

Examining the Compatibility of Student Pair Programmers

Laurie Williams¹, Lucas Layman¹, Jason Osborne², Neha Katira³

¹*Department of Computer Science, ²Department of Statistics*

North Carolina State University, Raleigh, NC 27695

[lawilli3, lmlayma2, jaosborn]@ncsu.edu

³*Micromass Communications*

neha.katira@micromass.com

Abstract

Pair programming has been shown to be beneficial for both students and teaching staff in university courses. A two-phased study of 1350 students was conducted at North Carolina State University from 2002-2005 to determine if teaching staff can proactively form compatible pairs based upon any of the following factors: personality type, learning style, skill level, programming self esteem, work ethic, or time management preference. We examined compatibility among freshmen, advanced undergraduate and graduate student pair programmers. We have found that overall 93% of students are compatible with their partners. Students notably preferred to pair with a partner that he or she perceived to be of similar or higher skill level to them, which can be predicted by grouping students with similar grade point average. Additionally, pairs comprised of a sensor and an intuitor learning style seem to be compatible, and pairs with differing work ethic are generally not compatible.

1. Introduction

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 and North Carolina State University (NCSU) [4, 15, 16, 18, 19, 28, 29]. 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 further with computer science classes and to major in computer science.

Additionally, studies conducted at NCSU [2, 3, 13, 18] have shown that pair programming creates an

environment conducive to more advanced, active learning and collaboration, leading to students being less frustrated, more confident, and more interested in information technology. Further, a study of undergraduate students at Pace University in New York found a positive correlation between out-of-class collaboration and student achievement based on project and examination grades [6]. Finally, pair programming may provide a means for educators to accommodate the preference of Millennial students (those born after 1982) to work in teams and to interact peer-to-peer [20, 21].

Most often, student pairs are compatible and have productive working relationships. However, we were interested in whether we could improve the likelihood of forming compatible pairs and to avoid incompatible pairs. To understand the factors that contribute to the compatibility of pair programmers in the computer science classroom, we studied 1350 undergraduate and graduate students at NCSU. The study was conducted between the fall 2002 and the fall 2005 semesters in a two-phased study.

The pair programming-based courses in which we performed our research are freshmen Introduction to Programming – Java (CS1), undergraduate (junior/senior) Software Engineering (SE), and graduate Object-Oriented Languages and Systems (OO). Phase 1 of the study ran from Fall 2002-Fall 2003 and involved the CS1, SE, and OO classes [7-9] did not indicate that teaching staff could do anything proactively to improve the compatibility of student pairs. Phase 2 involved only the Fall 2004-Fall 2005 SE classes. Phase 2 built upon Phase 1 results by adding more hypotheses and augmenting the research design to further examine possible proactive actions by teachers.

In the courses involved in the study, all students were required to complete a web-based peer evaluation survey about the contributions and compatibility of their partner after each paired assignment was

completed (each pairing cycle). The experiment focused on examining possible factors that we felt had the potential to affect the compatibility between student pair programmers. We tested seven hypotheses, as listed in Table 1. In the table, a check indicates that the hypothesis was tested with the data for that class.

Table 1: Research hypotheses

Hypothesis: Pairs are more compatible if students with [...] are grouped together.		CS1	SE	OO
H-1	... different personality types ...	✓	✓	
H-2	... different learning styles ...		✓	
H-3	... similar perceived skill levels ...	✓	✓	✓
H-4	... similar actual skill level ...	✓	✓	✓
H-5	... similar programming self-esteem ...	✓	✓	
H-6	... similar work ethic ...		✓	
H-7	... similar time management skills ...		✓	

The remainder of this paper is organized as follows: Section 2 describes the study conducted at NCSU. In Section 3, we analyze each of the seven hypotheses in Table 1. Finally, Section 4 summarizes the findings and discusses recommendations for educators.

2. Study details

Between 2002 and 2005, a pair compatibility study was conducted at NCSU in one graduate and two undergraduate classes to examine the hypotheses listed in Table 1.

