The Benefits of Pairing By Ability

Grant Braught  John MacCormick  Tim Wahls

Department of Mathematics and Computer Science
Dickinson College
Carlisle, PA
Pair Programming

- Key component of Extreme Programming (XP) methodology
  - Two programmers, one machine
  - Defined roles
    - driver / navigator
  - Dynamic transitions between roles
Pair vs. Solo Programming

- Students who pair-program are more likely to:
  - Receive a C or better in the course
  - Enjoy the course
  - Have greater confidence in their work
  - Continue in computer science
  - Develop stronger individual programming/testing skills

[Williams et al. 2007; McDowell et al. 2003, 2006; Mendes et al. 2006; Carver et al. 2009; Braught et al. 2008; 2010]
Pairing Methodology

Effects of how pairs are formed:

- **Pair Compatibility**
  - Students that perceive their partner to be of equal or greater ability report higher pair compatibility.
  - [Sherriff et al. 2010; Williams 2006; Chaparro et al. 2005; Katria et al. 2004, 2005; Melnik et al. 2002]

- **Program Quality**
  - Pairs with similar programming confidence produce better programs. [Thomas et al. 2003]
  - Pairs with heterogeneous personality types produce better programs. [Sfetsos et al. 2009]
Our Question

- Does how pairs are formed impact the development of the programming and testing skills of the individuals?

Random Pairings

- vs -

“Ability” Pairings

- vs -

No Pairs
Study Context

- CS1 at Dickinson College
  - Objects first introduction to programming
    - Java using the BlueJ text and IDE
    - Emphasis on unit testing
  - Three 50-minute lectures / week
  - One 2-hour lab / week
  - Maximum of 24 students per section
  - Primarily non-majors
    - Course satisfies all-college general education requirements:
      - Laboratory Science
      - Quantitative Reasoning
Data Set

- 259 total students
  - 7 Ability Pair Sections \((n=142)\)
  - 2 Random Pair Sections \((n=41)\)
  - 4 No Pairs \((n=76)\)
- Sections taught by 4 different instructors
  - Common lecture notes and in-class examples
  - Common homework and lab assignments
  - Identical exams within semesters
Students used pair programming:
- During lab periods
  - pairs assigned randomly for first several weeks
  - pairs reassigned every 2-4 weeks
  - roles changed every 12-15 minutes during lab
Metrics

- Primary Metric:
  - Lab Exams (Hands-on Programming)
    - 70% Web-CAT Auto-Graded
    - 30% Instructor Graded

- Other Metrics:
  - Homework
  - Written Exams
  - Lab Assignments
A Better Metric?

The only valid measurement of code quality: WTFs/minute
Previous work suggests:

weaker students trained to program in ability pairs do better on solo programming compared with no pairs.
Previous work suggests:

- Weaker students trained in ability pairs do better on solo programming compared with no pairs or random pairs.
Evidence weakly suggests that the lowest quartile does better with ability pairs, compared to random pairs.
Evidence weakly suggests that the lowest quartile does better with ability pairs, compared to random pairs.

Is this significant?  p-value = 0.12  (merely suggestive; worthy of further study)
Evidence also confirms that the lowest quartile does better with ability pairs, compared to no pairs.
Two other results are described in the paper:

1. There is extremely weak evidence for the same effect on written work; again, further investigation is merited.

2. There is very strong evidence confirming that paired students perform better on paired assignments.

Now, back to the effect of the type of pair on solo programming
A possible theory: ability pairs are “compatible”

matched ability of pairs

this paper

improved programming performance

Caveat: some studies find no evidence of these links. The quoted studies mostly find weak evidence.

self-reported compatibility of pairs

Katira et al 2005
Katira et al 2005
Thomas et al 2003
If you were a weak student, would you rather be paired with . . .

- . . . a *strong* student?
  - Pro: strong student teaches weak student
  - Con: strong student solves problems too quickly for weak student to understand the solution

- . . . another *weak* student?
  - Pro: two weak students must discuss problems and solve them together

not a strong effect, even with strict rotation of “driver” role?
Acknowledgements

- Special thanks to:
  - Nathan Vaillete
  - Stephen Edwards & the Web-CAT Team
  - The National Science Foundation
  - All of the students who participated
Evidence weakly suggests that the lowest quartile does better with ability pairs, compared to random pairs.

![Graph showing programming exam score vs. percentile for ability pairs and random pairs. The graph indicates that ability pairs perform better in the lower quartile.]