Perceptions of Pair Programming: Boon or Bane? A student survey

Premalatha Sampath
Whitireia Community Polytechnic
Premalatha.Sampath@whitireia.ac.nz

ABSTRACT

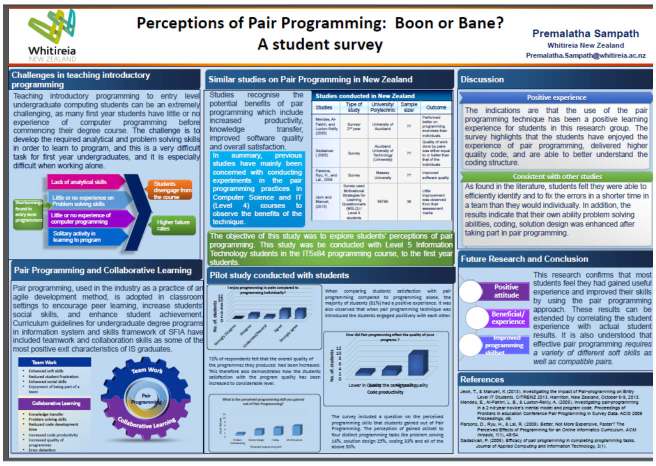
Teaching introductory programming to entry level undergraduate computing students can be an extremely challenging task, as many first year students have little or no experience of computer programming before commencing their degree course. The challenge is developing the required analytical and problem solving skills in order to learn to program. This can prove to be a very difficult task for first year undergraduates, and it is especially difficult when programming alone. The method of teaching programming is to enable two students to work collaboratively on a single computer on the same program or piece of code, a technique known as pair programming. Industry's pair programming methodology has been adopted in academic settings as a form of collaborative learning. The objective of this study is to explore students' perceptions of the impact of pair programming and the influence of pair programming on student's learning experience.

Keywords: Pair Programming, Collaborative Learning.

1. INTRODUCTION

Pair Programming is an approach used to assist first year programming students to overcome a perception that programming is difficult (Smith & Delugach, 2010). Anecdotal evidence suggested that first year programming students find the challenge unbearable, leading these students to potentially disengage from the course.

Pair Programming practice used in industry as an agile development method has been widely implemented in education because of the benefits it brings to entry-level students in tertiary education. Pair programming is the term used to describe the process by which two programmers work side by side, on the same task at one computer designing and coding the same algorithm. Typically two roles in pair programming: a 'driver', who actively types at the computer or records a design; and a 'navigator', who watches the work of the driver and attentively identifies problems and makes suggestions. The two programmers collaborate in designing, coding and reviewing.



Perceptions of Pair Programming: Boon or Bane?
A student survey

Premalatha Sampath
Whitireia New Zealand
Premalatha.Sampath@whitireia.ac.nz

Challenges in teaching introductory programming

Teaching introductory programming to entry level undergraduate computing students can be an extremely challenging, as many first year students have little or no experience of computer programming before commencing their degree course. The challenge is to develop the required analytical and problem solving skills in order to learn to program, and this is a very difficult task for first year undergraduates, and it is especially difficult when working alone.

Shortcomings found in entry level programmers
Lack of analytical skills
Little or no experience on Problem solving skills
Little or no experience of computer programming
Solitary activity in learning to program
Students disengage from the course
Higher failure rates

Pair Programming and Collaborative Learning

Pair programming, used in the industry as a practice of an agile development method, is adopted in classroom settings to encourage peer learning, increase students' social skills, and enhance student achievement. Curriculum guidelines for undergraduate degree programs in information system and skills framework of SFIA have included teamwork and collaboration skills as some of the most positive exit characteristics of IS graduates.

Team Work
- Enhanced soft skills
- Reduced student frustration
- Enhanced social skills
- Enjoyment of being part of a team

Collaborative Learning
- Knowledge transfer
- Problem solving skills
- Reduced code development time
- Increased code productivity
- Increased quality of programmes
- Error detection

Similar studies on Pair Programming in New Zealand

Studies recognise the potential benefits of pair programming which include increased productivity, knowledge transfer, improved software quality and overall satisfaction.

