LOOKING INTO THE FREEDOM OF PARTNER CHOOSING IN PAIR PROGRAMMING

Soo See Chai[1,*], Kok Luong Goh[2]

[1]*Department of Computing and Software Engineering, University Malaysia Sarawak, 93400 Kota Samarahan, Sarawak, Malaysia*

[2]*International College of Advanced Technology Sarawak (i-CATS), Kuching, Sarawak, Malaysia*

[*]For correspondence; Tel. + (60) 82-583637, E-mail: sschai@unimas.com

[2] E-mail: klgoh@ppks.edu.my

ABSTRACT: *The published research studies to date indicate that pair programming has a positive impact on some aspects of students' performance. In the normal practice of pairing programming in the academic field, the students were paired by assigning partners according to their level of programming skill. In another words, students were paired according to their programming compatibility that were perceived by their lecturers. However, research studies did not attempt to identify the main element that the students are looking into when they are given the freedom to select their partner in pair programming practice. An experiment with 76 students during a one-week programming workshop shows that 59.2% will choose their partner according to gender while 30.3% will choose their partner based on the ethnics group. The study shows that, only 5.2% of the students focus on the skills of their choice of partner. At the end of the workshop, 96% of the students agree that pairing with a partner helps them in solving programming problem. However, only 89.2% of the students prefer to work in pairs when solving programming while 5.4% prefer to work as individual. This initial finding tallies with the other research whereby it shows that pair programming benefits the students in solving programming problem. Despite the normal belief that the pairs are compatible if they are almost the same level in terms of technical competency in programming, students tend to choose according to gender when they are given a choice.*

Keywords: Pair programming, partner, gender, programming course, higher education, ethnics

1. INTRODUCTION

Pair programming, like its name implies, involves two programmers, sharing a single workstation, working together on a single task. One programmer plays the role of "driver" who is responsible for operating the resources, i.e. computer, keyboard and mouse. The other programmer, called "navigator" or "observer" will observe the work by the driver and offers suggestions, advices and corrections to both the design and code [1]. The pair will alternate their roles after a certain period of time throughout their work and therefore, both programmers share responsibility for all aspects of the program development [2]. This is not a new idea and has started at least from 1970 [3]. Pair programming practice has been long being used in software engineering industries and efficient results were achieved [4]. In the academic field, such practice has shown good result as well. The published research studies to date indicate that pair programming has a positive impact on some aspects of students' performance. The interest [5], enjoyment [6, 7], confidence [6, 8] and retention rate [9] of the students in learning programming had been reported to be increased and their success rate in continuing the successive programming courses in their undergraduate studies had greatly improved [10].

In the common practice of pairing programming in the academic field, the students were paired by assigning partners according to their level of programming skill. In another words, students were paired according to their programming compatibility that were perceived by their lecturers. Research shows that, students who are paired based on their technical competency produce the most compatible pairs [11]. There are also instructors who allowed their students to choose their own partners in pair programming [12, 13]. The main concern of these studies focused on the compatibility of the chosen partner. There is no research currently reporting on the elements that the students involving in pair programming are looking into when they select their own partner. This study intends to fill up this gap by looking into the elements that the students will look into when they are given the choice to select their own partner.

The remainder of this paper is organized as follows: Previous Studies will present a brief summary of the different partner assignment in pair programming. The actual experiment method carried out in this research is next presented in the Pair Programming Experiment section. Next, the results and analysis are explained. At the end of this paper, the conclusions obtained from this research work are delivered.

2. PREVIOUS STUDIES

Most of the pair programming research was conducted by assigning the students with a partner. Different ways were used in partner assignment. Some of the criteria used when assigning partners included ability or technical competency and personality type. Watkins and Watkins [14], for example, paired the students in the lab according to the performance of the students in the previous labs and also individual test performance. With students undertaking different majors, Radermacher and Walia [15] paired the students based on their majors. They also paired up the students who were taking Introductory Computer Science course based on similar grades on selected subjects. Some of the researchers used random assignment method whereby the assignment of partner was not based on any criteria. In the research carried out by Braught, Wahls and Eby [2], pairing for the first few labs were assigned randomly. However, in the later labs, the pairing matched students of similar ability based on the performance in the course to that point. In a virtual environment setting whereby the students collaborated in pairs via online technologies, Zacharis [16] in his research, assigned students with approximately equal knowledge and ability based on the grades of the four previous assignments. Besides from knowledge and ability, personality traits were also used as a guideline in pairing the students in pair

programming. Students were normally given personality test and then they will be paired up with partners of similar or totally different personality [17-20].

