Paired Programming & Personality Traits

Andrew J. Dick
Red Hook Group
andrew@redhookgroup.com
Agenda

- Paired Programming
- Project and Team Overview
- Observations
- Personality Traits
- Interview Techniques
- Resources
Paired Programming

- All production code written by two developers sitting at one machine.
- Supports several other XP practices
  - Refactoring
  - Simple design
  - Collective code ownership
  - Unit testing
- Integral part of XP

*XP 2002*
Project Overview and Team Selection

- Project overview
  - Financial web application
    - Shortly after Kent’s 1st book released
  - Multi-phase
    - 1st phase six 3-week iterations (~five months)

- Team selection
  - Paired interviewing by authors
  - Aptitude and attitude vs. pure technical skills
    - Final team: 2 senior developers, 4 junior developers
    - Authors filled roles of coach and tracker
Project Environment

- Development environment
  - L-shaped desks
  - Common whiteboard for design
  - Integration machine

- Development process
  - Stand up meetings each morning
  - Developers made estimates for tasks
  - Developers chose their tasks
  - Pair switching was encouraged

XP 2002
Initial Observations

- Observations of the pairs
  - No dynamic interchange of roles
    - Design drift from simplest design
    - Limited refactoring performed
  - Lack of mentoring
    - No knowledge transfer
    - No velocity increase
- Pair programming removed after 4th iter.
  - Planned to reintroduce at later date

XP 2002
Post Removal Observations

- After paired programming removed
  - Observable surge in communication
  - Paired debugging occurred
  - Ongoing design discussion
  - Frequent refactoring sessions
  - Increased mentoring
- 2\textsuperscript{nd} phase cancelled due to business reasons
  - Prevented observation of subsequent team performance

\textit{XP 2002}
Personality Traits

- Authors successfully pair programmed
- Development team had difficulty
- What was the difference?
Personality Traits

- Authors successfully pair programmed
- Development team had difficulty
- What was the difference?
  - Communication
Personality Traits

- Authors successfully pair programmed
- Development team had difficulty
- What was the difference?
  - Communication
  - Confidence
Personality Traits

- Authors successfully pair programmed
- Development team had difficulty

What was the difference?
- Communication
- Confidence
- Comfortable with each other
Personality Traits

- Authors successfully pair programmed
- Development team had difficulty
- What was the difference?
  - Communication
  - Confidence
  - Comfortable with each other
  - Compromise
Communication

- Effective communication leads to
  - Design analysis
  - Test strategy
  - Ongoing code review

- Lack of communication leads to
  - Decreased mentoring
  - Design drift
  - Reduction in the potential of the pair
Confidence

- Confident pair will
  - Add new functionality
  - Refactor without mercy
  - Remove dead and redundant code

- Unconfident pair will
  - Maneuver around dead code
    - ‘Lavaflow’ anti-pattern
  - Minimize refactoring
    - Afraid to ‘break the build’

XP 2002
Comfortable

- Developers that aren’t comfortable
  - Afraid to make bold suggestions
    - Fear of ridicule

- Additional factors
  - Different work ethic
    - i.e. Habitual surfer
  - Professional etiquette
    - i.e. Personal hygiene differences

XP 2002
Compromise

• Pairs that can compromise
  ➢ Best design strategy
    ➢ Blend of best ideas from both individuals
  ➢ Perform better
    ➢ Less arguments about trivial details

• Pairs unwilling to compromise
  ➢ Prone to arguments
  ➢ Slower pace
  ➢ Impact on design quality and simplicity

XP 2002
Interview Strategies

- Communication
  - Ability to be succinct
    - Too much information as bad as too little

- Confidence
  - Problem solving with non-technical scenarios
    - Avoids purely technical traps
    - e.g. Locating economical path between two nodes on a directed graph
More Interview Strategies

- Comfortable
  - Personable
  - Willing to discuss answers
    - e.g. No ‘single word’ syndrome

- Compromise
  - Difficult to ascertain
  - Common code convention can work
    - e.g. Resistance to conformance is a warning sign
Conclusions

- Paired programming is a learned skill but
  - Certain traits appear more beneficial
  - Team needs successful examples to emulate
- To build a new team for pair programming
  - Use interview techniques to select candidates
- To introduce to an existing team
  - Select existing members who have demonstrated the personality traits as the core
Resources

**Websites**
- www.pairprogramming.com
- www.c2.com/cgi/wiki?CategoryPairProgramming
- www.xpprogramming.org
- www.redhookgroup.com

**Books**
- Pair Programming Illuminated
- Extreme Programming Explained
- Extreme Programming Explored
- Extreme Programming Applied

**XP 2002**