In summary, previous studies have mainly been concerned with conducting experiments in the pair programming practices in Computer Science and IT (Level 4) courses to observe the benefits of the technique.

Studies conducted in New Zealand

Studies	Type of study	University/ Polytechnic	Sample size/	Outcome
Mendes, Al-Fakhri, and Luxton-Reilly (2005)	Survey/ 2nd year	University of Auckland	??	Performed better on programming exercises than individuals.
Sedasivan (2005)	Survey	Auckland University of Technology (University)	??	Quality of work done by pairs was either equal to or better than that of the individuals
Parsons, Ryu, H., and Lal., 2008	Survey	Massey University	??	Improved software quality
Jeon and Manuel, (2013)	Survey used Motivational Strategies for Learning Questionnaire (MSLQ) / Level 4 students	56789	38	Little improvement was observed from their assessment marks

The objective of this study was to explore students' perceptions of pair programming. This study was be conducted with Level 5 Information Technology students in the IT5x84 programming course, to the first year students.

Pilot study conducted with students

When comparing students satisfaction with pair programming compared to programming alone, the majority of students (81%) had a positive experience. It was also observed that when pair programming technique was introduced the students engaged positively with each other.

72% of respondents felt that the overall quality of the programmes they produced had been increased. This therefore also demonstrates how the students satisfaction with the program quality has been increased to considerable level.

What is the perceived programming skill you gained out of Pair Programming?
No.of Students
Problem Understanding
Solution Design
Coding
All of the above
Perceived programming skill

The survey included a question on the perceived programming skills that students gained out of Pair Programming. The perception of gained skillset to four distinct programming tasks like problem solving 16%, solution design 25%, coding 33% and all of the above 50%.

Discussion

Positive experience

The indications are that the use of the pair programming technique has been a positive learning experience for students in this research group. The survey highlights that the students have enjoyed the experience of pair programming, delivered higher quality code, and are able to better understand the coding structure.

Consistent with other studies

As found in the literature, students felt they were able to efficiently identify and to fix the errors in a shorter time in a team than they would individually. In addition, the results indicate that their own ability problem solving abilities, coding, solution design was enhanced after taking part in pair programming.

Future Research and Conclusion

This research confirms that most students feel they had gained useful experience and improved their skills by using the pair programming approach. These results can be extended by correlating the student experience with actual student results. It is also understood that effective pair programming *requires a variety of different soft skills as well as compatible pairs.*

References

Jeon, T., & Manuell, K (2013). Investigating the Impact of Pair-programming on Entry Level IT Students. CITRENZ 2013. Hamilton, New Zealand, October 6-9, 2013.
Mendes, E., Al-Fakhri, L. B., & Luxton-Reilly, A. (2005). Investigating pair-programming in a 2 nd-year novice's mental model and program code. Proceedings of Frontiers in education Conference Pair Programming in Survey Data. ACIS 2008 Proceedings, 46.
Parsons, D., Ryu, H., & Lal, R. (2008). Better, Not More Expensive, Faster? The Perceived Effects of Programming for an Online Informatics Curriculum. *ACM Inroads, 1*(1), 48-54.
Sadasivan, P. (2005). Efficacy of pair programming in completing programming tasks. Journal of Applied Computing and Information Technology, 3(1).

2. PAIR PROGRAMMING AND COLLABORATIVE LEARNING

Curriculum guidelines for undergraduate degree programme in the information system (IS 2010) and Skills Framework for the Information Age (SFIA) identified (Institute of IT Professionals New Zealand, 2016) that teamwork and collaboration skills are some of the main exit characteristics of IS graduates. Collaborative learning involves joint intellectual effort by groups of students working together. Pair programming is a form of collaborative learning. There has been a significant amount of research on pair programming as a collaborative learning tool in the traditional classroom. Collaborative learning is also considered one of the main benefits of pair programming in both professional and educational settings. Several studies have indicated that students perceive they learned more by working with a partner than they would have by working alone (Braught, Wahls, & Eby, 2011; Carver, Henderson, Lulu , Hodges & Reese 2007).