2.1. Student and course description

The following subsection describes the range of students and courses that were involved in the study. The enrollment statistics for the classes in this study are summarized in Table 2.

Table 2: Student enrollment

Class	Semester	Total	Women	Minority
CS1	Spr. 2003	234	112	116
	Fall 2003	570		
SE-P1	Fall 2002	143	42	22
	Fall 2003	106		
SE-P2	Fall 2004	70	5	3
	Spr. 2005	63	-	-
	Fall 2005	52	3	4
OO	Fall 2002	62	-	-
	Fall 2003	50	24	-
Total		1350	186	145

For the purposes of this study, we consider minorities to be African American, Hispanic, and Alaskan/American Indian. Minority females are counted in both categories. Counts of women and minority students exclude students who did not sign an Informed Consent form.

2.2. Peer evaluation and pre- and post-survey

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. To help identify non-participatory partners, all 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 called PairEval¹.

In Phase 1 of the study, students assigned 0 to 20 points to each of five questions Q1-Q5 in Appendix A, which gave the partner a score of 0%–100% on their peer evaluation. In Phase 2, the students began using a rating system developed by Kaufman et al. [10] in which they were asked five “thought jogger” questions followed by an overall rating. The questions of the rating system can be found in Appendix B.

Students understood that the answers to the peer evaluation questions could seriously impact their grade and that of their partner. The instructor had a meeting with students who received poor peer evaluations. For example, in the SE-P1 and SE-P2 classes, if Student A said that Student B did not do what he/she should have done (e.g. a rating of 70 or below in Phase 1 or “marginal” or below in Phase 2), the instructor would have a conversation with both students separately. If Student B admitted he/she only did half of what he/she should have done, he/she would receive the project

¹ <http://agile.csc.ncsu.edu/paireval>

grade * 0.5 and the points taken away from Student B could be added to Student A's grade up to a score of 100. Usually, warnings were given on the first assignment with no penalty.

For the purposes of our research, the students were also asked two additional questions when they evaluate their partners (Q6, Q7 in Appendix A). The students were asked to evaluate their perception of the partner's skill level and their joint compatibility on an ordinal scale: No penalty was attached to these questions.

At the beginning of each semester in Phase 2 students responded to survey questions asked via the PairEval tool. The students responded to 19 questions in total, though only seven of them are relevant to this study (Appendix C). Additionally at the start of the semester, students completed an online Myers-Briggs Type Indicator (MBTI) test² and an online Felder-Silverman learning style (LS) test³ and reported their results via the PairEval tool.

2.2.1. Introduction to Programming – Java (CS1).

The Spring and Fall 2003 classes of CS1 were included in this study. The students had two hours of lecture time and one three-hour closed lab each week. There were 26 lab sections with a maximum of 24 students in each lab section. During the lab time, the students completed a structured pairing assignment. The students had to complete four assignments during the semester. The TA assigned the students a new partner at the completion of each assignment. The compatibility data of these 804 students for four partner cycles are analyzed in this research.

2.2.2. Software Engineering (SE). There were two phases to the study with SE students. Phase 1 was Fall 2002 and Fall 2003; Phase 2 was Fall 2004, Spring 2005, and Fall 2005. The first author was the instructor for all but the Spring 2005 courses.

Each semester, the class had two hours of lecture time and two hours of closed lab weekly. The students had multiple two- to three-week programming assignments. Students were mandated to work in pairs, with different assigned partners for the assignments. In PairEval, students could list other students with whom they preferred not to work and were not assigned to work with any of these students. Each class ended with a three- to five-person, six-week team project.

