

Objectives

- Reviewing David Shepherd talk
- Project: Preliminary Implementation Demo

Dec 5, 2011

Sprenkle - CSCI209

1

Discussion of Friday's Talk

- Your questions?
- Lessons learned?
- How far from professional?

Dec 5, 2011

Sprenkle - CSCI209

2

Discussion of Talk

- *Effective Java*
 - Have looked at some examples (equals method)
- Testing using mock objects
 - "simulated objects that mimic the behavior of real objects in controlled ways"
- C#
- Refactoring
- SCM: software configuration management

Dec 5, 2011

Sprenkle - CSCI209

3

Discussion of Readings

- **5. Do you fix bugs before writing new code?**
- What they realized was that the project managers had been so insistent on keeping to the "schedule" that programmers simply rushed through the coding process, writing extremely bad code, because the bug fixing phase was not a part of the formal schedule.
- The story goes that one programmer, who had to write the code to calculate the height of a line of text, simply wrote "return 12;" and waited for the bug report to come in about how his function is not always correct.
- In the *post-mortem*, this was referred to as "infinite defects methodology".

<http://www.joelonsoftware.com/articles/fog0000000043.html>

Dec 5, 2011

Sprenkle - CSCI209

4

Discussion of Readings

- Microsoft universally adopted "zero defects methodology"
 - "zero defects": at any given time, highest priority is to eliminate bugs *before* writing any new code
 - In general, longer you wait before fixing a bug, costlier in time and money to fix
 - Example: make a typo or syntax error that the compiler catches, fixing it is basically trivial
 - When you have a bug in your code the first time you run it, will be able to fix it in no time because all the code is fresh in your mind
- Easier to *predict* how long to write new code than to fix an existing bug
- Keeping product *ready to ship* at all times

Dec 5, 2011

Sprenkle - CSCI209

5

David's PhD Research

- Improved search with natural language

Dec 5, 2011

Sprenkle - CSCI209

6

Review

- What is the decorator design pattern?
 - When should you use it?

Dec 5, 2011

Sprenkle - CSCI209

7

Demos

- 5 minutes with each team
 - Use time to coordinate with your team
- Reconvene and discuss design decisions and APIs

Dec 5, 2011

Sprenkle - CSCI209

8

My Notes

- FD vs FORWARD
- Context class
- My understanding of code
 - Which is the main
 - Comments on commits
- Statistics online
- Project Analysis: Make sure you understand the others' design/code/parts
 - At least at a high level

Dec 5, 2011

Sprenkle - CSCI209

9

To Do

- Dave Shepherd write up by Wednesday
- Regular Expressions extra credit due Friday
- Project implementation continued

Dec 5, 2011

Sprenkle - CSCI209

10