Besides from assigning partners from the course instructor, some researchers allow the students to choose their own partner. In a freshman programming class at University of California at Santa Cruz (UCSC), students were asked to turned in a list of three other students in the order of pairing preference [21-23]. The students' preferences were turned into a weighed graph and paired accordingly. Those who are not selected would be paired randomly. The same method of allowing the students to list their preferred partner was done for pair programming in a middle school game programming course [24].

3. PAIR PROGRAMMING EXPERIMENT

Introduction to Programming, TMF1414, provides a general introduction of programming to the students in the Faculty of Computer Science and Information Technology (FCSIT), University Malaysia Sarawak (UNIMAS) in Kota Samarahan, Sarawak, Malaysia. The course is designed to give students the ability to write simple console programs and to be able to understand programs written by others. This course forms part of the core courses for the undergraduate programs within the faculty and provide programming foundation for other courses in the faculty. The course covers problem solving skills, writing algorithm, basic programming syntax, control structures, functions and input/output operations.

A five-day programming workshop from 24 October to 28 October 2016 was held during the one week mid-semester break. During the first four days of the workshop, a mixed of lecture and lab was held within the 2 hours session. The students executed the codes samples given and next modified and extend the functionalities of these sample programs after explanations are given and goals set by the instructor. A programming exercise which covered the knowledge of the topics learnt in each day was given at the end of each session. A total of 76 students completed the 5 days workshop. Some of the students volunteered to participate in the workshop while the others were asked to join the workshop by their respective instructors. It is important to note that, all the students had attended 7 weeks of normal lecture and lab before the workshop. During the semester, the students who took this course would attend 2 hours of lecture and 2 hours of lab per week. As lab normally started in Week 3, the students had basically attended 14 hours of lecture and 10 hours of lab before the workshop. The workshop covered all the topics that were already taught during the first half of the semester.

During the first four days of the workshop, the students did all the given tasks individually. The students who attended the workshop were informed on the first day of the workshop that they will need to attend a lab test on the last day of the workshop. They were also well informed that whether they did well or not in the lab test on the last day would not contribute any marks to their final grade of the course. On the last day of the workshop, the students were given a lecture for about 30 minutes on pair programming. They were asked to find their partners after the brief lecture and decide on the workstation that they would use. The students were next asked to decide on the "driver" and "navigator" in their pair works. Next, the students were given 30 minutes to solve the programming exercise given. Under the supervision of the instructor, the students were asked to change their role after 15 minutes. At the end of the 30 minutes, the instructor discussed and provided a suggested solution to the programming exercise given and the students are asked to fill in a questionnaire.

4. SURVEY RESULTS

4.1 Personal Data

The first part of the questionnaire collects some personal data of the students: gender, age, nationality and ethics group.

a. Gender

There were altogether 76 students participated in this workshop with 33 male and 43 female students. The distribution of the gender of the students is shown in Table 1.

Table 1: Frequency of students' gender

Gender	Frequency	Percentage (%)
Male	33	43.4
Female	43	56.6

b. Age

The candidates were asked to choose from the three range of age groups: less or equal to 20, between 21 to 30 and more than 30 years old. 82.9% of the participants were in the age group of less or equal to 20 years old while the others (17.1%) were in between 21 to 30 years old. The age distribution according to gender can be seen in Table 2.

Table 2: Age distribution according to gender

Gender	Age Group	Frequency	Percentage (%)
Male	<=20	23	30.3
	21 to 30	10	13.2
Female	<=20	40	52.6
	21 to 30	3	3.9

c. Nationality

There are 4 male non-Malaysians among the participants. Three (3) of them aged less than equal to 20, while another was between 21 to 30 years old.

Table 3: Malaysian participants' ethnics group distribution according to gender

Gender	Ethnic Group	Frequency	Percentage (%)
Male	Malay	9	12.0
	Chinese	11	14.7
	Indian	2	2.7
	Others	11	14.7
Female	Malay	19	25.3
	Chinese	17	22.7
	Indian	0	0
	Others	6	8

d. Ethics Group

Among the 72 Malaysian participants, 28 of them were Malay, 28 Chinese, 2 Indian, 13 other races and 1 did not state her ethics group. The country of origin and the ethnic groups of the non-Malaysians were unknown. Table 3 shows the distribution of the Malaysian participants' ethnics group according to the gender.

4.2 Previous Programming Experiences

This part of the questionnaire will look into the self-perceived programming skill level of the participants and also the programming experiences of the participants in terms of years.

a. Self-perceived programming skill level

A total of 61 students perceived themselves as beginner while 15 students perceived themselves as moderate programmers. None of the participants think that they are at the expert level in programming. Table 4 shows the distribution of the self-perceived programming skill level according to gender.

