

# The impact of pair-programming on entry level information technology students

Terry Jeon

Wellington Institute of Technology  
Petone, Wellington  
[Terry.jeon@weltec.ac.nz](mailto:Terry.jeon@weltec.ac.nz)

Ian Hunter

Wellington Institute of Technology  
Petone, Wellington.  
[Ian.Hunter@weltec.ac.nz](mailto:Ian.Hunter@weltec.ac.nz)

## ABSTRACT

This paper outlines a study to investigate the effectiveness of Pair Programming on a small group of first-time programming students. Participants in this study were enrolled in a Level 4 Information Technology (IT) qualification at a New Zealand Institute of Technology. Numerous international studies have recommended Pair Programming as a pedagogical tool for first year tertiary level programming students. Anecdotal evidence confirms that a student's performance in programming classes at the entry level requires improvement. Consequently, it was deemed crucial to investigate the effectiveness of Pair Programming on this group of students. Participants in this study confirmed that Pair Programming reduced their anxiety during the programming classes. Factors that may have contributed to this include pairing of students with different ability levels, random pair selection, and negative pair pressure.

The outcome of this research shows the exact opposite of our hypothesis as mentioned at the beginning of this paper. We found that there are lots of variables in running pair-programming work and if we misuse them, the pair programming work can't be better performing than individual work. Therefore we are pointing out here what prevented Pair Programming from out-performing individual work and recommend better ways of running Pair Programming.

**Keywords:** Pair Programming, Motivational Strategies for Learning Questionnaire (MSLQ), first-year computing students, novice programmer, New Zealand.

## 1. INTRODUCTION

Pair Programming utilizes two students working synchronously on one exercise creating a single piece of code. With one member entering the code on a Personal Computer (PC), the other member planning future code entry, and continually checking for coding errors. Every successive session, these roles are reversed to allow each pair to appreciate each other's role.

Teague (2011) questioned "Is programming really that difficult – or are there other barriers to learning that have a serious and detrimental effect on student progression?" She concluded that Paired students outperformed non-paired students in exams.

Pair Programming is an approach to assist first year programming students to overcome a perception that programming is hard (Smith & Delugach, 2010). Anecdotal evidence suggested that first year programming students find

the challenge unbearable which leads them to 'disappear' from the course and become disengaged. For those that attempt computer programming for the first time, many find it rather a stressful experience demanding long hours spent in the computer Labs completing laboratory exercises and programming assignments.

Consequently, the researchers were interested in how Pair Programming may motivate first-time programming students at a New Zealand Institute of Technology and give the students the opportunity to share their experience of Pair Programming with us.

## 2. LITERATURE REVIEW

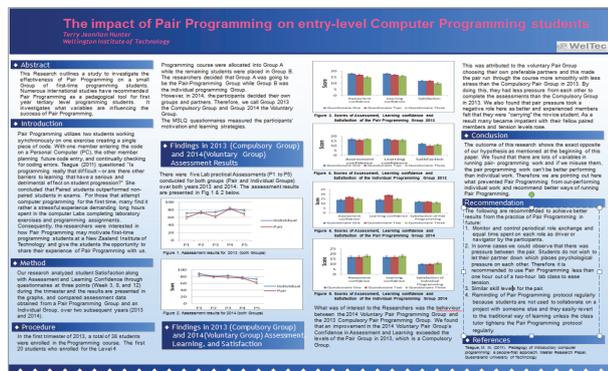
Educational benefits of Pair Programming has been recorded as improving students' retention and success rates in programming courses, success in consecutive programming courses, higher quality of programmes produced by Pair Programming, confidence in solutions, and an improved attitude towards team work.

Mendes, Al-Fakhri, and Luxton-Reilly (2005) from the University of Auckland concluded the students in Pair Programming classes performed better on programming exercises than individuals.

Similarly, William (2007) details benefits for both teachers and students that Pair Programming can bring from a stronger social relationship between learners, less delay times in striving to complete the required work, being exposed to a new experience, and a sense of achievement. At the same time developing confidence and increasing academic completion levels across each paper on which he used this method.

## 3. METHOD

Our research analyzed student Satisfaction along with Assessment and Learning Confidence, and compared the feedback data obtained from a Pair Programming Group and



This poster appeared at the 6<sup>th</sup> annual conference of Computing and Information Technology Research and Education New Zealand (CITREnz2015) and the 28<sup>th</sup> Annual Conference of the National Advisory Committee on Computing Qualifications, Queenstown, New Zealand, October 6-9, 2015. Michael Verhaart, Amit Sarkar, Rosemarie Tomlinson and Emre Erturk (Eds).

an Individual Group, over two subsequent years (2013 and 2014).

