On Understanding Compatibility of Student Pair Programmers

Neha Katira[1], Laurie Williams[1], Eric Wiebe[2], Carol Miller[1], Suzanne Balik[1], Ed Gehringer[1]

[1]Department of Computer Science
[2]Department of Math, Science and Technology Education
North Carolina State University, Raleigh, NC 27695
{nakatira, lawilli3, wiebe, miller, spbalik, efg}@ncsu.edu

ABSTRACT

In recent years, educators have increasingly used pair programming in their computer science courses. Pair programming has been shown to be beneficial for both the teaching staff and the students in the courses. Occasionally, though, students are not compatible with their partners. An extensive study was done at the North Carolina State University to provide guidance on forming student pairs to improve the chances that pairs will be compatible and have a productive work relationship. We examined compatibility among freshman, advanced undergraduate, and graduate students. We have found that the students' perception of their partner's skill level has a significant influence on their compatibility. Graduate students work well with partners of similar actual skill level. Freshmen seem to work better with partners with different Myers Briggs personality type. Students' self-esteem does not appear to be a major contributor to pair compatibility.

Categories and Subject Descriptors

D.2.9 [**Management**]: Programming teams

General Terms

measurement, experimentation, human factors

Keywords

pair programming, compatibility, computer science classroom, collaboration, Myers Briggs

SIGCSE'04, MARCH 3–7*,* 2004, NORFOLK*,* VIRGINIA, USA*.*

1 INTRODUCTION

The authors of this paper have over three years of experience with pair programming in the computer science classroom. Most often, student pairs are compatible and have productive working relationships. We were interested in seeing if we could improve the compatibility of student pair programmers. To understand the factors that contribute to the compatibility of pair programmers in the computer science classroom, we studied over 564 undergraduate and graduate students at North Carolina State University (NCSU) during the Fall 2002 and Spring 2003 semesters.

The courses in which we performed our research are freshman Introduction to Programming – Java (CS1), undergraduate Software Engineering (SE), and graduate object-oriented (OO) programming. The CS1 and SE classes had multiple two- to three-week programming assignments throughout the semester; students were mandated to work in pairs, with different assigned partners for each one of these programs. The OO students had the option to pair or to work solo; pairing students were assigned their partner by the teaching assistant (TA) after indicating their preference on a message board.

After each pairing cycle, all students were required to complete a web-based peer-evaluation survey about the contributions of their partner for the assignment and how compatibly the pair worked together. Our analysis focuses on the compatibility questions answered by the students after each pairing cycle.

The remainder of this paper is organized as follows: Section 2 describes the hypotheses of the study. Section 3 gives the background and related studies. In Section 4, we describe the experiment, and Section 5 provides the results of the experiment. Finally, Section 6 summarizes the findings and the future work

2 HYPOTHESES

The experiment at NCSU focused on examining possible factors that we felt had the potential to have an effect on the compatibility between student pair programmers. We tested four hypotheses, as listed in Table 1. In the table, a check indicates that the hypothesis was examined with the data for that class. Our findings on these hypotheses will be discussed in Section 5.

Table 1: Research Hypotheses

Hypothesis: Pairs are more compatible if students with …		**CS1**	**SE**	**OO**
H-1	… different **personality types** are grouped together.	√	√	
H-2	… similar **actual skill levels** are grouped together.	√	√	√
H-3	… similar **perceived technical competence** are grouped together.	√	√	√
H-4	… similar **self-esteem** are grouped together.	√		

3 BACKGROUND AND RELATED WORK

Much of the research on pair programming in an academic environment has concentrated on establishing the efficacy of the practice for educating students. A family of pair-programming experiments encompassing over 1200 beginning computer science students was conducted at the University of California at Santa Cruz (UCSC) and NCSU [1, 5, 6, 8, 9, 12, 14]. The researchers concluded that the students who programmed in pairs performed at least as well as the students who programmed independently. They were also more likely to continue with further computer science classes and to major in computer science. Additionally, the results of an experiment conducted at NCSU [8] show that pair programming creates a laboratory environment conducive to more advanced, active learning than traditional labs, leading to students being more productive and less frustrated. Further, a study of undergraduate students in a computer architecture course at Pace University in New York found a positive correlation between out-of-class collaboration and student achievement based on student project and examination grades [3].

Based on our three years of experience with pair programming in computer science classes, we conjectured that there was a relationship between pair compatibility and personality type, skill level, and self-esteem.