Table 4: Self-perceived programming skill level according to gender

Gender	Skill level	Frequency	Percentage (%)
Male	Beginner	31	40.8
	Moderate	2	2.6
Female	Beginner	30	39.5
	Moderate	13	17.1

b. Programming experiences

In terms of programming experiences, the participants were asked on the number of years that they have been programming. Three options were available: Less than a year, Between 1 to 3 years and More than 3 years. 59 students have been programming for less than a year, 14 between 1 to 3 years while 3 participants had more than 3 years of programming experiences. Table 5 shows the distribution of programming experiences according to the gender.

Table 5: Programming experiences according to gender

Gender	Experiences	Frequency	Percentage (%)
Male	Less than 1 year	30	39.5
	Between 1 to 3 years	3	3.9
	More than 3 years	0	0
Female	Less than 1 year	29	38.2
	Between 1 to 3 years	11	14.5
	More than 3 years	3	3.9

4.3 Pair Programming

This section of the questionnaire surveys on whether the participants think that pair programming help them in solving programming problem, the way they choose their partner and whether they prefer to use pair programming when solving programming problem.

a. Does pair-programming help?

All the male students agree that pairing with a partner help them in solving programming problem while 3 female students disagree with this. All these three female students perceived themselves as beginner in programming and has less than 1 year of programming experiences. The distribution of this is shown in Table 6.

Table 6: Usefulness of pair programming according to gender

Gender	Pair programming help?	Frequency	Percentage (%)
Male	Yes	33	43.4
	No	0	0
Female	Yes	40	52.6
	No	3	4.0

b. How do you choose your partner?

A total of 45 students (59.2%) chose their partner according to the gender of their partner. Among these, 42 of the participants will choose their partner of the same gender while another 3 chose partner of different gender. The distribution of this is shown in Table 7.

Table 7: Choose partner based on gender

Gender	Gender	Frequency
Male	Same	13
	Different	1
Female	Same	29
	Different	2

There was 23 (30.3%) students who would look into the ethnics of the partner when they choose their partner. This is shown in Table 8.

Table 8: Choose partner based on ethnics group

Gender	Gender	Frequency
Male	Same	8
	Different	1
Female	Same	10
	Different	4

A total of 31 (40.8%) students did not look into the gender and ethnics factor when choosing their partner. These students choose their partner based on skills of their partner (12.9%), friends (35.5%), someone who sits beside them (12.9%), not focusing on any criteria (22.5%) and other unknown reasons (16.1%).

c. Preference of solo or pair programming?

The participants are asked to choose whether they prefer to work individually or in pairs when solving programming problem. 74 participants had responded with either yes or no in their answer while another two participants chose both yes

and no. Among the 74 respondents, 66 (89.2%) preferred to work in pairs while 8 (10.8%) preferred to work individually. Table 9 shows this distribution according to gender.

Table 9: Preference of solo or pair programming according to gender

Gender	Solo/Pair	Frequency
Male	Solo	4
	Pair	29
Female	Solo	4
	Pair	37

5. ANALYSIS

From the personal data obtained, the number of female and male participants was almost the same. The participants were all less than 30 years old from different ethnics group. There were also some foreigner participants. Majority of the candidates (80.3%) considered themselves as the beginner level of programming and 77.7% of these participants has less than 1 year of programming experience. Pair programming has helped the participants in solving programming problems (96%). This is important as most of the candidates were in the beginner stage of programming. From the 33 male and 43 female participants, if the participants choose to work on same gender, there would be 15 and 20 pairs of male and female participants respectively, with another 3 pairs of different gender. From the results obtained, there were only 3 participants who would choose their partner of different gender. The possible reason behind this would be, the candidates feel that the partner that they chose were compatible and they were comfortable with their partner. The top 3 criterions that the participants would look into when they chose their partner can be ranked in the following order: gender, friends and ethnics. Only 12.9% of the candidates focused on the skill of the chosen partner when they were working in pair. This is totally different from the pair assignment done by the instructor or lecturer whereby practical skill is always the main concern in deciding the pair work.

6. CONCLUSIONS

This study fills the need to better understand the pair assignment especially in the context of Malaysian students under which pair programming is more beneficial than individual programming for the programming freshman. The study looks into the criterions that the freshman will focus into when they are self-pairing. From this initial study, it is found that, the candidates agreed that pair programming helps them in solving programming problems and they preferred to do pair programming after they were self-paired. The main reason for this is that, the candidates who were self-paired were compatible with their partner. However, this study did not work on the impact of self-pairing by looking into the achievements of the participants in terms of marks or grades obtained before and after the pair programming practise. In addition, in this study, the participants were only allowed to self-pair once. A thorough study on whether the participants will select the same partner and whether the same criterions will be used to select their partner in the future programming practices would be needed to verify the initial results obtained in this study.

ACKNOWLEDGEMENT

This research work is supported by the UNIMAS Cross Disciplinary Research Project with the grant number: F08/CDRG/1832/2019.

