

On Pair Rotation in the Computer Science Course

Hema Srikanth, Laurie Williams, Eric Wiebe, Carol Miller, Suzanne Balik
North Carolina State University
{hlsrikan, lawilli3, wiebe, miller, spbalik}@unity.ncsu.edu

Abstract

In a course environment, pairing a student with one partner for the entire semester is beneficial, but may not be optimal. The authors conducted a study in two undergraduate level courses to observe the advantages and disadvantages of pair rotation whereby a student pairs with several different students throughout the semester. This paper summarizes teaching staff and student perceptions on the viability of pair rotation. Teachers find pair rotation valuable because the teaching staff can obtain multiple peer evaluations on each student and because dysfunctional pairs are regularly disbanded. However, pair rotation adds to the burden of assigning pairs multiple times per semester. The majority of students in the study perceived pair rotation to be a desirable approach. Additionally, most students considered peer evaluation to be an effective means of providing feedback to teaching staff. However, they did not significantly believe that peer evaluation was an effective means for motivating students.

1. Introduction

Pair programming [7] is a software development practice in which two programmers work together at one computer, collaborating on the same algorithm, code, or test. One of the pair is the *driver* who actively types at the computer; the other plays the role of *navigator*, watching the work of the driver and identifying defects. The two are also continuous brainstorming partners. In industry, pair programming has been sporadically used for decades [7]. Only in recent years, educators have begun encouraging the use pair programming practice in the classroom. Prior to this time, such collaboration was generally considered cheating; educators were concerned about allowing pair programming because the approach had the potential for only one of the pair to learn while the other would passively allow their partner to do the work. However, research has shown that pair programming in the computer science classroom is generally valuable [7].

In a course environment, pairing a student with one partner for the entire semester is beneficial for both students and educators, but may not be optimal. This paper explores the viability of pair rotation. From an educational perspective, the term *pair rotation* [7] is used to denote when students pair with different classmates throughout the semester. With each partner, students assume both the roles of driver and navigator for varying periods of time in each programming session. The authors of this paper applied pair rotation in two undergraduate computer science courses. At the end of the semester, the students shared their perception via an administered survey. Additionally, the teachers, who are also the authors of this paper, shared their observations via structured meetings. This paper summarizes the results of the survey and the structured meetings.

2. Related Work

Researchers at several universities have applied pair programming in computer science courses and found this practice to be beneficial to students. The advantages of,

disadvantages of, and concerns with pair programming in an educational context as noted by these researchers are discussed below.

2.1. Advantages of pair programming:

- *Pair programming improves quality, teamwork, and communication.* Researchers observed that pair programming conducted with senior-level undergraduates led to a 15% reduction in total number of observed defects with a 15% increase in total development time [8, 11]. Additionally, the researchers found that the students working in pairs were more satisfied, solved tougher problems faster, and had improved team communication and overall team effectiveness [8, 11].
- *Pair programming improves retention and confidence.* Researchers demonstrated that pair programming students were significantly more likely to complete the course and achieve a grade of C or higher [3, 10]. The pairing students demonstrated higher confidence in their project work in comparison to students who did not pair program [3, 10]. Additionally, the students who paired were more likely to take the follow-on programming course than those who did not pair program [3, 9].
- *Pair programming leads to improved comprehension and learning.* In a study conducted at two different universities, the students reported that pair programming allowed them to understand the project better and improved their learning and comprehension of unfamiliar topics [2, 5].

2.2. Disadvantages of pair programming:

- *Pair programming could lead to schedule issues, pair incompatibility, and unequal participation.* Some students in a study opposed the pair programming practice because of the following disadvantages: pair incompatibility, schedule-related issues between pairs, and unequal participation by the individuals in a pair [4, 5].

2.3. Concerns with pair programming:

- *Pair incompatibility could be a cause for concern.* Several studies have shown that matching pairs based on skill level is beneficial for productivity [4, 5]. Additionally, students with a higher self-reported skill level report the least satisfaction when they pair program with students of lesser self-reported skill level [6]. Researchers found that students produce their best work when they are paired with a partner of equal skill level [6].

3. Research Methodology

Pair rotation was used in the freshman (CS1) Introduction to Programming – Java course and in the advanced undergraduate software engineering (SE) course at North Carolina State University (NCSU) during the Fall 2002 (SE) and Spring 2003 (CS1) semesters. These courses are required for the successful completion of undergraduate curriculum in the department of Computer Science at NCSU. Students in both classes were given a brief overview of the pair programming practice during the first week of the semester and completed all their programming projects in pairs.

Four course sections of CS1 with 270 participants were involved in the study. The students were assigned four partners throughout the semester. One course offering of SE (140 students) was involved in the study. These students worked in pairs for five assignments throughout the semester and were assigned a new partner for each assignment. Additionally, in the last third of the semester they completed a five-week team project with three or four other teammates. Due to the timing of this study, the SE students were emailed the same pair programming survey as the CS1 pairing students after the completion of the semester. Because the survey was conducted after the semester was over, only 17 of the students returned the survey. Because only 17 of the SE students completed the survey, our SE findings may be biased. However, the course grade profile of these 17 students was roughly equivalent to the class as a whole.