2.2.3. Object-Oriented Languages and Systems (OO). The Fall 2002 and Fall 2003 offerings of the Object-Oriented Languages and Systems graduate class were included in the study. The class required students to work on three class projects. There were no lab sections associated with the class. Students had the option to work in pairs or work solo; 37 (60%) students in Fall 2002 and 45 (90%) students in Fall 2003 chose to pair for at least one assignment; some students worked in groups of three. The students signed up on a message board for pairing. The class TA then paired the students for the project. There was no moderated pair programming in the OO class. The compatibility data of these 82 graduate OO students was analyzed over three pairing cycles. Ethnicity and gender information was available only for the fall 2003 class due to the lack of Informed Consent in Fall 2002.

2.3. Missing data

Phase 1: Many students did not fully complete the optional compatibility questions on the peer evaluation or did not sign an Informed Consent form giving us full access to their data. When the compatibility response was not completed, the whole record was deleted when examining the experiment's hypotheses. If the compatibility response from one partner was missing, the compatibility response from both students was discarded. Values for missing variables (other than the compatibility rating) were imputed using SAS Enterprise Miner.

Phase 2: If a partner did not respond to the MBTI or LS questionnaires with legitimate values, that user's MBTI, LS, and survey data were excluded from analysis. Each semester less than five students dropped or audited the courses. These students were also excluded from analysis. Finally, there were many cases where one student evaluated the same student twice on the same assignment. In this case, only the second evaluation was considered. The number of data points for Phase 2 varied depending on the hypothesis under test. These numbers varied based on data cleansing as well as consent provided for access to academic information.

3. Quantitative results

We analyzed the compatibility of 1350 students over three or four pairing cycles. As shown in Table 3, overall, 93% of students were compatible with their partners. However, we still were motivated to determine if any factors could improve pair compatibility, including moving more from the "OK" compatible rating to the "Very" compatible rating. In

² <http://www.humanmetrics.com/cgi-win/JTypes2.asp>

³ <http://www.engr.ncsu.edu/learningstyles/ilsweb.html>

the remainder of this section, we examine the seven hypotheses listed in Section 1.

Table 3: Student compatibility summary

Class	Very compat.	OK	Not comp.
CS1	64%	32%	4%
SE-P1	60%	33%	7%
SE-P2	56%	35%	9%
OO	76%	15%	9%
Total	60%	33%	7%

3.1. Personality types

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

The instrument that we used to assess personality types was the Myers Briggs Type Indicator [11] (MBTI). MBTI has served as a popular means of characterizing personality traits in both the classroom and the workplace. A considerable amount of work has been published on Myers-Briggs personality types (e.g. [14, 23]) among engineering students. The Myers-Briggs scale has four dimensions:

Introvert-Extravert. Introverts are generally introspective and are energized by spending time alone, whereas extraverts thrive in a group setting.

Sensing-Intuition. Sensors prefer information gathered through experience and are attentive to details, while intuitors prefer abstract concepts and are bored by details, preferring innovative thoughts instead.

Thinking-Feeling. Thinkers rely on objective rationalization to make decisions and are considered to be impartial, whereas feelers are more likely to make subjective decisions based on social considerations rather than strict logic.

Judging-Perceiving. Judgers are typically orderly people who prefer rigid structure and planning but may ignore facts that do not fit their plans or structures. Perceivers do little planning and work spontaneously and are more open to facts that do not conform to their views.

Our hypothesis was that different personality types should be grouped together utilizing the “opposites attract” adage. The strengths and weaknesses of different types complement each other. We surmise that if the students with different personality types are given an opportunity to work together, they could explore a variety of views resulting from differing opinions and succeed in producing better results.

In Phase 1, the correlation between the Myers Briggs personality type of the students and their

perceived compatibility was examined for those students who provided their personality types. Pairs were categorized as having the same MBTI or not, and then a Spearman rank-order correlation was conducted with the compatibility response (Not Compatible, OK, Very Compatible). The results, shown in Table 4, for the students who provided their MBTI do not indicate any association between an exact match of the personality types of the students and compatibility.

