Retaining Women in Computer Science: The Impact of Pair Programming
Project Update ITWF/ITR Workforce PI's Meeting October, 2004
Revised November, 2004

Linda L. Werner (PI), Computer Science Department
Heather Bullock (co-PI), Psychology
Charlie McDowell (co-PI), Computer Science Department
Julian Fernald (researcher),
Brian Hanks (graduate researcher), Computer Science Department
University of California, Santa Cruz

Goals

As the President's Information Technology Advisory Committee Report[1] clearly indicates, women remain a largely untapped resource in computer science fields. In order for high technology computer related fields to meet their ever-growing need for skilled programmers, it is essential that the talents and abilities of traditionally underrepresented populations be fostered. Furthermore, the relative high paying nature of high tech computer jobs may be an important equalizer in terms of the gender wage gap. Unfortunately, it appears that women's perceptions of computing as isolating, lonely, competitive, and male dominated continue to be significant barriers to their participation in the field. In addition to increasing programming efficiency, pair-programming[2] may make careers in these fields more appealing to women. The primary purpose of this study is to assess the effectiveness of pair-programming on the performance and retention of women in computer science and other related fields such as information systems management and computer engineering. Specifically, in our proposal we hypothesized that:

1) Students who program in pairs will produce better programs in terms of functionality and readability, report greater confidence in their solutions, enjoy programming more, and have higher retention rates than students who program independently.

2) Women who program in pairs will produce better programs in terms of functionality and readability, report greater confidence in their solutions, enjoy programming more, and have higher retention rates than women who program independently.

3) Differences between women and men in performance, confidence, enjoyment, and retention will be smaller among paired programmers than among solo programmers.

4) Women's performance and retention will be positively correlated with egalitarian gender role beliefs, and perceptions of computer science as useful, and negatively correlated with perceptions of computer science as a male domain.

Activities to Date

During the 2000-01 academic year we collected data on 555 residential students (413 men and 141 women1) who attempted an introductory computer programming course, intended primarily for computer science, computer engineering, electrical engineering, and information systems management majors, at the University of California-Santa Cruz (UCSC). We will refer to the data from the 2000-01 period as the primary study. We have also collected data for an additional 55 students for the same UCSC course during the Winter 2003 term. We will refer to this later data which was compared against data

[1] Gender information was unavailable for one student.

collected during Spring 2001 as the secondary study. Finally, we have collected data for 214 commuter students at a 2-year community college in Santa Cruz, Cabrillo College, and at a 4-year state university, San Jose State University (SJSU), for the 2003-2004 period. We will refer to the data from the 2003-2004 period as the tertiary study.

During 2000-01, four sections of the course were offered: one in the Fall quarter, two in Winter, and one during the Spring. One of the principal investigators of this study, Charlie McDowell, taught the Fall and Spring sections of the course. The two Winter sections were taught by UCSC faculty members not associated with this project.

Students enrolled in the Fall 2000, Winter 2001, and Winter 2002 sections were required to complete all assignments using pair programming. Students in the Spring 2001 section, on the other hand, were required to complete programming assignments independently. Students in the pairing sections were asked to submit a list of three names of potential partners, and partners were assigned based on these preferences. In nearly all instances, students were assigned a partner from their list. Those that stated no preference were randomly assigned a partner. Whenever possible, students remained with the same partner throughout the quarter, however, due to schedule changes and drops, a small number of partner reassignments were necessary. Seventeen students across the three pairing sections were permitted to program alone for various reasons. For the tertiary study, for the fall 2003 and spring 2004 semesters at Cabrillo College, two instructors participated with one paired and one solo section of beginning programming each. These instructors also allowed the students to choose their own partners. For the fall 2003 semester at SJSU, we had one instructor participating with one paired and one solo section of beginning programming. In the spring this same instructor participated with a paired section and a second instructor participated with a solo section.

Regardless of whether they completed assignments in pairs, all students took exams independently. The final exam assessed students' knowledge of programming concepts and their ability to write new code.