3. TRANSFORMING STUDENTS LEARNING

Faja (2014) indicated that students perceived paired programming beneficial in terms of learning and enjoyment. A number of studies have also shown that pair programming improved the quality of the program (Chigona & Pollock, 2008; Zacharis, 2011). Zacharis (2011) and Salleh, Mendes, and Grundy (2011) found that paired students were more productive than individual programmers and they completed tasks in a shorter amount of time. On the other hand, Parsons, Ryu and Lal (2008) found that pair programming may or may not take less time. Several studies focused on the pair formation approach. Studies adopted different pair formation techniques such as assigning pairs randomly (Faja, 2014), matching pairs based on academic performance (Zacharis, 2011), and pairs with slightly different academic levels. Several studies suggested another approach called **pair rotation** (Carver et al. 2007; Braught et al., 2011).

4. STUDIES CONDUCTED IN NEW ZEALAND

Mendes, Al-Fakhri, and Luxton-Reilly (2005) from the University of Auckland concluded that the students in Pair Programming classes performed better on programming exercises than individuals. Likewise, a study conducted by Auckland University of Technology revealed that the quality of

This poster appeared at ITx 2016, incorporating the 7th annual conference of Computing and Information Technology Research and Education New Zealand (CITRENZ2016) and the 29th Annual Conference of the National Advisory Committee on Computing Qualifications, Wellington, New Zealand, July 11-13, 2016. Michael Verhaart, Emre Erturk, Aaron Steele and Scott Morton (Eds).

work done by pairs was either equal to or better than that of the individuals, despite having spent only half the amount of time on the task as the individual programmers (Sadasivan, 2005). A similar study of Massey University students, acknowledged that there are benefits to pair programming, in particular improved software quality and therefore supporting the claims made by previous studies (Parsons et al., 2008). A study conducted by Weltec, confirmed that pair programming reduced students levels of anxiety during the programming classes, however little improvement was observed from their assessment marks (Jeon & Manueli, 2013). In summary, a number of previous studies have mainly been concerned with conducting experiments on the pair programming practices in Computer Science and IT (Level 4) courses to observe the benefits of the technique. The objective of this study was to explore students' perceptions of the impact of pair programming and the influence of pair programming on student's learning experience. This study will be conducted with Level 5 Information Technology students in the IT5x84 programming course, an entry level programming paper offered to the first year students.

5. METHODOLOGY

The students of IT5x84 Programming courses at Whitireia Polytechnic in Auckland were invited to participate in the study. The IT5x84 paper introduces the students to fundamental programming skills over 13 weeks in a Trimester. In the 13 week course, students had to complete 7 lab exercises and 3 assignments. Out of 8 lab exercises, 4 exercises were equally divided to complete their tasks individually and in pair programming. Likewise, out of 2 assignments, 1 assignment was completed individually and another assignment in pair programming. The pair programming partners were selected according to skillset. The students were paired in rotation fashion (Carver et al. 2007; Braught et al., 2011), therefore over the 13 weeks each student programmed with different partners. The survey was administered among students after they completed their pair programming tasks. The survey consisted of 16 questions asking respondents to compare their pair programming activity to the experience of programming alone, and the impact of pair programming on their learning experience. The questions were grouped into the following areas: enjoyment, problem solving, code productivity and quality, error detection, team work, knowledge transfer and overall satisfaction. 12 students completed the survey.

6. RESULTS:

The quantitative approach was utilized to analyse the results.

6.1 Enjoyment

In relation to their level of satisfaction with pair programming compared to programming alone the majority of students (81%) had a positive experience. It was also observed that when pair programming technique was introduced, the students engaged positively with each other.

6.2 Problem Solving

The survey included a question on the perceived programming skills gained out of Pair Programming. The survey shows that 16% of respondents believed that pair programming enhances problem solving skills, 25% in solution design, 33% of respondents in coding and most of the respondents acknowledged a combination of all skillsets.

