perceptions of the pair programming and group of pairs were also examined by end-of-course survey.

B. Procedure

Part1 Pair Programming:

Changes to a new programming paradigm, such as OOP has needed lecturing that is the most effective in transmitting information as what to do. While lecture was taking, pair programming has been exercised to learn as how to do. Each class was employed in OOP concepts and constructs with pair programming practice during 4-hours closed lab period. Prior to beginning the pair programming, guidelines for implementing pair programming in the classroom were introduced to students in step by step.

An important decision was that how to pair up the students at the beginning of course. In this study, students chosen their own pairs (closed-friend) for the whole term and the students were required to switch role for every week. Here was hoped that letting students choose their own pairs would be pleased and be willing to work and sit next to each other as they work. Interestingly, it was found that students agree that their skills were well matched and they got along with their partner (as measured by randomly interviews in class).

Seven weeks before midterm, 14 pairs of students were assigned to pair programming tests (PPT) for every a couple of weeks. Thus, three PPTs – PPT1, PPT2, and PPT3 had conducted on week 3, 5, and 7, respectively-see timeline in Fig. 1.

Part 2 Groups of Pair Programming:

After individual written midterm examination in week 8, a couple of pairs were grouped to be a software project team, working on group programming or group work. The strategy for grouping students into a team-four students from two pairs, would be to ask them to rate their pair's programming ability and assign partner in pair based their rating-weaker pair matched with stronger pair.

The hope here was that weaker pair could learn more from stronger pair in project team. Students would have the opportunity to meet more of their peers and to benefit from their added knowledge. The use of group of pairs programming in practical laboratory would facilitate peer learning and enhance students' confidence in their programming ability, although Braught et.al [1] disagreed with mismatched pairs (stronger student matched with weaker student). It means that top students do not want to be "slow down" by another student and do not see benefit in teaching others [8]. But in this study, an adaptation occurred with a group of pairs programming exercises when it was observed that the weaker pair consulted with stronger pair in their group.

Part 3 End-Of-Course Survey:

At the last class (week 15), survey was handed out in lectures to obtain information on students' perceptions of pair programming and group of pairs. This required students to make decision on their agreements regarding issues such as enjoyment of programming, self-efficacy of programming, time spent programming, and confidence at writing a program to solve a real world problem (software project).

These surveys questioned students about their attitudes toward the course, their experience in the course, and also their perceptions of their own ability.

Part 4 Individual OOP Concepts and Construct Assessment:

Individual written midterm and final exams were conducted on week 8 and 16, respectively. Both exams were designed to evaluate individual OOP concepts and construct. Midterm is as pre-test before group working through pairs programming in group. Final exam is as post-test after then. The focus has been on trying to determine the effect pair programming and group of pairs on student performance as measured by exam performance and the quality of the software team's product, they submitted.

V. RESULTS

This study was carried out with a CS2 class, taking in OOP. There were a total number of 28 students enrolled in fall-semester, 2020. A quick examination of the comparison among 14 PPTs (as measured by time to complete program and the completeness and correctness of program) on week 3, 5, and 7 before midterm was shown in Fig. 2.

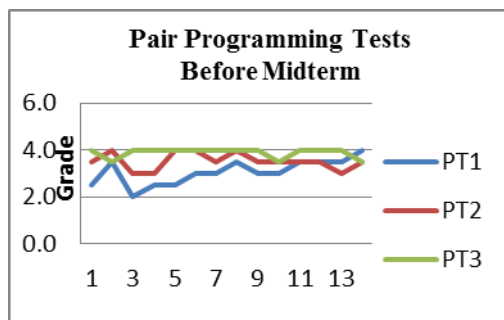


Fig. 2 Three-pair programming tests among 14 pairs

Fig. 2 Three-pair programming tests among 14 pairs. The graphs in Figure 2, shows that PPT1 line appearing in the bottom. PPT2 line is the middle and PPT3 line is on the top. This means that pair programming consistently leads to improve grades of pair programming tests.

After then, written midterm exam was conducted to determine individual OOP concepts and construct. All of 28 students were recorded midterm scores.

From literatures, it is found that in Extreme Programming (XP), an emerging software development methodology, all production code is written or worked on collaboratively by a pair of developers. Thus, after midterm, seven weeks were spent for software development project in OOP through a group of pairs. In this study, seven teams (be formed by a couple of pairs-4 students a team) were assigned software development projects. Each team had to work together on all designs in OOP, and then be allocated to each pair in team to code in pair programming. After then, they have learned to

merge all classes (codes) to form a complete project. Finally, software product from each team was submitted and be shown that students have good learning outcomes when paired the weaker pairs with the stronger pairs. The weaker pair could learn and consult with stronger pair in their group, working toward to a common goal.

This was proved by the graph of a comparative of midterm and final scores of all of 28 students in class, shown in Fig. 3.

Fig. 3 shows that final scores line is higher than midterm scores line. And the average midterm and final scores are 36.2 and 43.3 percent, respectively. This means that students has progressed or grown up in individually knowledge and programming ability, success in the course.