The data collected consisted of:

- all course grades including exam scores and programming assignment scores,
- a log of time spent on each programming assignment,
- a short (3 question questionnaire) at the end of each programming assignment,
- a detailed questionnaire at the start of the class asking for self reporting of computer skills, math and computer classes, and perceptions of the role of women in computer science (among other things) (shorter questionnaire for tertiary study),
- a detailed questionnaire at the end of the class containing a subset of the questions from the questionnaire at the start of the class,
- SAT scores (primary study only),
- the number of computer science related courses taken in the 3 quarters following the introductory programming course and the grades in those courses (primary study only), and
- the students' declared major 1 year after taking the introductory programming course (primary study only).

Most of the above data has been analyzed with regard to 3 of the 4 four hypotheses of the study (see findings). We are still in the process of analyzing the data from the tertiary study. To date a total of 1 journal submission, 7 conference papers, and 1 poster

presentation have been made by the PIs on the work from this project [3,4,5,6,7,8,9,10,11].

The main activities on this project during the past 12 months have been

- dissemination of our results via paper preparation and presentation,
- analysis of the secondary study data,
- preparations for extending our project to collect data from two additional institutions in order to try and replicate our findings with different student populations,
- collection of the tertiary study data, and
- initial analysis of the tertiary study data.

We initiated the secondary study in order to try and strengthen our findings with regard to the quality of the programs produced by pairs versus those produced by students working alone. That data is of particular interest because the students in that class worked in pairs completing the exact same assignments as those used during the non-pairing section of our primary study. This allowed us to more directly compare the quality of the programs produced between the pairs and the solo programmers. We evaluated the last 3 of the 5 programming assignments for the classes.

We initiated the tertiary study to try to replicate our results at two additional institutions with students different than those at UCSC. Whereas UCSC is primarily a residential campus, Cabrillo College and SJSU are commuter campuses. We have set up a web site to collect the programming logs from the students[2].This was necessary because we do not want to burden the instructors at the other institutions with this data collection. We believe that the web site can also serve as a model for other instructors using pair programming in their classrooms. The programming logs have a pedagogical value as well as an experimental value. The logs provide regular checkup on the pair programming process when it is performed outside of a controlled lab, as is the case in these classes.

We enrolled the cooperation of faculty members at SJSU and Cabrillo College. We collected data from eight additional sections in all using both a pair and solo section at each school during both fall and winter semesters. We had approximately 100 participants each semester for a total of 214 additional subjects.

Results to Date

We have completed our analysis of the primary and secondary data with regard to the first three hypotheses. We are still analyzing data from the tertiary study. The specific numbers listed below are for the data from one or both of these studies and are labeled either primary, secondary, or tertiary.

Finding 1: hypothesis 1 is generally supported by our data. Specifically:
The programs produced by the students working in pairs were significantly better than those of the students working alone, 86.6% vs. 68.1% ($p<.001$) (primary study). This result could be because different program problems were assigned to the different sections or because of different instructors. We did not generally find the same result in

[2] The website is https://www.soe.ucsc.edu/programmingstudy/. There are two test logins username:pairTest with password:pairTest123, and username:soloTest with password:soloTest123.

the secondary study. In the secondary study, we controlled for instructor and program assignment. We found that paired students were more likely to turn in working programs and these programs correctly implemented more required features. Results were not consistent regarding program complexity measures.

- For the tertiary study, programs produced by paired students were not significantly different than programs produced by students working alone, 72% vs. 70% ($p>.5$).
- The students working in pairs reported greater confidence, on a 100 point scale, in their solutions, M=89.4 vs. M=71.2 ($p<.001$) (primary study), M=81.6 vs. M=72.7 ($p<.001$) (secondary study), and M=86.6 vs. M=76.0 ($p<.005$) (tertiary study).
- The students working in pairs reported greater enjoyment, on a 7 point scale, of the programming process, M=5.14 vs. M=4.69 ($p<.005$) (primary study) and M=5.86 vs. M=4.80 (corresponding to $p<.001$) (secondary study). For the tertiary study there was no significant difference.
- More students in the pair programming sections stayed in the class and took the final exam, 90.8% vs.80.4% ($p <.001$) (primary study).
- More students in the pair programming sections passed the class, 72.3% vs. 62.8% ($p<.05$) (primary study) and 66.0% vs. 52.3% ($p<.05$) (tertiary study).
- Among the students initially intending a computer science major, and who passed the introductory course and remained at UCSC for at least a year, the pairing students were more likely to have declared a computer science related major 1 year after completing the introductory programming class, 70.8% vs. 42.2%, $\chi 2(1) =13.19$, $p <.001$ (primary study).
- Bayesian analysis indicates that the programs produced by the pairs were better than those produced by the non-pairing students for two of the three programming assignments that we evaluated. For the other program we analyzed our analysis showed no difference in the quality of the programs produced.
- A significantly greater percentage of paired students submitted programs. For the three assignments studied, on average 97% of the pairs submitted programs. For the non-pairing students only 86% of the students submitted programs (secondary study). We believe that it is especially noteworthy that pairing students turned in their homework at higher rates than non-pairing students on the fourth and fifth assignments. The students disliked the fourth assignment and found it very challenging. The fifth assignment was due during the last week of the term, when students have many conflicting due dates in their other courses. We are very encouraged that students who pair, attempt the homework assignments at very high rates, even when they are frustrated or feel overwhelmed by their workload. Perhaps pairing students feel pressure not to let their partner down, or they encourage and motivate each other when they would otherwise give up. Pairing students' increased confidence and satisfaction may also be playing a role here (secondary study).