At the end of the semester, the authors (three of them were instructors for the courses) met to gather qualitative information about the use of pair rotation in their classrooms. Additionally, a survey was administered to students in which they were asked to share their opinion on the pair rotation practice.

4. Results

In this section, we summarize the results of our student survey and our qualitative research. Our goal was to examine the student perception on the pair rotation practice and the advantages and disadvantages of pair rotation and the associated peer evaluation.

4.1. Student Perception of Pair Rotation

The students were surveyed on their perception on pair rotation. Two groups were involved in the study:

Group 1: 270 students from CS1 who employed pair programming.

Group 2: 17 of the SE students who employed pair programming and returned the survey.

In analyzing the survey data, we explored whether students who changed partners after each project/assignment found pair rotation to be advantageous by asking the following yes/no question: “*Do you think it was a good idea to change partners after each assignment?*” The chi-square test was applied at 95% confidence level to identify if pair rotation was useful. Our null and alternate hypotheses are as follows:

- H_0 : *The ratio of the number of surveyed students who preferred pair rotation to the number of the students who did not prefer pair rotation is 2:1.*
- H_1 : *The ratio of the number of students who preferred pair rotation to the number of students who did not prefer pair rotation is greater than 2:1.*

For Group 1, 197 students [73%] acknowledged that pair rotation was a good idea. We accept the alternate hypothesis ($\chi^2(1) = 7.207, p < 0.010$). For Group 2, 16 [94%] agreed that pair rotation was a good idea. We accept the alternate hypothesis ($\chi^2(1) = 5.989, p < 0.025$). Based on both these findings, we conclude that the ratio of students who preferred pair rotation to those who did not is significantly greater than 2:1.

4.2. Advantages and Disadvantages of Pair Rotation to Students

Students were asked to respond to an open-ended question where they were asked to qualitatively discuss the advantages and disadvantages of pair rotation. We grouped similar advantages and disadvantages noted by students and report only the advantages and disadvantages that were independently noted by over 25% of the total students.

Advantages:

- *Exposure to more classmates.* Pair rotation enabled the students to work with different classmates and learn new ways to solve problems. This advantage was noted by 183 [67.8%] of Group 1 and 14 [82.4%] of Group 2.
- *Desire for new partner.* Pair rotation permitted students to switch partners for future projects if they were not compatible with his/her current partner. This advantage was noted by 80 [29.6%] of Group 1 and 5 [29.4%] of Group 2.

Disadvantages:

- *Need to re-adjust.* Pair rotation could be inefficient, as it requires the students to re-adjust to a different partner and his/her programming style at the beginning of every project. This disadvantage was noted by 111 [41.11%] of Group 1 and 11 [64.7%] of Group 2.
- *Compatible with partner.* If a student's current partner is highly compatible, pair rotation may lead to the loss of a perfectly good partner, and introduces the risk of not having a good partner for the next project. This disadvantage was noted by 116 [42.9%] of Group 1 and 7 [41.1%] of Group 2.

4.3. Advantages and Disadvantages of Peer Evaluation to Teaching Staff

When students work in pairs, the potential exists for one partner to do most or all of the work while the credit is assigned equally to both partners in the pair. As a result, students were required to complete a short peer evaluation on the performance and the contributions of their partner. Via a web-based peer evaluation tool, students assigned 0 to 20 points on each of the following five questions, which gave the partner a score of 0%-100% on their peer evaluation:

1. Did the student read the lab assignment and preparatory materials before coming to the scheduled lab?
2. Did the student do their fair share of work?
3. Did the student cooperatively follow the pair-programming model (rotating roles of driver and navigator)?
4. Did the student make contributions to the completion of the lab assignment?
5. Did the student cooperate?

Students in all classes were aware that their grades could be significantly affected by their peer evaluation. However, peer evaluations were handled differently in the CS1 class and in the SE class. In the CS1 class, the instructors examined the peer evaluations at the completion of the semester. In SE, a student was approached by the instructor as soon as he or she had two peer evaluation scores of 80% or less. Depending upon the severity of the situation, the student was either given a warning or his or her grade on the assignment was dropped. The grade was dropped to no lower than the earned grade multiplied by the peer evaluation score. The partner of the violating student's score was then increased to (the earned score * (1 + (1-partner's peer evaluation score))) but not exceeding a value of 100.

In this study, we queried the students on their perception of the effectiveness of peer evaluation in providing feedback to teachers and in motivating students to improve contribution. The majority of the students in both CS1 and SE agreed that peer evaluation was an effective means of feedback to the teacher as shown in Table 1. However, a large number of students from CS1 believed that peer evaluation was not a good source of motivating students

to contribute. Possibly the majority of students from CS1 disagreed to peer evaluation being a motivating factor because they did not receive interim feedback on how peer evaluation impacted the grades of the inactive students during the semester. Several factors have likely affected these results: (1) good students do not need peer evaluation to be motivated to succeed and (2) only violators (and their partners) were aware that the instructors acted upon the peer evaluation scores.