3.1 Myers Briggs Personality Type

The Myers Briggs Type Indicator [4] (MBTI) is often used in studies to classify people into one of the sixteen personality types based on the dimensions such as introversion, intuition, feeling, and judgment. Research [2] involving a pilot study of 34 introductory programming class students at a medium-size midwestern university and a follow-up study of 114 college students attending a different campus of the university, taking the same computer programming class as the pilot study students, indicates that there is a relationship between programming performances and the four MBTI personality dimensions.

The results revealed that sensing and judging students performed better than intuitive and perceptive students respectively on programming assignments. However, the personality type did not influence student test achievement or overall achievement. The researchers also concluded that personality type does not appear to be an important factor in predicting whether a student will drop a class.

David Keirsey [4] conjectures that different types of people have different sets of strengths and weaknesses. The strengths and weaknesses of different types complement each other. We contended that if the students with different personality types are given an opportunity to work together, they could explore different views resulting from differing opinions and succeed in producing better results.

3.2 Similarity of Skill Level

A survey-based study [7] performed at the University of Calgary and Southern Alberta Institute of Technology (SAIT) in Canada reported that students preferred working with a partner who matched their own qualifications and experiences. If the pairs' skill level and experience were not a match, the stronger partners tended to complain of carrying an unbalanced share of the work. On an open-ended survey, some of the stronger partners also suggested that such pairs are more productive when the stronger of the pairs is the driver. However, this type of pairing relationship is not advisable because the weaker student feels disjoint and unimportant [13].

Theory [11] presented by psychologist Lev Vygotsky can explain students' preference toward working with partners of similar skill level. Vygotsky defined the "Zone of Proximal Development" (ZPD) as the range of ability of a student with and without the assistance of a teacher or a more capable peer. The lower end of the range is the student's ability without assistance; the higher end of the range is the student's ability with assistance. Vygotsky contends that it is of utmost importance for the teacher/peer to remain within the student's ZPD. If the teacher/peer works too far above the student's ZPD, the student will become confused and no intellectual growth will occur. If the teacher/peer works too far below the student's ZPD, the student will not be challenged enough and will stagnate. Accordingly, pair-programming students would learn the most from a partner who operates within their own ZPD. Vygotsky's theory further supports pair programming for students because he professed that learning is a social process that does not happen by itself but instead occurs through interaction with others.

3.3 Similarity of Self-Esteem

A survey-based study involving more than 60 students at the University of Wales [10] examined the correlation between the self-reported self-esteem and the performance of first-year students in an introductory programming course that used pair programming. The results revealed that students with lower self-esteem liked pair programming more than students with higher self-esteem. There was also some evidence that high self-esteem students disliked pair programming the most when paired with a low self-esteem partner. Additionally, students seemed to do their best work when paired with students of similar levels of self-esteem.

4 EXPERIMENT

In the 2002–2003 academic year, a pair compatibility experiment was conducted at NCSU in one graduate and two undergraduate classes to examine the hypotheses listed in Section 2. A pair programming lecture was presented to all students and TAs. Our

research centers around the responses to questions asked during student peer evaluations.

4.1 Peer evaluation

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 schedule lab?
2. Did the student do his/her fair share of the 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 understood that the answers to the above five questions on contribution could impact their grade. For the purposes of our research, the students were also asked two additional questions which were no-penalty questions. The students were asked to evaluate their perception of the partner's skill level and their joint compatibility on an ordinal scale:

1. Assess the technical competency of your partner relative to yourself [better, about the same, weaker].
2. Assess how compatible you and your partner were [very compatible, OK, not compatible].

4.2 Students/Courses

The following subsection describes the range of students and courses that were involved in the study.

4.2.1 CS1: Introduction to Programming – Java

The Spring 2003 class of CS1 was distributed into six class sections. There were a total of 26 lab sections with a maximum of 24 students in each lab section. Of the 26 lab sections, 24 sections participated in the pairing experiment. A total of 387 students were enrolled in these 24 lab sections. The students had two hours of lecture time and one three-hour closed lab each week. During the lab, the students completed a structured assignment. The students had four projects to complete during the semester. The TA assigned the students a new partner at the completion of each project. A student never paired with the same student more than once. We analyzed the compatibility data of these 387 students for four partner cycles. Out of 1,548 possible responses, we obtained 1,003 responses.

4.2.2 SE: Software Engineering

In the Fall 2002 semester, 140 students were enrolled in the SE class. The students were divided into six lab sections with a maximum of 24 students in each lab section. The students had two hours of lecture time and two hours of closed lab each week. The students completed five assignments in pairs. They had a new partner for each of the first four assignments and kept the fourth partner for the fifth assignment. A student was not paired with the same partner for more than one pairing cycle except for the last assignment. We analyzed the compatibility data from these 140 students for four partner cycles. Out of 560 possible responses, we obtained 496 responses.

