Pair programming

Two people, one computer

Roles:
  driver
  navigator
Benefits

- Produces better software
  - simpler designs
  - fewer defects
- Spreads knowledge and expertise
  - avoid single point of expertise
  - obvious for training/mentoring
- Reduces wall-clock time
- Improves learning
- Team building
  - communication, bonding
  - morale higher, more fun
My observations

Getting stuck less often
Becoming distracted (“pair pressure”)
Challenged by another POV
Following good practices
Higher morale (more fun)
Reduced fear (“my pair has my back”)
Facilities: mortal enemy

- Single computer per pair (cost savings!)
- Wide tables without legs
- Open spaces
- Cubicles are often terrible
- Noise level
Studies

• Cockburn & Williams
  – 15% overhead for pairing (controlled, academic)
  – Improved design, defect rate, morale

• Jensen
  – 1975 study of Fortran project, 50k LOC
  – Productivity 2.2x greater in pairs (LOC/person-month)
  – 1000x reduction in defect rate
Pragmatic Pairing

• No compromise on
  – All new code
  – All design questions
  – All testing challenges

• Solo work
  – When there’s an odd number of developers
  – With most experienced person
  – On “cloning” work
  – On exploration/learning
Questions?