This data was obtained by using three questionnaires from the Motivational Strategies for Learning Questionnaire (MSLQ). It was based on a general cognitive view of Motivation and Learning Strategies which resulted in the creation of two sub sections of the questionnaire, namely the Motivational and the Learning Strategies Sections.

Clark, Ernst, and Scales (2009) claim that Motivational Strategies for Learning Questionnaire enabled researchers to better understand the students' Motivation and Learning Strategies in any chosen course of study.

We investigated what effects Pair Programming had on the students Assessment Confidence, Learning Confidence, and Satisfaction for both groups over a 14 week period. Seeking some measure of their ability that could gauge that Pair Programming did help the younger students to maintain interest and gain confidence to continue with programming into the future.

One research was undertaken in 2013 and another study was done in 2014, each comparing assessment results from five assessments across the 14 week course. Our observations were obtained from the three questionnaires from the Motivational Strategies for Learning Questionnaire. This included Assessment and Learning Confidence, along with Satisfaction Scores. These were taken at weeks three, eight and twelve.

#### 4. PROCEDURE

The programming course was delivered through two hours of lecture and three hours in the Computer Laboratory per week. Students enrolled in the course had been allocated to two groups which enabled them to attend Computer Laboratory classes which had a maximum of 20 computers.

In the first trimester of 2013, a total of 38 students were enrolled in the programming course. The first 20 students were allocated to Group A while the remaining students were placed in Group B. The researchers decided that Group A was going to be the pair programming group while Group B was the individual programming group. However, in 2014, the participants decided their own partners in the Pair Programming Group. Therefore, we call the group 2013 as Compulsory Group and the group 2014 the Voluntary Group.

For the statistical results of our research findings, we used Standard Error, which is the standard deviation of the sampling distribution of a statistic.

The MSLQ surveys measured the participants' motivation and learning strategies at three points (Week 3, 8, and 12) during the trimester and the results are presented in the graphs.

#### 5. RESEARCH FINDINGS

What was of interest to the Researchers was the behaviour between the 2014 Voluntary Pair Programming Group and the

2013 Compulsory Pair Programming Group. We found that an improvement in the 2014 Voluntary Pair Group's Confidence in Assessment and Learning exceeded the levels of the Pair Group in 2013, which was a Compulsory Group. This was attributed to the voluntary Pair Group choosing their own preferable partners and this made the pair run through the course more smoothly with less stress than the Compulsory Pair Group in 2013. By doing this, they had less pressure from each other to complete the assessments than the Compulsory Group in 2013. We also found that pair pressure took a negative role here as better and experienced members felt that they were "carrying" the novice student. As a result many became impatient with their fellow paired members and tension levels rose.

#### 6. CONCLUSION

The researchers conceded it was too early to confirm the effectiveness of Pair Programming for this group of students.

The outcome of this research shows the exact opposite of our hypothesis as mentioned at the beginning of this paper.

We found that there are lots of variables that may have influenced the outcome of this study and if we misuse them, the pair programming work can't be better performing than individual work. Therefore we are pointing out here what prevented Pair Programming from out-performing individual work and recommend better ways of running Pair Programming.

#### 7. REFERENCES

- Clark, A., Ernst, J., & Scales, A. (2009). Results of a study using the Motivation Strategies for Learning Questionnaire (MSLQ) in an introductory engineering graphics course. In *annual meeting of the American Society for Engineering Education Southeast Section Conference, Austin, TX*.
- Mendes, E., Al-Fakhri, L. B., & Luxton-Reilly, A. (2005, June). Investigating pair-programming in a 2 nd-year software development and design computer science course. In *ACM SIGCSE Bulletin* (Vol. 37, No. 3, pp. 296-300). ACM.
- Smith, B. J., & Delugach, H. S. (2010, October). Work in progress—Using a visual programming language to bridge the cognitive gap between a novice's mental model and program code. In *Frontiers in Education Conference (FIE), 2010 IEEE* (pp. F3G-1). IEEE.
- Teague, M. M. (2011). Pedagogy of introductory computer programming: a people-first approach. Master Research Paper, Queensland University of Technology
- Williams, L. (2007). Lessons learned from seven years of pair programming at North Carolina State University. *ACM SIGCSE Bulletin*, 39(4), 79-83.