Table 4: Compatibility and Personality Type – Phase 1

Class	N	p-value	ρ
CS1	421	0.089	0.103
SE-P1	903	0.827	-0.006
OO	30	0.901	-0.021

In Phase 2 of our study, we analyzed the students’ numeric Myers-Briggs scores on each dimension. This allowed us to determine if the strength of the students’ personality traits influenced the compatibility scores. The absolute value of the difference between evaluator and evaluatee of each Myers-Briggs scale was taken. The scale ranged from -100 to 100 on each dimension with discrete qualities (scores of 33, 44, 55, 67, etc. were common).

First, logistic regression was used to determine if a correlation existed between each dimension and a binary compatibility score (Not compatible vs. Very Compatible and Okay.). No correlation was found. Second, we repeated the logistic regression, but this time to see if the dimensions predicted Very Compatible ratings (a response of Very Compatible on the survey). The test showed that the pair’s sensing-intuition scores significantly correlated with whether they were Very Compatible or not. Further *t*-tests showed that students who rated themselves Very Compatible with their partners had a statistically higher *difference* on the sensing-intuition dimension than those students who rated their partners Ok or Not Compatible. The primary difference between sensors and intuitors is that they assimilate information in different ways. Therefore, it may be the case that these pairs are more compatible because they assimilated different sets of information and were able to complement each other’s knowledge.

Therefore, our results show some support for H-1 by indicating that a difference on the sensing-intuition scale can help predict highly compatible pairs.

3.2. Learning styles

H-2 Pairs are more compatible if students with different learning styles are grouped together.

The instrument used to assess learning styles was the Felder-Silverman learning styles assessment [5]. Felder-Silverman learning styles have been used to help students understand their own learning needs and to help professors better tailor their courses to different types of students. The purpose of these learning styles is to help characterize the way in which students absorb and retain information. The Felder-Silverman scale has four dimensions:

Active-Reflective. Active learners learn best by trying things out and working with others, while reflective learners learn more by thinking things out on their own.

Sensing-Intuitive. The sensing-intuitive dimension is intended to be the same as in the Myers-Briggs scale.

Visual-Verbal. Visual learners absorb information best through pictures, graphs, and charts, whereas verbal learners prefer written or spoken explanations.

Sequential-Global. Sequential students learn in orderly, incremental steps with one point or fact connecting to the next, whereas global learners have trouble learning fact-by-fact and learn in cognitive leaps after accumulating all the facts.

In Phase 2 of our study, we examined the students' learning styles scores on each dimension similar to our analysis of their Myers-Briggs personality types. The absolute value of the difference between the evaluator's and evaluatee's scores on each dimension was taken. The scale ranged from -11 to 11 on each dimension with values at -11, -9, -7, -5, etc.

Logistic regression was used to see if the partners' differences in learning styles could predict compatibility or incompatibility. No significant results were found.

As with personality types, analysis showed that differences on the sensing-intuitive dimension were significant in predicting Very Compatible pairs, reaffirming our findings with the personality type data. We note that the sequential-global dimension is nearly significant ($p=0.08$) as well.

3.3. Skill level

Anecdotally, students indicate that they like best to work with students of similar or higher skill level relative to themselves. A survey-based study [17] 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 levels and experiences 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 [27].

In this section we investigate the relationship between actual skill level and compatibility and between perceived skill level and compatibility. Students' actual skill level is measured by midterm scores, GPA, and SAT. As will be discussed, students' perceived skill level is subjectively based upon a question on the peer evaluation system in which each student rates their partner's skill level relative to themselves. The actual skill level was examined to determine if educators could use any objective measures to proactively assign students to a partner or she would perceive to be of similar or higher skill.

Theory [26] 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 occurs through interaction with others.

In the next two sub-sections, we analyze the relationship between actual and perceived skill level relative to compatibility.

3.3.1 Perceived skill level. We examine the following hypothesis:

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

As part of peer evaluation, 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 5.

