

# Pair Programming

Luther Tychonievich

Tapestry 2013

# **A single practice that**

- Reduces teacher work
- Increases student enjoyment
- Increases student retention
- Increases student learning
- Appeals to girls & boys
- Enables cooler projects
- Increases employability

# Structured Cooperation

- Two (2) people
- One (1) focus
- Defined roles: Driver + Navigator
  - 1 driver, rest navigators
- Role switches
  - Switching protocols:  
give      take      timer      milestone

# Activity

- 1 paper, 1 pen
- Draw an outdoor scene
- Include at least 3 of
  - pet, livestock, fisher, children, river, trees, hills, crops, house, sunset, fantasy element

# Forming Pairs

- I know of studies that suggest
  - Random
  - Similar ability
  - Homogeneity
  - Self-selection
  - Orchestrated
  - Diverse ability
  - Heterogeneity
  - Personality Type
- No clear winner  
(middle school: pair within gender)

# Soloists and Experts

- Pro solo:
  - Don't chain expert to novice
- Pro pair anyway:
  - Cooperation important skill
  - Ego  $\neq$  ability
- Suggestion: decide case-by-case

# Tips for Students

- Talk & Listen
- Patience & Respect
- Breaks
- Cleanliness, Breath, ~~Perfume~~
- Personal Space
- Don't be intimidated

# Tips for Teachers

- Control switching (at first)
- Unsolicited feedback on dynamic
- Large project, open upper bound
- Take Qs from pair, not person
- Pairs “break;” can work through, re-pair, make triples, or make solo



# Grading (1 of 2)

- Post-survey: “% of work you/partner did”
  - Scale grades accordingly
  - Disagree? Talk with each
- Some skip this step...
- ...some increase its complexity

# Grading (2 of 2)

- Mix solo, pair, and group work
  - Individual assessment
  - Think-pair-share
  - Simultaneous feedback (clickers)
- Only pair large-enough projects
  - Each should drive 2+ times

# Caveats

- Tolerate increased volume
- Projects need to be  $\sim 2\times$  larger
- More human management
- Change details at your own risk

# Industry

- Pair part of Xtreme Programming
- Throughout development
- Pair vs. two independent:
  - Pair ~15% less productive
  - Pair ~85% higher quality
- Coding Dojos, Mob Programming
  - 1 driver,  $n$  navigators

# Resources

[cstapestry.wikidot.com/  
pair-programming](http://cstapestry.wikidot.com/pair-programming)

- NCWIT Pair Prog. in a Box
- Online group management tools
- Video for students to watch
- Slides with citations

# **A single practice that**

- Reduces teacher work
- Increases student enjoyment
- Increases student retention
- Increases student learning
- Appeals to girls & boys
- Enables cooler projects
- Increases employability