4.2.3 OO: Object-Oriented Languages and Systems

The graduate class in Object-Oriented Languages and Systems offered in the Fall 2002 had 62 students. This class required the students to work on three class projects. There were no lab sections associated with this class. They had the option to work in pairs or work solo as per their choice; 37 of the students chose to pair for at least one assignment. The students would sign up on a message board for pairing if they chose to pair with the preference of a partner. The class TA then paired the students for the project. These students did not complete the MBTI test, therefore we could not analyze hypothesis H-1 for the graduate students. We analyzed the compatibility data of these 37 students for three partner cycles. Out of 111 possible responses, we obtained 64 responses.

5 QUANTITATIVE RESULTS

Are compatibility problems common among pair programmers? We analyzed the compatibility of 564 students over three or four pairing cycles providing us with 1,563 data points. However, as shown in **Table 2**, overall, students were compatible with their partners. Less than 10% of the 1,563 data points indicated incompatible pairs.

Table 2: Overall Student Compatibility

Class	N	Very Compat.	OK	Not Compat.
CS1	1003	63% (633)	26% (264)	11% (106)
SE	496	65% (324)	27% (132)	8% (40)
OO	64	72% (46)	19% (12)	9% (6)

However, we still were motivated to determine if any factors could improve pair compatibility. In this subsection, we explain the four hypotheses examined and the results of the study. Since the compatibility data is ordinal [very compatible, OK, not compatible], and the personality type, exam grade differences, perceived skill level, and the self-esteem can be ranked, the Spearman rank-order correlation (rho) was used in the analyses of the hypotheses.

5.1 Personality Type

H-1 Pairs are more compatible if students with different personality types are grouped together.

Table 3 below displays the results of the analysis of the self-reported compatibility based on whether the pair was match/unmatched; a pair was considered matched if they had identical MBTI (all four letters). About 98 CS1 students did not provide their personality type.

Table 3: Difference in Personality Type

Class	N	Very Compat.	OK	Not Compat.
CS1-Matched	509	61% (307)	26% (132)	13% (70)
CS1-Un matched	123	73% (90)	19% (23)	8% (10)
SE – Matched	259	65% (157)	27% (78)	8% (24)
SE – Un matched	237	65% (154)	26% (61)	9% (22)

The results of Spearman rank-order correlation indicate that pairs reported being more compatible if a student's partner had a different personality type ($\rho < .039$) in the CS1 class. However, this same result was not found in the SE class ($\rho < .34$).

5.2 Actual Skill Level

H-2 Pairs are more compatible if students with similar actual skill level are grouped together.

We examined the correlation of self-reported compatibility with the difference in the actual skill level (as measured by their midterm scores) between the pair. The results of the Spearman rank-order correlation suggest that there is no association between the actual skill level and the compatibility of the CS1 ($\rho < .638$) and the SE students ($\rho < .322$). However, there is a strong positive correlation between the actual skill level and the compatibility of the graduate OO students ($\rho < .037$).

5.3 Perceived Technical Competence

H-3 Pairs are more compatible if students with similar perceived skill level are grouped together.

As discussed in Section 4.1, students were asked to provide their perception of their partner's technical competence with regard to their own competence [better, about the same, weaker]. We analyzed the relationship between student's perception of their partner's technical competence and compatibility. The results are shown in Table 4. About 152 students in the first cycle, 81 students in the second cycle, 41 students in the third cycle, and 57 students in the fourth cycle did not respond to the question that evaluated their perceived technical competence of the partner.

Table 4: Perceived Skill Level

Class	N	Very Compat.	OK	Not Compat.
CS1— Better	253	55% (140)	25% (62)	20% (51)
CS1— Same	260	68% (176)	25% (66)	7% (18)
CS1— Weaker	159	34% (54)	50% (80)	16% (25)
SE – Better	138	66% (91)	31% (43)	3% (4)
SE – Same	300	71% (213)	27% (80)	2% (7)
SE – Weaker	58	28% (16)	48% (28)	24% (14)
OO – Better	15	93% (14)	7% (1)	0% (0)
OO – Same	34	82% (28)	18% (6)	0% (0)
OO – Weaker	15	27% (4)	46% (7)	27% (4)

The results of the Spearman rank-order correlation suggest that there exists a significant positive relationship between compatibility and the perceived skill level of the partner in all three computer science courses ($\rho < .001$) for all the three courses.

5.4 Self-Esteem

H-4 Pairs are more compatible if students with similar self-esteem are grouped together.

