Pair Programming in the Classroom

Mark Sherriff
University of Virginia

July 16, 2009

Some material courtesy of Laurie Williams, NCSU

Computer Science
at the UNIVERSITY *of* VIRGINIA

Overview

- What exactly is Pair Programming?

- The Case for Pair Programming

- The Costs

- Guidelines for a successful pairing experience

- Myths and Legends

- Resources

Pair Programming Definition

- "Pair programming is a style of programming in which *two* programmers work side-by-side at one computer, continuously collaborating on the same design, algorithm, code, or test."
  - Laurie Williams

The Roles

- ## The Driver
  - The person with "control" of the computer
  - Does the bulk of the typing

- ## The Navigator
  - Actively follows along with the driver with comments
  - Can take over at any time

Computer Science
at the UNIVERSITY *of* VIRGINIA

Partners vs. Pair Programming

- How is Pair Programming different than just assigning partners?

- Partnering:
  - "You go do this part and I'll go do this part and then we'll put it back together."

- Pair Programming:
  - "Let's first do this part together, then we'll tackle the rest."

Why Pair Programming?

- Pair programming students tend to:
  - Make it through the first class
  - Perform comparably or better on exams and projects
  - Perform just fine in future solo programming
  - Stick with computer science

Why Pair Programming?

- An instant support system
  - We have found that pairing cuts down on nearly all of the "trivial" questions (syntax, assignment clarification, etc.) and a large portion of the more complex questions (debugging, etc.)
  - We have been able to reduce the number of TAs for some courses
  - Instructor office hours are much quieter, and the instructor can spend more time with students that need more help

Computer Science
at the UNIVERSITY of VIRGINIA

7

Why Pair Programming?

- Sometimes it **is** a numbers game

- In a lab of 40 students…

  - having 20 pairs makes it easier for TAs to get to everyone

  - 20 assignments are easier/faster to grade than 40

All for the low, low price of…!

- What's the cost?

- Pair management
  - How do we assign pairs effectively for every assignment so it's not too much overhead?

- Pair evaluation
  - How can students report pair issues to the staff?

- Grading
  - How do you "split" grades up between partners?

Pair Management and Evaluation

- PairEval
  - Available for free from NCSU Realsearch Group
  - http://agile.csc.ncsu.edu/pairlearning/
- Personality and Work Ethic Tests
- Assignment/Partner Creation
- Reporting Tool
- How to "split" grades

Computer Science
at the UNIVERSITY *of* VIRGINIA

10

PairEval

PairEval

Tapestry 2009

PairEval

PairEval

Select your partner be evaluated: Yonghee Shin ▼

Has the student attended your group meetings?	rarely ▼
Has the student notified a teammate if he/she would not be able to attend a meeting or fulfill a responsibility?	never ▼
Has the student made a serious effort at assigned work before the group meetings?	never ▼
Does the student attempt to make contributions in group meetings when he/she can?	sometimes ▼
Does the student cooperate with the group effort?	rarely ▼
Assess the technical competency of your partner relative to yourself.	Weaker than me ▼
Assess how compatible you and your partner were	Very Compatible ▼

Very Compatible
OK
Not Compatible

Overall rating

○	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, acceptable 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, reraly prepared.
○	Unsatisfactory	Consistently failed to show up or complete assignments, unprepared.
◉	Superficial	Practically no participation.
○	No show	No participation at all.

Comments: no more than 255 characters.

She never met with us outside of lab and very rarely did any sort of work.

Computer Science
at the UNIVERSITY *of* VIRGINIA

PairEval

Peer Evaluation Report

CSC 326 Section All Assignment 2 Review 1

O = Overall Rating
Cty = Compatibility

	Assignment 1a #1			Assignment 2 #1				Assignment 2 #2		
Michael Gegick 001	Partner	O	Cty	Partner	O	Cty	Comment	Partner	O	Cty
	Sarah Heckman	9	Y	Jiang Zheng	9	Y				
Sarah Heckman 001	Partner	O	Cty	Partner	O	Cty	Comment	Partner	O	Cty
	Michael Gegick	9	Y	Chih-wei Ho	9	Y	Great partner!			
Chih-wei Ho 001	Partner	O	Cty	Partner	O	Cty	Comment	Partner	O	Cty
	Mark Sherriff	4	OK	Sarah Heckman	8	Y				
Mark Sherriff 001	Partner	O	Cty	Partner	O	Cty	Comment	Partner	O	Cty
	Chih-wei Ho	9	Y	Yonghee Shin	1	N	She never showed up. I did the whole thing by myself.			
Yonghee Shin 001	Partner	O	Cty	Partner	O	Cty	Comment	Partner	O	Cty
	Jiang Zheng	9	N	Mark Sherriff	8	N	Didn't get along.			
Jiang Zheng 001	Partner	O	Cty	Partner	O	Cty	Comment	Partner	O	Cty
	Yonghee Shin	2	N	Michael Gegick	7	OK				

Computer Science *at the* UNIVERSITY *of* VIRGINIA

15

The Biggest Cost

- Training!
- Instructors, TAs, **and students** need to be taught how to do effective pair programming in a controlled environment!
- The controlled environment could be a closed lab or lecture-lab system.