	Yes	No	Unsure	Number of Students
PE effective for providing feedback to instructors for CS1 group	170 [63%]	77 [29%]	23 [8%]	270
PE effective for providing feedback to instructors for SE group	11 [65%]	5 [29%]	1 [6%]	17
PE motivates the students to contribute for CS1 group	99 [37%]	142 [53%]	29 [10%]	270
PE motivates the students to contribute for SE group	11 [65%]	3 [18%]	3 [18%]	17

Table 1: Peer Evaluation Results

Based on the reflection of the instructors/authors, the following are the advantages and disadvantages of peer evaluation to the teaching staff relative to having static partners throughout the semester.

Advantages

- *Multiple forms of feedback.* When a student is paired with only one student for the entire semester, it can be difficult to handle the situation in which one student is saying he or she did all the work while the other partner insists he or she had contributed. To overcome this situation, instructors can apply pair rotation to get the input of several partners for each student and confidently deal with an ineffective student.
- *Handling dysfunctional pairs.* Dysfunctional pairs are broken up via normal course administration. Without pair rotation (when a student works with only one partner in a semester), the teacher is presented with the problem of needing to break up another pair in order to separate and reassign the students in dysfunctional pairs.

Disadvantages

- *Reassigning of pairs.* Pair rotation requires teaching staff to assign/create pairs each cycle. At NCSU, we have developed a web-based partner-assigning program to aid in this process. Students are queried on the names of classmates they wish *not* to be paired with, and the students are not paired with these classmates. The tool is being further modified to consider factors, which may increase pair compatibility, such as personality type, skill level, and self-esteem.
- *Need for peer evaluation.* Pair rotation requires the teaching staff to consider the peer evaluation factor and its resulting issues. When all students work alone, the grade determination process is based on student contribution and is less complicated.

5. Conclusions

We have found that pair rotation is beneficial to both teachers and students. Students have the opportunity to learn from several partners and change partners in situations where the current pairing is ineffective. Teachers benefit from pair rotation by effectively dealing with dysfunctional pairs and inactive students via peer evaluation.

Although the study shows that pair rotation is preferred, the practice has some drawbacks such as the adjustment period encountered by a student after every partner change. We believe that this drawback can be overcome by identifying a balance between a student having too many partners and having just one partner throughout the semester. Further study in identifying an optimal number of pairs a student can have during a semester might be worthwhile. There is also a need to use peer evaluation in pair programming to improve the productivity of all students, both strong and weak.

6. Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. DUE CCLI 0088178. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. We would like to thank the NCSU Software Engineering Reading Group for their careful review and helpful suggestions.

7. References

- [1] K. Beck, *Extreme Programming Explained: Embrace Change*. Reading, Massachusetts: Addison-Wesley, 2000.
- [2] D. Johnson and J. Caristi, "Extreme Programming and the Software Design Course," in *Extreme Programming Examined*, M. Marchesi, G. Succi, D. Wells, and L. Williams, Eds. Boston, MA: Addison Wesley, 2002, 2002, pp. 261-271.
- [3] C. McDowell, L. Werner, H. Bullock, and J. Fernald, "The Impact of Pair Programming on Student Performance of Computer Science Related Majors," presented at International Conference on Software Engineering 2003, Portland, Oregon, 2003.
- [4] G. Melnik and F. Maurer, "Perceptions of Agile Practices: A Student Survey," presented at XP/Agile Universe, Chicago, IL, 2002.
- [5] D. Sanders, "Student Perceptions of the Suitability of Extreme and Pair Programming," in *Extreme Programming Examined*, M. Marchesi, G. Succi, D. Wells, and L. Williams, Eds. Boston, MA: Addison Wesley, 2002, pp. 261-271.
- [6] L. Thomas, M. Ratcliffe, and A. Robertson, "Code Warriors and Code-a-Phobes: A study in attitude and pair programming," presented at SIGCSE, Reno, NV, 2003.
- [7] L. Williams and R. Kessler, *Pair Programming Illuminated*. Reading, Massachusetts: Addison Wesley, 2003.
- [8] L. Williams, R. Kessler, W. Cunningham, and R. Jeffries, "Strengthening the Case for Pair-Programming," in *IEEE Software*, vol. 17, 2000, pp. 19-25.
- [9] L. Williams, C. McDowell, N. Nagappan, J. Fernald, and L. Werner, "Building Pair Programming Knowledge Through a Family of Experiments," presented at International Symposium on Empirical Software Engineering (ISESE) 2003, Rome, Italy, 2003.
- [10] L. Williams, E. Wiebe, K. Yang, M. Ferzli, and C. Miller, "In Support of Pair Programming in the Introductory Computer Science Course," *Computer Science Education*, vol. September 2002.
- [11] L. A. Williams, "The Collaborative Software Process PhD Dissertation," in *Department of Computer Science*. Salt Lake City, UT: University of Utah, 2000.