At the end of the first pairing cycle, one extra question was added to the CS1 peer evaluation. This question was motivated by the self-esteem study in Wales [10] discussed in Section 3.3. The students were asked to place themselves on a 1 to 9 scale with the following endpoints:

1 = I don't like programming, and I think I am not good at it. I can write simple programs, but have trouble writing new programs for solving new problems.

9 = I have no problems at all completing programming tasks to date, in fact they weren't challenging enough. I love to program and anticipate no difficulty with this course.

This Spearman rank-order correlation indicates that there is a weak negative association between the self-esteem and the compatibility of student pair programmers in the computer science classroom ($\rho < .882$).

6 CONCLUSIONS AND FUTURE WORK

We found that 90% of pairs report that they and their partners work compatibly. In many ways, this suggests that pairs will be highly compatible and successful if we pair them randomly, without necessarily considering personality type, skill level, or self-esteem. In our research on how to improve this pair compatibility, we examined four hypotheses. Table 5 summarizes the results of this research.

Table 5: Results Summary

	Hypothesis Pair are more compatible if students with …	**CS1**	**SE**	**OO**
H-1	… different **personality type** are grouped together	Yes	No	
H-2	… similar **actual skill level** are grouped together	No	No	Yes
H-3	… similar **perceived technical competence** are grouped together	Yes	Yes	Yes
H-4	… similar **self-esteem** are grouped together	No		

The most significant result is that students prefer to pair with someone they perceive to be of similar technical competence. However, educators cannot predict this perception nor can pairs be formed based on this fact. The data suggests that pair compatibility in beginning courses may increase if pairs are formed by joining students of dissimilar personality type. Graduate student pairings could be formed by grouping together students of similar actual skill level, such as midterm scores.

We plan to continue this experiment in the computer science classroom by examining additional factors such as gender and ethnicity of the students.

7 ACKNOWLEDGMENTS

The National Science Foundation Grant DUE CCLI 0088178 provided funding for the research in this pair programming experiment. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views if the National Science Foundation. We would like to thank the NCSU Software Engineering Reading Group for their careful review of and helpful suggestions for the paper. In particular, we would like to thank Nachiappan Nagappan and Anne Jackson who meticulously administered the pairing system during the experiment.

8 REFERENCES

[1] J. Bevan, L. Werner, and C. McDowell, "Guidelines for the User of Pair Programming in a Freshman Programming Class," Conference on Software Engineering Education and Training, Kentucky, 2002.

[2] C. Bishop-Clark and D. D. Wheeler, "The Myers-Briggs personality type and its relationship to computer programming," J*ournal of Research on Computing in Education*, vol. 26, pp. 358-370, Spring 1994.

[3] A. Joseph and M. Payne, "Group Dynamics and Collaborative Group Performance," presented at Thirty-fourth SIGCSE Technical Symposium on Computer Science Education, Reno, NV, March 2003.

[4] D. Keirsey, *Please Understand Me II*. Del Mar, CA: Prometheus Nemesis Book Company, 1998.

[5] C. McDowell, L. Werner, H. Bullock, and J. Fernald, "The Effect of Pair Programming on Performance in an Introductory Programming Course," ACM Special Interest Group of Computer Science Educators, Kentucky, 2002.

[6] C. McDowell, L. Werner, H. Bullock, and J. Fernald, "The Impact of Pair Programming on Student Performance of Computer Science Related Majors," International Conference on Software Engineering 2003, Portland, OR, 2003.

[7] G. Melnik and F. Maurer, "Perceptions of Agile Practices: A Student Survey," Extreme Programming/Agile Universe, Chicago, IL, 2002.

[8] N. Nagappan, L. Williams, M. Ferzli, K. Yang, E. Wiebe, C. Miller, and S. Balik, "Improving the CS1 Experience with Pair Programming," SIGCSE 2003, 2003.

[9] N. Nagappan, L. Williams, E. Wiebe, C. Miller, S. Balik, M. Ferzli, and J. Petlick, "Pair Learning: With an Eye Toward Future Success," Extreme Programming/Agile Universe, New Orleans, 2003.

[10] 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.

[11] L. S. Vygotsky, "Mind in Society: The Development of Higher Psychological Processes," M. Cole, V. John-Steiner, S. Scribner, and E. Souberman, Eds. Cambridge, MA: Harvard University Press, 1978.

[12] 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.

[13] L. Williams and R. Kessler, *Pair Programming Illuminated*. Reading, Massachusetts: Addison Wesley, 2003.

[14] L. Williams, C. McDowell, N. Nagappan, J. Fernald, and L. Werner, "Building Pair Programming Knowledge Through a Family of Experiments," International Symposium on Empirical Software Engineering (ISESE) 2003, Rome, 2003.