Finding 2: hypothesis 2 is generally supported by our data from primary study. Some results are given from the tertiary study and are identified. Specifically:

- The program assisgnment grades (out of 100%) for programs produced by the women working in pairs were significantly better than those of the women working alone, 86.9% vs. 70.1%.

- The women working in pairs reported greater confidence, on a 100 point scale, in their solutions, 86.8 vs. 63.0, F(1, 482) = 50.54, p<.001. For the tertiary study, our results are not statistically significant; however, they are practically significant. We have 82.6 vs. 72.6, (p>.353).
- The women working in pairs reported greater enjoyment, on a 7 point scale, of the programming process, M=4.9 vs. M=4.6.
- Although more women in the pair programming sections stayed in the class and took the final exam, 88.1% vs.79.5% and more women in pairs passed the class, 65.3% vs. 59.0%, due to the small numbers of women, these differences were not significant.
- Among the students initially intending a computer science major, and who passed the introductory course and remained at UCSC for at least a year, women who paired were more likely than women who worked independently to be in a computer science related major (59.5% vs. 22.2%), $\chi^2(1) = 4.14$, $p < .05$.

Finding 3: hypothesis 3 was generally not supported by our data. In most cases the use of pair programming affected men and women the same, that is, they both benefited. The two exceptions were in the area of confidence in programming solutions and retention. Specifically:

- Overall, students who paired reported significantly higher confidence in their program solutions (89.4%) than students who worked independently (71.2%), $F(1,482)=99.38$, $p<.001$, and men were significantly more confident (87.0%) than women (81.1%), $F(1,482)=14.62$, $p < .001$. There was also a significant interaction between pairing and gender with regard to reported confidence. Simple effects follow-up tests of the interaction indicated that pairing resulted in more confidence for both women (86.8% vs. 63.0%), F(1, 482) = 50.54, p<.001 and men (90.3% vs. 74.6%), F(1, 482) = 54.94, p<.001. The 24 point increase in confidence that pairing afforded women was even greater than the 15 point confidence boost experienced by men who had the benefit of pairing. Unpaired men reported 1.18 times greater confidence than unpaired women, while paired men reported 1.04 times greater confidence than paired women. Pairing seems to close the confidence gap between women and men.
- Regarding retention, among the students initially intending a computer science major, and who passed the introductory course and remained at UCSC for at least a year, women who paired were more likely than women who worked independently to be in a computer science related major (59.5% vs. 22.2%), $\chi^2(1) = 4.14$, $p < .05$. Given that for the similar subset of men we found 74.0% vs. 47.2%, $\chi^2(1) = 9.70$, $p < .005$, we see a possible positive impact on the gender gap. Without pairing we see that men are 2.13 times as likely as women to declare a CS-related major. When pairing is used, men are only 1.24 times as likely as women to declare a CS-related major.

Finding 4: hypothesis 4 was generally not supported by our data. We have not yet completed our analysis of data from the primary study with regard to hypothesis 4 but we do have a preliminary result that gender role attitudes were uncorrelated with performance or retention. We still have to investigate whether perceptions of the usefulness of computer science are related to the outcome measures.