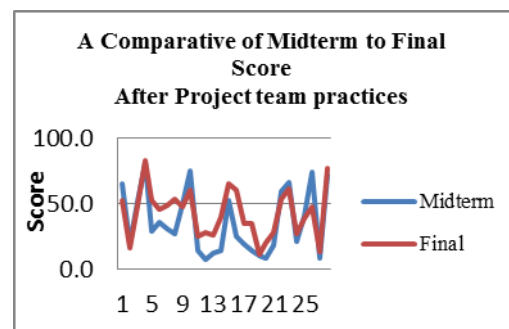


Fig. 3 A Comparative of Midterm and Final Exams

The findings provide direct evidence that the use of pair programming in group working aids in the development of individual OOP concepts and construct.

Finally, survey was conducted in the last lecture. The total number of 28 students was taking in OOP course. One of them was absent class. Thus, 27 students were asked each question to express their agreement with pair programming and group of pairs in XP software development methodology. The findings in this survey indicates that the use of pair programming and group of pairs gave benefits include more self confidence in OOP, increased knowledge and programming skill, less time to complete programming projects, reduced stress in programming work, more enjoyable in OOP, and quicker in error findings, provided in Table 1.

TABLE 1 PERCENTAGE OF AGREEMENT OF RESPONSES

Issues	Perceptions of agreements (%)
More confidence at writing a program	85.2
Increase knowledge and programming skill	92.6
Less time to complete programming projects	92.6
Reduce stress in programming work	88.9
More enjoyment of OOP	92.6
Quicker in error finding	85.2

Furthermore, the last question required students to rate their agreement on a five-point Likert scale regarding issues as self-efficacy at pair programming. The average of scale was 4.07.

VI. CONCLUSION

The shortage of CS graduates in Thai workforces has been appeared, because of declining enrollment [8]. The perceptions of CS as a “difficult” subject discourage students from choosing to study it [5]. Students often find computer programming challenging, resulting in a drop-out rate of approximately 25% [3]. This study employs a collaborative programming technique referred to as pair programming. A wide range of benefits through pair programming exercises in programming class, have been reported such as more confidence at program writing, increased programming skill, decreased time to complete program, reduced stress and enhanced learning.

This study was carried out with CS2 class, taking in OOP, which is a new programming concept for CS2 students. Therefore, students had been struggled with two issues as shifting programming paradigms and programming skill. Pair programming and group of pairs were utilized in OOP class. The finding indicates that letting students choose their own pair with their closed-friend was a much more enjoyable experience and less stress in pair programming. However, this study exercised XP methodology in software project team. All production code in group team was written collaboratively with a partner. There were two pairs (4 students) forming a team-stronger pair matched with weaker pair. The weaker pair could learn and consult with stronger pair. In order to prove the success of group of pair programming, written midterm and final exams have shown that group of pairs (multiple pairs) in XP methodology leads to improve scores and increases student satisfaction. This

means that utilizing pair programming in CS-OOP class effects on the development of individual programming skill.

At the last activity, the survey was conducted and the finding indicates that pair programming could be utilized in CS programming class, in particular OOP concepts and construct.

Therefore, it is a greater confidence that pair programming and group of pairs programming were important keys of the success of individual programming skill. This would be a benefit, appear to help increase retention in CS in the future.

REFERENCES

- [1] B. Grant, W. Tim, and L. M. EBy, “The Case for Pair Programming in the CS Classroom”, *ACM Transaction on Computing Education*, Vol. 11 No. 1, 2011, pp. 2.
- [2] M. Charlie, H. Brian, and W. Linda, “Experimenting with Pair Programming in the Classroom”, *ACM SIGCSE Bulletin*, Vol. 35, No. 3, 2003, pp. 60-64.
- [3] M. Phil, M. Rebecca, H. Phylip, and M. Patrick, “Enhancing Collaborative Learning Using Pair Programming :Who Benefits?”, *AISHE-J*, Vol. 6, No. 2, 2014, pp. 1411-14125.
- [4] T. Wanwipa, “An Impact of CS Undergraduate Programming Skill on Career Development”, *International Journal of Computer Science and Software Engineering*, Vol. 9, No. 3, 2020, pp. 60-63.
- [5] T. Wanwipa, “Efforts to Revitalize the Computer Science Education in Thailand”, *International Journal of Engineering Research and Applications*, Vol. 6, No. 2, 2016, pp. 34-41.
- [6] W.A.Laurie, and K.R. Robert, “Experimenting with Industry’s Pair-Programming Model in the Computer Science Classroom”, *ACM SIGCSE Bulletin*, Vol. 35, No. 3, 2003, pp. 60-64.
- [7] W. Keith, and Y. Michael, “Active Learning for Classroom Management Model”, in *Proceeding of the Informing Science and IT Education conference*, 2008.
- [8] W. Laurie, M. Scott, L. Lucas, and H. Khaled, “Eleven Guidelines for Implementing Pair Programming in the Classroom”, in *Agile 2008 Conference*, 2008.