

LOOKING INTO THE FREEDOM OF PARTNER CHOOSING IN PAIR PROGRAMMING

Soo See Chai^{1,*}, Kok Luong Goh²

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

²International 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 other words, students were paired according to their programming compatibility that was 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 a programming problem. However, only 89.2% of the students prefer to work in pairs when solving programming while 5.4% prefer to work as an individual. This initial finding tallies with the other research whereby it shows that pair programming benefits the students in solving a 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, a 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, advice, 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 results 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 other words, students were paired according to their programming compatibility that was 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 assignments in pair programming. The actual experiment method was 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 of programming research were conducted by assigning the students with a partner. Different ways were used in the 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 an Introductory Computer Science course based on similar grades on selected subjects. Some of the researchers used random assignment methods whereby the assignment of the partner was not based on any criteria. In the research carried out by Braught, Wahls, and Eby [2], pairing for the first few labs was 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 knowledge and ability, personality traits were also used as a guideline in pairing the students in pair programming.