But we don't have a closed lab?

- ## CS1:
  - Pair programming not advisable (partner-split mentality!)
  - Use for in-lab work only

- ## CS2:
  - Proceed with (extreme) caution
  - At least bond in lab + some outside work

- ## CS2+:
  - After at least one paired class
  - Bonding still beneficial, outside work fine

Getting Involved

- **Instructors and Teaching Assistants have to take an active role in lab**

  - Must monitor and approach pairs if they seem to be dysfunctional

  - Should "strongly encourage" drivers and navigators to switch

- **Instructors also must understand that some pairings are just not going to work**

  - Don't let it discourage you!

Computer Science
at the UNIVERSITY *of* VIRGINIA

How Many Pairings Fail?

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

Computer Science *at the* UNIVERSITY *of* VIRGINIA

Tapestry 2009

Guidelines To Follow

- Strict tardiness / absence policy must be followed for pair activities to guard against lazy partners.
  - Loss of partner, points, and bad evaluation
- There **must** be a reporting mechanism for students to provide feedback on partners
  - PairEval works pretty well; also CATME
  - "If you could rate your effort based on 100%.."

Guidelines To Follow

- ## Assignments should be a bit more challenging
  - "Softball" assignments tend to be finished by a single person without consulting their partner

- ## The environment for pairing must be conducive to pairing

Lab Setup
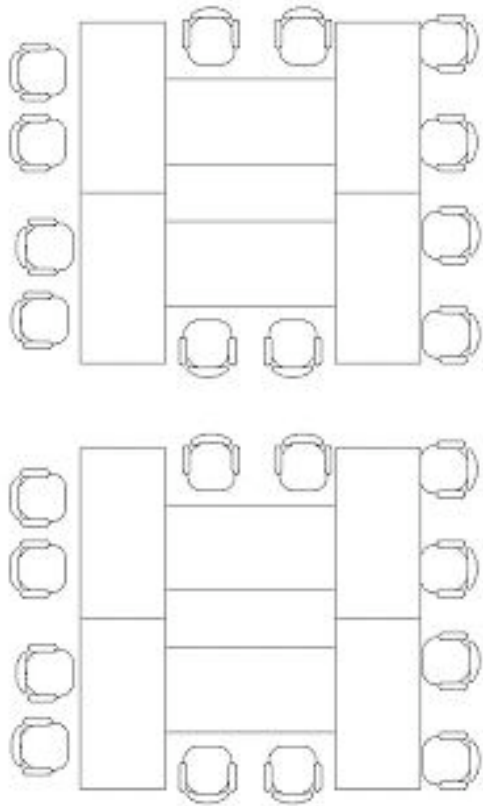
Guidelines To Follow

- ## Don't go overboard!
  - Pairing isn't for every assignment
  - There must be a balance (in work and in grade)

- ## Pairing **is not random!**
  - Thought must go into pairs
  - Students should ideally get new partners for every assignment

Pair Rotation

- Reassign several times per semester
- Good for students
  - Get to meet new people, learn about working with new people
  - If they don't like their partner, they know they will get a new one soon
- Good for instructor
  - Multiple forms of feedback
  - Natural handling of dysfunctional pairs

Myths and Legends

- Myth: Half the students will learn
  - *"In the first course, students need some time to absorb the ideas themselves."*
  - *"My inclination is to allow more group work starting in the second course."*
  - *"We want to be sure that each student writes enough code him/herself to learn the introductory concepts."*
  - *"I am against pair-programming in introductory courses, where students need to develop strong programming skills themselves."*

Computer Science
at the UNIVERSITY *of* VIRGINIA

Tapestry 2009

Myths and Legends

- In fact, all the students learn pretty well…

  - Studies at NCSU and SDSU showed that exam scores were comparable or improved for all students in introductory classes

  - Also, the percentage of students whose grade in CS2 went down by over 1/3 of a grade dropped once pairing was used in CS1

Williams, L., Layman, L.,
   Lab Partners: If They're Good Enough for the Sciences, Why Aren't They Good Enough for Us?,
   Conference on Software Engineering Education and Training (CSEE&T '07)

Myths and Legends

- ## Myth: Cheating will increase

  - *"With loose rules about who partners are, people will just pass code around. There has to be structure!"*

  - *"Old partners may feel obliged to help their former teammates."*

Myths and Legends

- Think about it a little differently…

- When we provide partners, students now have a support system they can turn to
  - Anecdotal evidence from students indicated that the stress of feeling alone and isolated made them consider cheating

- Two people now have to agree on cheating!
  - Well… there are exceptions to this one…
  - Moss is still a valuable tool

Other Guidelines and Myths

- Any others to add?

Resources

- http://agile.csc.ncsu.edu/pairlearning/

- http://www.ncwit.org/pairprogramming

- My email: sherriff@virginia.edu

Computer Science *at the* University *of* Virginia