Table 5: Perceived skill level – Phase 1

Class	N	p-value	ρ
CS1	770	0.019	0.776
SE-P1	924	< 0.001	0.510
OO	104	< 0.001	0.547

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.

Similar results were found in Phase 2 of the study. The frequency distributions of compatibility ratings and perceived technical competence for Phase 2 of the study are shown in Table 6.

Table 6: Frequency of compatibility and perceived technical competence – Phase 2

N=677	Very Compatible	OK	Not Compatible
Better	68.75%	26.14%	5.11%
Equal	64.16%	32.37%	3.47%
Worse	23.87%	52.26%	23.87%

Chi-Square tests were performed to determine if the compatibility rating distributions varied according to the partner's perceived technical competence. The Chi-Square test was significant ($X^2=110.68$, $df=4$, $p<0.001$), indicating that the compatibility rating distributions differed significantly based on perceived technical competency. The results indicate that students were much less likely to rate a partner Very Compatible whom they considered to be of lower competence, thus raising the amount of OK and Not Compatible ratings significantly.

Therefore, our results support H-3. The remaining challenge is to determine if there is any way to predict the student's perceptions of each other's competency based on available objective information.

3.3.2 Actual skill level. We examine the following hypothesis:

H-4 Pairs are more compatible if students with similar actual skill level are grouped together.

In Phase 1, absolute differences in the partners' midterm grades, SAT, GRE, and overall GPA were examined for a correlation with compatibility. SAT/GRE scores were only available for the SE and the Fall 2003 OO class since the other classes had not signed an Informed Consent form. The results are shown in Table 7.

Table 7: Actual skill level – Phase 1

Measure	Class	N	p-value	ρ
Midterm	CS1	802	0.154	-0.066
	SE-P1	951	< 0.001	-0.604
	OO	146	0.375	-0.023
SAT	CS1	821	0.602	-0.006
	SE-P1	795	< 0.001	-0.624
GRE	OO	21	0.246	0.172
GPA	SE-P1	937	0.066	0.020
	OO	21	0.704	0.012

In the Phase 1 SE class, the greater the difference in the midterm or SAT of partners, the less likely it is that they perceive compatibility. In this phase, none of the other measures indicated that these objective measures of actual skill level can be used to predict pair compatibility.

Phase 2 yielded no statistically significant results in predicting pair compatibility based on academic data. However, differences in both computer science GPA and total GPA were predictors of student's perceived competency of their partner. The greater the difference in these scores, the more likely students would perceive their partner as either more or less competent (accordingly).

Therefore, our results do not generally support H-4 and indicate that it is not feasible for an instructor to proactively match students based on available measures of skill (midterm, SAT, GRE, GPA), except for the use of midterm and SAT in the SE class. However, measures of computer science GPA and total GPA may be used to predict perceived competency, which is a predictor of compatibility (as discussed in 3.3.1).

3.4. Programming self-esteem

H-5 Pairs are more compatible if students with similar programming self-esteem are grouped together.

A survey-based study involving more than 60 students at the University of Wales [24] 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. These findings motivated the analysis of H-5.

Our means of gathering self esteem data changed from Phase 1 to Phase 2 of our study. In the first phase, one extra question was added to the pre-semester survey. This question was motivated by the self-esteem study in Wales [24]. 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.

Our findings, as shown in Table 8, indicate a strong correlation in SE, but not in the other classes.

Table 8: Programming self-esteem – Phase 1

Class	N	p-value	ρ
CS1	161	0.117	0.140
SE	411	< 0.001	0.501
OO	54	0.913	-0.122

In Phase 2, the programming self-esteem question was broken up into several theme-related questions (Q1-Q5 in Appendix C). Logistic regression showed that Q2 was significant with regard to pair incompatibility. This question regards students' confidence in their abilities to tackle new problems. Student pairs who were incompatible had significantly higher differences than pairs that were compatible. We hypothesize that this may be due to a difference in "starting points" when working on a new problem. For example, some students may have an advanced concept of the task at hand and can easily relate it to problems experienced in the past. Some other students may not be familiar with the problem at all and have no idea where to begin. This expert-novice problem solving dichotomy has been observed in previous studies of programmers [1, 22, 25]. A large difference between a pair of students in this regard may require one student to spend time educating the other student about the problem and potential solutions, rather than beginning to work on the problem together immediately.

