
CSE 142

Pair Programming

1/10/2003

(c) 2001-3, University of Washington

H-1

Outline for Today

- **Pair Programming**
 - What it is
 - Why we're doing it
 - Expectations
- **Demonstration**
 - Including an introduction to uwcse.graphics

1/10/2003

(c) 2001-3, University of Washington

H-2

Pair Programming

- **Observation:** two people are more effective than one
- **Idea:** work with a partner on the programming projects
 - Program together, don't split up the work
- **Sit at one computer together to program**
- **Roles**
 - Driver – person with the keyboard
 - Navigator – partner who is watching what is going on, contributes ideas, looks things up, catches bugs, critiques
- **Switch roles often**
 - At least every 15 minutes

1/10/2003

(c) 2001-3, University of Washington

H-4

Effective Pair Programming

- **Mutual respect**
 - Both partners have something to contribute, even if backgrounds are different
 - Never "you goofed" or "I goofed" – always "we goofed" (but more often, "we got it right!")
- **Share everything**
 - You and your partner should produce a single piece of work
- **Play fair – switch off regularly**
 - No "professional drivers"

1/10/2003

(c) 2001-3, University of Washington

H-7

Driver/Navigator

- **Driver is the one with the keyboard, who actually types the code**
- **Navigator is watching over the driver's shoulder**
 - Stay engaged
 - Contribute ideas
 - Look things up – books, documentation, ...
 - Catch typos/bugs
 - But do it politely
- **Together have a shared responsibility for the success of the project**
- **And remember, switch roles frequently**

1/10/2003

(c) 2001-3, University of Washington

H-8

Expectations

- **Better overall quality**
 - Navigator helps driver catch bugs before they become part of the code
- **Faster time to finish**
 - Better focus when working together
 - Can be intense, take breaks as needed
- **Someone to talk to – less likely to get stuck**
- **Collective teaching and learning**
 - Both partners will learn things from each other
- **Less frustration**
- **More confidence and satisfaction developing software**

1/10/2003

(c) 2001-3, University of Washington

H-9