
One Last Book, One Last Topic

- Code reviews / software inspections



What is a Software Peer Review?



- Having other people read the code
- Why do that?
 - Because, in many studies, it is the single most effective way to find and fix bugs
 - Human beings are pretty smart
 - Open source theory “many eyes make for shallow bugs”

What Kind of Review?



- Ad hoc “look at this”
- Peer deskcheck / “pass around”
- Pair programming
- Walkthrough
- Team Review
- Formal Inspection

Differences?



- Vary in amount of planning required, amount of formality, number of people and number of roles
- More heavyweight tend to be more effective, and more efficient (“more bugs for your buck”) but sometimes aren’t possible

Formal Inspections



- Include all of the following:

- Planning
- Preparation
- An actual meeting
- Correction of found defects
- Verification of correction



- Role of moderator/reader is not given to the person/people who created the code in a formal inspection

What Have We Learned?



- Software engineering is like other engineering disciplines
 - But it is also unlike other engineering disciplines
 - The way we do testing is one key difference
- Testing requires a special kind of thinking
 - Testing is applied epistemology
 - How to find out things about a program
 - Most common way to find out is by having a test case that makes the program fail

What Have We Learned?



- There are many kinds of testing
 - There is no one “right way to test”
 - Manual and automated testing both have a role
 - Random testing is an especially useful automated testing technique
- Coverage metrics help us measure what we have and have not tested
- Debugging is like the scientific method
 - Formulate hypotheses about what is wrong
 - Divide and conquer to narrow down the problem
 - Use evidence (tests and examining executions) to drive your hypothesis making