Therefore, our results partially support H-5 and indicate that students' confidence in their problem-solving abilities may be an indicator of pair compatibility.

3.5. Work ethic

H-6 Pairs are more compatible if students with similar worth ethic are grouped together.

Students indicate that they prefer to work with another student who has similar ambitions for success in the course to their own. During Phase 2, in the pre-semester survey (Appendix C, Q6), we asked students rate themselves on a scale from 1 to 9 on the following question:

In your classes, do you work hard enough to:

1: Just barely get by

9: Get the best grade you possibly can.

It is worth noting that over 90% over the students responded with a 5 or higher to this question. We examined the correlation between the results to this question and compatibility for 567 students. Logistic regression revealed that the difference in work ethic was a statistically significant predictor of compatibility. Pairs who were incompatible had a greater difference in their self-reported work ethics than pairs who were compatible. There is also a clear directionality to this data. The mean difference of work ethic in non-compatible pair ratings was 0.64. Thus, the incompatible pair ratings are marked by students with a higher work ethic rating themselves as incompatible with lower work ethic partners.

Therefore, our results support H-6, indicating that self-reported work ethic may be a predictor of pair compatibility.

3.6. Time management preference

H-7 Pairs are more compatible if students with similar time management skills are grouped together.

Students indicate that they prefer to work with a student who has similar time management preferences to their own. During Phase 2, in the pre-semester survey (Appendix C, Q7), we asked students rate themselves on a scale from 1 to 9 on the following question:

When you have homework, when do you get started?

1: Very early

9: Very late

Our results yielded no statistical significance for time management being a predictor of pair compatibility. This may indicate that students do not care about when their partner does the work, but only that they contribute. Also, if the students actually practice pair programming, time management may not be an issue. Arranging a time to work together can be a difficulty for students and is an obstacle for pair programming outside the classroom [12]. The students work whenever they can get together, which may not depend on their time managements preferences. "Pair pressure" tends to motivate students to start earlier [27].

Therefore, our results do not support H-7 that time management preference may be a predictor of pair compatibility.

4. Conclusions and future work

We found that 93% 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, self-esteem, work ethic, or time management skills. In our research on how to improve this pair compatibility, we examined seven hypotheses.

Table 9 summarizes the results of this research.

Table 9: Results Summary

	Hypothesis: Pairs are more compatible if students with ... are grouped together.	CS1	SE	OO
H-1	... different personality types ...	No	Yes*	No
-2	... different learning styles ...	-	Yes*	-
H-3	... similar actual skill level ...	No	Yes*	No
H-4	... similar perceived technical competence ...	Yes	Yes	Yes
H-5	... similar programming self-esteem ...	No	Yes*	No
H-6	... similar work ethic ...	-	Yes	-
H-7	... similar time management skills ...	-	No	-

*Indicates partial support

This paper has three significant results. First, students prefer to pair with someone they perceive to be of similar technical competence. Predicting perceived technical competence may be done by examining the difference between a pair's computer science GPA and total GPA scores. However, the availability of this data may be limited. Second, pairing sensors and intuitors together yields very compatible pairs, potentially because of their ability to complement each other's knowledge. Finally, pairing students with strongly dissimilar work ethics will more likely yield incompatible pairs.

We plan to continue this experiment in the computer science classroom, and we will examine

additional factors such as gender and ethnicity of the students as factors in pair compatibility.

5. Acknowledgements

The National Science Foundation grants DUE CCLI 0088178 and ITWF 00305917 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 of the National Science Foundation. We would like to thank the NCSU Software Engineering Realsearch group for their careful review of and helpful suggestions for the paper. In particular, we would like to thank Ed Gehringer, Carol Miller, and Suzanne Balik for allowing us to gather compatibility data from their classes.