REFERENCES

1. Salleh, N., E. Mendes, and J.C. Grundy, *Empirical studies of pair programming for CS/SE teaching in higher education: a systematic literature review.* IEEE Transactions on Software Engineering, 2011. **37**(4): p. 509-525.
2. Braught, G., T. Wahls, and L.M. Eby, *The case for pair programming in the computer science classroom.* ACM Transactions on Computing Education (TOCE), 2011. **11**(1): p. 2.
3. Jensen, R., *A pair programming experience.* CrossTalk, 2003. **16**(3): p. 22-24.
4. McChesney, I. *Three years of student pair programming: action research insights and outcomes*. in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. 2016. ACM.
5. Liebenberg, J., E. Mentz, and B. Breed, *Pair programming and secondary school girls' enjoyment of programming and the subject Information Technology (IT).* Computer Science Education, 2012. **22**(3): p. 219-236.
6. Freeman, S.F., B.K. Jaeger, and J.C. Brougham, *Pair programming: More learning and less anxiety in a first programming course.* age, 2004. **8**: p. 1.
7. Gupta, S., V. Bhattacharya, and M. Singha, *Pair Programming" Potential Benefits and Threats"*. International Journal of Advanced Computer Research, 2013. **3**(1): p. 108.
8. Balijepally, V., *Task complexity and effectiveness of pair programming: an experimental study.* 2007.
9. Brereton, P., M. Turner, and R. Kaur. *Pair programming as a teaching tool: a student review of empirical studies*. in *2009 22nd Conference on Software Engineering Education and Training*. 2009. IEEE.
10. McDowell, C., L. Werner, and H. Bullock, *Pair programming improves student retention, confidence, and program quality.* Communications of the ACM, 2006. **49**: p. 8.
11. Carver, J.C., L. Henderson, L. He, J. Hodges, and D. Reese. *Increased retention of early computer science and software engineering students using pair programming*. in *20th Conference on Software Engineering Education & Training (CSEET'07)*. 2007. IEEE.
12. Chaparro, E.A., A. Yuksel, P. Romero, and S. Bryant, *Factors Affecting the Perceived Effectiveness of Pair Programming in Higher Education.* PPIG, 2005.
13. Jacobson, N. and S.K. Schaefer, *Pair programming in CS1: overcoming objections to its adoption.* ACM SIGCSE Bulletin, 2008. **40**(2): p. 93-96.
14. Watkins, K.Z. and M.J. Watkins, *Towards minimizing pair incompatibilities to help retain under-represented groups in beginning programming courses using pair*

programming. Journal of Computing Sciences in Colleges, 2009. **25**(2): p. 221-227.
15. Radermacher, A.D. and G.S. Walia. *Investigating the effective implementation of pair programming: an empirical investigation*. in *Proceedings of the 42nd ACM technical symposium on Computer science education*. 2011. ACM.
16. Zacharis, N.Z., *Measuring the effects of virtual pair programming in an introductory programming java course*. IEEE Transactions on Education, 2010. **54**(1): p. 168-170.
17. Chao, J. and G. Atli. *Critical Personality Traits in Successful Pair Programming*. in *Proceedings of the conference on AGILE 2006*. 2006. IEEE Computer Society.
18. Choi, K.S., F.P. Deek, and I. Im, *Exploring the underlying aspects of pair programming: The impact of personality*. Information and Software Technology, 2008. **50**(11): p. 1114-1126.
19. Sfetsos, P., I. Stamelos, L. Angelis, and I. Deligiannis, *An experimental investigation of personality types impact on pair effectiveness in pair programming.* Empirical Software Engineering, 2009. **14**(2): p. 187.
20. Salleh, N., E. Mendes, and J. Grundy, *Investigating the effects of personality traits on pair programming in a higher education setting through a family of experiments.* Empirical Software Engineering, 2014. **19**(3): p. 714-752.
21. Bevan, J., L. Werner, and C. McDowell. *Guidelines for the use of pair programming in a freshman programming class*. in *Proceedings 15th Conference on Software Engineering Education and Training (CSEE&T 2002)*. 2002. IEEE.
22. Werner, L.L., B. Hanks, and C. McDowell, *Pair-programming helps female computer science students.* Journal on Educational Resources in Computing (JERIC), 2004. **4**(1): p. 4.
23. McDowell, C., L. Werner, H.E. Bullock, and J. Fernald, *Pair programming improves student retention, confidence, and program quality.* 2006.
24. Denner, J., L. Werner, S. Campe, and E. Ortiz, *Pair programming: Under what conditions is it advantageous for middle school students?* Journal of Research on Technology in Education, 2014. **46**(3): p. 277-296.

*For correspondence; E-mail:sschai@unimas.my