National Center for Women & Information Technology

PROMISING PRACTICES

Pair Programming (Case Study 1)

Retaining Women through Collaborative Learning

K-12 Education | **Undergraduate** | **Graduate**

Pair programming assignments have contributed to greater retention of both male and female students at the University of California-Santa Cruz(UCSC). In 2000, Linda Werner and colleagues undertook research to understand the effect of collaborative learning on the retention rate of female students in computer science. Based on the overwhelmingly positive evidence on collaborative learning for student outcomes, a secondary goal of the study has been to measure improvements in student achievement compared to non-paired students.

At UCSC, pair programming research began in introductory courses and now has been expanded to advanced courses. The research also has included introductory courses at both San Jose State University and Cabrillo College, a two-year state community college.

Research results show that pair programming:

- Increases the percentage of introductory students (especially women) who declare a computer science major;
- Increases the number of students who remain in the computer science major one year later, as compared to their non-paired peers;
- Reduces the so-called "confidence gap" between female and male students, while increasing the programming confidence of all students;
- Leads to higher-quality student programs relative to non-paired students' programs (a link to complete research results is provided below.)

Implementing the program can be as easy as simply telling students they can work with a partner if they want. However, it is more likely that faculty will require that students pair off and, therefore, that it will require more faculty time and resources to implement and manage the course. Preparation involves establishing guidelines and mechanisms to help students pair properly and to keep them paired. For example, students should take turns "driving the mouse." In addition, effective preparation requires contingency plans in case one partner is absent or decides not to participate for one reason or another. In these cases, making it clear that the active student will not be punished because the pairing did not work well is important. Effective pairing attaches students of similar (though not necessarily equal) abilities to each other as partners; pairing mismatched students often can lead to unbalanced participation. Faculty must impress upon students that pairing is not a "divide-and-conquer" strategy, but rather a true collaborative effort in every endeavor for the entire project.

Some faculty have been concerned that this strategy enables students to "slack off" or receive grades higher than what they deserve because of the effort of their partner. Werner suggests that faculty avoid pairing very weak students with very strong students.

Like faculty, students also have concerns. They may have experience with poorly managed or implemented collaborations. When pairing is not required, applying pressure to partner up or offering extra incentives can help motivate students to pair, especially with advanced students. Some faculty members have found it helpful to require students to pair for only one or two assignments.

RESOURCES

Please see NCWIT's Pair Programming-in-a-Box: The Power of Collaborative Learning, a set of resources that helps instructors of introductory college programming classes to easily integrate pair programming into their courses, http://www.ncwit.org/pairprogramming.

Research and implementation guidelines for pair programming: http://www.soe.ucsc.edu/~charlie/projects/pairprogramming

Resources about agile education techniques: http://agile.csc.ncsu.edu/education/

Smith, K., Sheppard, S., Johnson, D., Johnson, R. (January 2005). Pedagogies of engagement: classroom- based practices. *Journal of Engineering Education*, 87-101.

Many web sites serve as portals for implementation of collaborative environments in higher education. Here is one: http://www.iasce.net/resources.shtml

Stephens, J. (May 2004). Justice or Just Us? What to Do About Cheating. *Carnegie Perspectives*. http://www.carnegiefoundation.org/perspectives/perspectives2004.May.htm

Cockburn, A. & Williams, L. (2001). The Costs and Benefits of Pair Programming. In G. Succi & M. Maresi (Eds.), *Extreme programming examined* (pp. 223-247). Boston, MA: Addison-Wesley. http://collaboration.csc.ncsu.edu/laurie/Papers/XPSardinia.PDF

NCWIT offers practices for increasing and benefiting from gender diversity in IT at the K-12, undergraduate, graduate, and career levels.

This case study describes a research-inspired practice that may need further evaluation. Try it, and let us know your results.

ncwit.org **Authors | Lecia Barker and J. McGrath Cohoon**

NCWIT Investment Partners: National Science Foundation, Avaya, Microsoft, Pfizer, and Bank of America

National Center for Women & Information Technology

PROMISING PRACTICES

How Do You Retain Women through Collaborative Learning?

with Case Study 1

K-12 Education | **Undergraduate** | **Graduate**

A collaborative learning environment occurs anytime an instructor requires students to work together on learning activities. Collaborative learning environments can involve both formal and informal activities and may or may not include direct assessment. For example, pairs of students work on programming assignments; small groups of students discuss possible answers to a professor's question during lecture; and students work together outside of class to learn new concepts. Collaborative learning is distinct from projects where students "divide and conquer." When students divide the work, each is responsible for only part of the problem solving and there are very limited opportunities for working through problems with others. In collaborative environments, students are engaged in intellectual talk with each other.

A long tradition of research shows that well-managed collaborative learning environments lead to improved student outcomes, including:

- improved critical thinking
- increased retention, especially for women advancing from the introductory to second course
- appreciation of diversity
- development of social and professional skills

In computer science, collaborative learning environments can improve retention and student learning. Pair programming is shown to improve retention of both female and male undergraduates. Students who pair in their introductory programming classes gain more confidence in programming than do their non-paired peers, are more likely to complete and pass the class, and are more likely to persist in IT majors. Aside from project-based courses, there are many other educational interventions based in collaborative learning, such as peer teaching and the non-lecture based "conversational classroom."

Collaborative learning environments have to be carefully planned and managed by instructors. For example, research shows that when there is too large a gap in collaborators' experience or knowledge, the benefits of collaborative learning disappear. Tasks must also be "shareable" for true collaboration to occur.

IS COLLABORATION CHEATING?

In an educational system that rewards individual work, student collaboration is often considered cheating. Computing syllabi around the world have statements like, "we encourage collaborative learning," but "students must turn in individual work." When students work together to design solutions to homework problems, however, their individual work may strongly resemble that of their collaborators.

So how do you know whether a student learned or cheated? The answer requires careful thought about the behaviors that count as academic dishonesty. In the working world, individuals rarely complete assignments in isolation. The challenge for faculty is to find ways to assess individual outcomes, while leveraging the benefits of collaboration.

Experts recommend revisiting course design and assessment and explicitly and concretely discussing the behaviors that will be interpreted as cheating, and they encourage faculty to make assignments meaningful to students and to explain the value of what students will learn by completing them. Most university campuses have resource centers that will work with faculty to integrate collaborative learning in ways that deter academic dishonesty.

Most capstone courses in computing education include project learning. However, student collaboration should be introduced early, often, and in both graded and un-graded situations to give undergraduates greater experience and to avoid early socialization that computing is a career in which people work alone. Pair programming (see reverse) has been shown to be effective for realizing increased retention, application to the major, and learning outcomes, and is a frequently practiced method in the workforce. While implementing collaborative environments is complex, the benefits to students, faculty, and industry appear to be worth the cost.

NCWIT offers practices for increasing and benefiting from gender diversity in IT at the K-12, undergraduate, graduate, and career levels.

Visit www.ncwit.org/practices to find out more.

ncwit.org **Authors | Lecia Barker and J. McGrath Cohoon**

NCWIT Investment Partners: National Science Foundation, Avaya, Microsoft, Pfizer, and Bank of America