6. References

- [1] B. Adelson, D. Littman, K. Ehrleich, J. Black, and E. Soloway, "Novice-Expert Differences in Software Design," *Human-Computer Interaction (INTERACT '84)*, 1984 pp. 187-192.
- [2] S. B. Berenson, K. M. Slaten, L. Williams, and C.-w. Ho, "Voices of Women in a Software Engineering Course: Reflections on Collaboration " *ACM Journal on Educational Resources in Computing*, vol. 4, no. 1, March 2004.
- [3] S. B. Berenson, L. Williams, and K. M. Slaten, "Using Pair Programming and Agile Development Methods in a University Software Engineering Course to Develop a Model of Social Interactions," *Crossing Cultures, Changing Lives Conference*, Oxford, UK, 2005 pp. to appear.
- [4] 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 pp. 100-107.
- [5] R. M. Felder and L. K. Silverman, "Learning and Teaching Styles in Engineering Education," *Engineering Education*, vol. 78, no. 7, 1988, pp. 674-681.
- [6] A. Joseph and M. Payne, "Group Dynamics and Collaborative Group Performance," *Thirty-fourth SIGCSE Technical Symposium on Computer Science Education*, Reno, NV, March 2003 pp. 368-371.
- [7] N. Katira, "Understanding the Compatibility of Pair Programmers MS Thesis," *North Carolina State University Computer Science*, Raleigh, NC, 2004.
- [8] N. Katira, L. Williams, and J. Osborne, "Towards Increasing the Compatibility of Student Pair Programmers," *International Conference on Software Engineering (ICSE)* 2005, St. Louis, MO, 2005 pp. 625 - 626.

[9] N. Katira, L. Williams, E. Wiebe, C. Miller, S. Balik, and E. Gehringer, "On Understanding Compatibility of Student Pair Programmers," ACM Technical Symposium on Computer Science Education (SIGCSE), Norfolk, VA, 2004 pp. 7-11.

[10] D. B. Kaufman, R. M. Felder, and H. Fuller, "Peer Ratings in Cooperative Learning Teams," American Society for Engineering Education, Charlotte, NC, 1999 pp.

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

[12] W. Krebs, C.-w. Ho, L. Williams, and L. Layman, "Rational Unified Process Evaluation Framework Version 1.0," North Carolina State University, Raleigh, NC, NCSU CSC TR 2006-1, 2006.

[13] Layman, L. L. Williams, J. Osborne, S. Berenson, K. Slaten, and M. Vouk, "How and Why Collaborative Software Development Impacts the Software Engineering Course," *Frontiers in Education (FIE) 2005*, Indianapolis, Indiana, 2005 pp. T4C9-T4C14.

[14] M. H. McCaulley, "The MBTI and Individual Pathways in Engineering Design," *Journal of Engineering Education*, vol. 80, no., 1990, pp. 537-542.

[15] 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, Covington, KY, 2002 pp. 38-42.

[16] 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, Oregon, 2003 pp.

[17] G. Melnik and F. Maurer, "Perceptions of Agile Practices: A Student Survey," *XP/Agile Universe*, Chicago, IL, 2002 pp. 241-250.

[18] N. Nagappan, L. Williams, M. Ferzli, K. Yang, E. Wiebe, C. Miller, and S. Balik, "Improving the CS1 Experience with Pair Programming," *ACM Special Interest Group Computer Science Education (SIGCSE) 2003*, Reno, 2003 pp. 359 - 362.

[19] 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 pp. 185 - 198.

[20] D. Oblinger, "Boomers, Gen-Xers, and Millennials: Understanding the New Students," *Educause Review*, vol. 38, no. 4, July/August 2003, pp. 37-47.

[21] D. Oblinger and J. Oblinger (Eds.), *Educating the Net Generation*. Boulder, CO: Educause, 2005.

