By Charlie McDowell, Linda Werner,
Heather E. Bullock, and Julian Fernald

Pair Programming Improves Student Retention, Confidence, and Program Quality

Pair programming produces more proficient, confident programmers—and may help increase female representation in the field.

In recent years, the growth of extreme programming (XP) has brought considerable attention to collaborative programming. Developed over a 15-year period by Kent Beck and his colleagues, Ron Jeffries and Ward Cunningham [1], XP is a computer software development approach that credits much of its success to the use of pair programming by all programmers, regardless of experience [9]. The pair programming dimension of XP requires that teams of two programmers work simultaneously on the same design, algorithm, code, or test. Sitting shoulder to shoulder at one computer, one member of the pair is the "designated driver," actively creating code and controlling the keyboard and mouse. The "non-driver" constantly reviews the keyed data in order to identify tactical and strategic deficiencies, including erroneous syntax and logic, misspellings, and implementations that don't map to the design. After a designated period of time, the partners reverse roles. Code produced by only one partner is discarded, or reviewed collaboratively before it is integrated.

Women and minorities continue to be underrepresented in computer science, and the number of women pursuing college majors in this area is declining. In 1985, 37% of computer science bachelor's degrees were awarded to women; in 2001 that percentage was down to 28% [11]. A number of variables have been proposed to account for this gender difference, including traditional socialization practices that reinforce math and science as male domains, lower confidence ratings and greater math anxiety among women, and women's tendency to take fewer advanced mathematics courses. The belief that computer science is a competitive, alienating field may further discourage women from pursuing careers in this area [5].

Pair programming, when used as a form of collaborative learning, has been shown to increase the number of women (and men) persisting in their previously stated intent to pursue degrees in computer science. In addition, paired teams have been found to significantly outperform individual programmers in terms of program functionality and readability, to report greater satisfaction with the problem-solving process, to have greater confidence in their solutions, and to be more likely to complete a programming assignment [10]. Nevertheless, many instructors continue to require students to complete programming assignments independently. Presumably, continued reliance on solo programming in academic settings is rooted in instructor concern that at least one of the partners in a pair will not learn as much as if he or she completed the assignment alone. In the worst case, one member of the pair might do essentially all of the work. Although this would not be "pair programming," it is often difficult, if not impossible, to monitor how students actually spend their programming time and how closely they are following the pairing protocol.

A Study of Pair Programming

We investigated the effects of pair programming on student performance and subsequent pursuit of computer science-related degrees among both female and male college students taking an introductory programming course designed for computer science-related majors (computer science, computer engineering, and information systems management). We collected data on 554 students who attempted the course at the University of California-Santa Cruz [4]. Data was collected from a total of four sections of the course: Fall 2000, Winter 2001 (two sections), and Spring 2001. One of the principle investigators of this study, Charlie McDowell, taught the Fall and Spring sections of the course. The Winter 2001 sections were taught by UCSC faculty members not associated with this project.

Students in the spring section were required to complete programming assignments independently. Students enrolled in the other sections were required to complete all assignments using pair programming. On the first day of class, students in the pairing sections were given a brief 15 to 20 minute description of pair programming and instructed to read Williams and Kessler's article "All I Really Need to Know about Pair Programming I Learned in Kindergarten" [7]. As an incentive they were told the first quiz might include a question on the article.

Students in the pairing sections submitted a list of three names of potential partners, and partners were assigned based on these preferences. In nearly all instances, students