We are currently analyzing the data from the tertiary study. All three of the instructors that taught using pair programming in the tertiary study have continued using pair programming even after the completion of our data collection. They are personally convinced that pair programming is preferable over solo programming and suggest to their students to use it. The instructors do not require pair programming in all of their courses, but some provide incentives to their students to give it a try.

Next Steps

We will continue the analysis on the tertiary study data. We will be looking at pass rates, programming grades, satisfaction, and confidence in solution. We don't think that we'll get any findings regarding hypothesis 2 because the number of women in the tertiary study is very low.

We need to address our 4th hypothesis, mentioned above using data from the primary study.

We are not completely confident that complexity measures should be used to determine program quality. Subjective measures such as meaningful variable names, well-organized methods, appropriate indentation and use of whitespace, and use of Booleans instead of two-valued integers as control predicates may be much better indicators of program quality. The programs in the secondary study need to be analyzed for these subjective measures.

Dissemination plans are being made and will continue throughout the coming year.

Open Issues after report in 2003

How can we disseminate our results so that more people hear about them?

Does pair programming work at the middle school level? Since interest in computers and computer science decreases around the fifth grade, we need to determine how to move this technique to the middle schools. The PI has worked with pair programming and middle school girls and has a publication describing some of the ways to do so [10].

Acknowledgments

This work was funded by National Science Foundation grant EIA-0089989. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation. We wish to thank undergraduate student Tristan Thomte and graduate students Wendy R. Williams, Wendy M. Limbert, Ya Li Wang, and Jennifer Bevan for help with the data. Additionally, we wish to thank computer science instructors Scott Brandt, Michael Burke, Vladimir Drobot, Susan Nerton, Alex Pang, and Ed Parrish for their participation. Finally, we wish to thank all those students at UCSC, SJSU, and Cabrillo College who were subjects of our studies.

References

1. Tech-Savvy Educating Girls in the New Computer Age Executive Summary. 2000, American Association of University Women Education Foundation (available at http://www.aauw.org/2000/techsavvy.html).
2. Williams, L.A. and R.R. Kessler, All I really need to know about pair programming I learned in kindergarten. Communications of the ACM, 2000. 43(5): p. 108-114.
3. Bevan, Jennifer, Linda Werner, and Charlie McDowell, Guidelines for the Use of Pair Programming in a Freshman Programming Class, Proceedings of the 15th Conference on Software Engineering Education and Training (CSEE&T), 25-27 February 2002, p. 100-107. Covington, KY:IEEE Computer Society.
4. McDowell, Charlie, Heather Bullock, Julian Fernald, and Linda Werner, The Effects of Pair-Programming on Performance in an Introductory Programming Course, Proceedings of 33rd Technical Symposium on Computer Science Education SIGCSE, 27 Feb.-3 March 2002, p. 38-42. ACM Press.
5. Williams, Laurie, Charlie McDowell, Nachiappan Nagappan, Julian Fernald, Linda Werner (2003). Building Pair Programming Knowledge Through a Family of Experiments, IEEE International Symposium Empirical Software Engineering.
6. McDowell, C, Brian Hanks, and Linda Werner (2003). What Constitutes a "Better" Student Program, poster at 34th Technical Symposium on Computer Science Education (SIGCSE), February, 2003.
7. McDowell, Charlie, Linda Werner, Heather Bullock, and Julian Fernald, The Impact of Pair Programming on Student Performance, Perception, and Persistence, Proceedings of the 25th International Conference on Software Engineering (ICSE), pages 602 - 607, May 3 - 10, 2003.
8. McDowell, Charlie, Brian Hanks, and Linda Werner, Experimenting with Pair Programming in the Classroom, Proceedings of the 8th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE), June 30 - July 2, 2003, Thessaloniki, Greece.
9. Hanks, Brian, Charlie McDowell, David Draper, and Milovan Krnjajic (2004). Program Quality with Pair Programming in CS1 Annual Joint Conference Integrating Technology into Computer Science Education archive, Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education, 176-180.
10. Werner, Linda, Jill Denner, and Steven Bean (2004). Pair Programming Strategies for Middle School Girls. Proceedings of the Seventh IASTED International Conference Computers and Advanced Technology in Education, August 16-18, 161-166.
11. Werner, Linda, Brian Hanks, and Charlie McDowell (2004). Female Computer Science Students Who Pair Program Persist. Submitted for publication.