[22] B. Schneiderman, "Measuring Computer Programing Quality and Comprehension," *International*

Journal of Man-Machine Studies, vol. 9, no. 4, 1977, pp. 465-478.

[23] A. Thomas, M. R. Benne, M. J. Marr, E. W. Thomas, and R. M. Hume, "The Evidence Remains Stable: The MBTI Predicts Attraction and Attrition in an Engineering Program," *Journal of Psychological Type*, vol. 55, no., 2000, pp. 35-42.

[24] L. Thomas, M. Ratcliffe, and A. Robertson, "Code Warriors and Code-a-Phobes: A study in attitude and pair programming," *SIGCSE*, Reno, NV, 2003 pp. 363-367.

[25] W. Visser and J.-M. Hoc, "Expert software design strategies," in *Psychology of programming*, J.-M. Hoc, Ed. San Diego: Academic Press, 1990, pp. 235-247.

[26] L. S. Vygotsky (Eds.), *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press, 1978.

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

[28] 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, Italy, 2003 pp. 143-152.

[29] 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. 12, no. 3, 2002, pp. 197-212.

Appendix A: Phase 1 peer evaluation questions

#	Question
Q1	Did the student read the lab assignment and preparatory materials before coming to the scheduled lab?
Q2	Did the student read the lab assignment and preparatory materials before coming to the scheduled lab?
Q3	Did the student do his/her fair share of the work?
Q4	Did the student cooperatively follow the pair-programming model (rotating roles of driver and navigator)?
Q5	Did the student make contributions to the completion of the lab assignment? Did the student cooperate?
Q6	Assess the technical competency of your partner relative to yourself [Better, About the same, Weaker].
Q7	Assess how compatible you and your partner were [Very Compatible, OK, Not Compatible].

Appendix B: Phase 2 Peer Evaluation Questions

There were five “thought jogger” questions which were not counted in grading followed by one overall rating, which did count in grading. The instrument was developed by Kaufman et al [10]. Students entered the responses to these questions via the pairEval web application. The thought jogger questions were as follows:

- Has the student attended your group meetings?
- Has the student notified a teammate if he/she would not be able to attend a meeting or fulfill a responsibility?
- Has the student made a serious effort at assigned work before the group meetings?
- Does the student attempt to make contributions in group meetings when he/she can?
- Does the student cooperate with the group effort?

The student then provides an overall rating of their partner based upon the following rubric:

Excellent	Consistently went above and beyond—tutored teammates, carried more than his/her fair share of the load
Very good	Consistently did what he/she was supposed to do, very well prepared and cooperative
Satisfactory	Usually did what he/she was supposed to do, acceptably prepared and cooperative
Ordinary	Often did what he/she was supposed to do, minimally prepared and cooperative
Marginal	Sometimes failed to show up or complete assignments, rarely prepared
Deficient	Often failed to show up or complete assignments, rarely prepared
Unsatisfactory	Consistently failed to show up or complete assignments, unprepared
Superficial	Practically no participation
No show	No participation at all

Appendix C: Student Survey

#	Question
Q1	When working on a programming assignment, do you think that: 1: It takes me longer to complete programming tasks than my colleagues. 9: I am much quicker at solving programming tasks than others
Q2	When working on a programming problem: 1: I have difficulty knowing where to begin on new programming problems. 9: I have no difficulty in starting the problem.
Q3	When helping others with their computer science work, do you: 1: Feel that you do not know enough to help others 9: I know enough to help others
Q4	After a test: 1: I tend to think that I did poorly 9: I tend to think that I did very well
Q5	When I run into a roadblock in my thinking: 1: My mind goes around and around in circles 9: I know that I will find the answer somewhere
Q6	In your class, do you work hard enough to: 1: Just barely get by, 9: Get the best grade you possibly can.
Q7	When you have homework, when do you get started? 1: Very Early; 9: Very late.