6.3 Overall Satisfaction

The results highlighting the overall level of satisfaction with the peer programming technique indicated that 66% of respondents believe that pair programming enhances their programming skills. Further to this, 70% of respondents indicated that they would recommend pair programming to other students.

7. DISCUSSION

The indications are that the use of the pair programming technique created a positive learning experience for students in this research group. The survey highlights that the students have enjoyed the experience of pair programming, delivered higher quality coding, and are able to understand more in relation to coding structure. As found in the literature, students felt they were able to efficiently identify and fix errors in a shorter time in a team than they would individually (Sedasivan, 2005). In addition, the results show that the majority of students thought their programming skills were enhanced by the pair programming technique, therefore this research is consistent with other research undertaken with students at the tertiary level.

8. CONCLUSION

As a discipline, IT is constantly changing and it is relevant to introduce students' to new software development approaches in the classroom that connects with the industry. In addition, most modern organizations require individuals to work collaboratively in teams to perform their tasks. Most of the previous studies on pair programming have dealt with Computer Science students, and only a few studies have used Information Systems students. There are important differences in the skills of the students in these two disciplines as well as programming requirements. The research showed several benefits of using pair programming in academic settings such as positive learning, greater confidence in work quality, higher problem solving skills, enhanced interaction skills, and improved team building skills.

9. REFERENCES

Braught, G., Wahls, T., & Eby, L. M. (2011). The case for pair programming in the computer science classroom. *Transactions on Computing Education*, *11*(1), 2:1–2:21. http://doi.org/1921607.1921609

Carver, J. C., Henderson, L., He, L., Hodges, J., & Reese, D. (2007). Increased retention of early computer science and software engineering students using pair programming. In *Proceedings of the 20th Conference on Software Engineering Education & Training* (pp. 115–122). Washington, DC:

Chigona, W., & Pollock, M. (2008). Pair programming for information systems students new to programming: Students' experiences and teachers' challenges. In *PICMET '08 - 2008 Portland International Conference on Management of Engineering Technology* (pp. 1587–1594).

Faja, S. (2014). Evaluating effectiveness of pair programming as a teaching tool in programming courses. *Information Systems Education Journal*, *12*(6), 36–44.

Institute of IT Professionals New Zealand. (2016). IITP degree accreditation. Retrieved June 1, 2016, from http://iitp.nz/activities/accreditation

IS. (2010). *IS 2010: Curriculum Guidelines for Undergraduate Degree Programs in Information Systems.* Retrieved May 30, 2016, from https://www.acm.org/education/curricula/IS%202010%20ACM%20final.pdf

Jeon, T., & Manuele, K. (2013). Investigating the impact of pair-programming on entry level IT students. Presented at the 4th Annual Conference of Computing and Information Technology Research and Education New Zealand (CITRENZ2013), Hamilton, New Zealand: CITRENZ.

Mendes, E., Al-Fakhri, L. B., & Luxton-Reilly, A. (2005). Investigating pair-programming in a 2nd-year software development and design computer science course. In *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education* (pp. 296–300). New York, NY:

Parsons, D., Ryu, H., & Lal, R. (2008). Better, not more expensive, faster? The perceived effects of pair programming in survey data. In *ACIS 2008 Proceedings* (pp. 710–719). Christchurch: Association for Information Systems.

Sadasivan, P. (2005). Efficacy of pair programming in completing programming tasks. *Journal of Applied Computing and Information Technology*, *3*(1).

Salleh, N., Mendes, E., & Grundy, J. (2011). Empirical studies of pair programming for CS/SE teaching in higher education: A systematic literature review. *IEEE Transactions on Software Engineering*, *37*(4), 509–525.

Smith, B. J., & Delugach, H. S. (2010). Work in progress - Using a visual programming language to bridge the cognitive gap between a novice's mental model and program code. In *2010 IEEE Frontiers in Education Conference (FIE)* (pp. F3G–1–F3G–3).

Zacharis, N. Z. (2011). Measuring the effects of virtual pair programming in an introductory programming Java course. *IEEE Transactions on Education*, *54*(1), 168–170.