

Inspections

CS314 – Spring 2016

Why inspect? Inspect what?

Geri Georg

Sudipto Ghosh

James M Bieman

- Objective: EFFECTIVELY FIND DEFECTS
 - As many **different** defects as possible
 - With an **efficient** use of people
 - Each person is looking for specific things (role)
 - Roles: e.g. maintainer, tester, end user, support, ...
- Caper Jones and Olivier Bonsignour (2011*) report:
 - Most forms of testing are less than 30% efficient
 - Formal inspections from 50-85% efficient; some results 90-95% efficient
 - Static analysis similar, but only handle narrow set of code-related defects
- **Any** artifact can be inspected – examples:
 - User manuals
 - Requirements
 - Design
 - **Code**
 - Test plans/Tests

* <http://www.drdoobs.com/architecture-and-design/do-you-inspect/231903203?pgno=2>

1

When should an inspection happen?

- The earlier defects are found, the less expensive they are to fix
- Inspect BEFORE other “testing”
 - If you “test” first, and inspections cause substantial rework, testing is wasted
- Documents:
 - Requirements: before design or user feedback
 - Design: before any lower-level of design or formal checking (e.g. model, theorem proving, etc.)
 - Developer documentation: before general use by developers
 - User documentation: before iterations with users
- Code:
 - **after** successful compilation
 - **before** developer testing

2

Inspecting using Roles

If you are inspecting using a Role, this means you study the code from that perspective – you look for problems that a person who has this job might find. Checklists can be useful to help you decide what would be a problem for someone with this job.

End User role: What matters to the end user? Useful program messages, input validated quickly with suggestions on how to fix problems, program not crashing, sort cuts for experienced user efficiency, ...

Maintainer role: What matters to a maintainer/developer? Descriptive names for classes, variables, ..., consistent exception handling that handles all exceptions, small classes, functions, ..., good comments, ...

Tester role: What matters to a tester? Descriptive names, good comments, no hidden variables, validate inputs, document security-related information, ...

Moderator: This is really a job – the moderator is responsible for making an inspection meeting efficient and effective at finding defects. Only logging should occur – no fixing!

In the Design Studio, whoever has the Maintainer role is the “owner” of the code – they need to understand the problem, but discussion shouldn’t include anything else about how to fix it.

3

Example Logging Sheet

Inspector		Prep time		Inspector		Prep time	
Ann - Maintainer		30 min		Dan - Moderator		50 min	
Bob - Tester		60 min					
Cherese - End User		50 min					

Artifact Inspected: Adventure Game Code - Adventure.java, Room.java						Log Page <#> of <total> 1/...
Author: Ann						
Moderator: Dan						
Number	Location (page num, paragraph, line num (L), class)	Severity (High, Med, Low)	Reporter	Issue	Action (filled in by author later)	
1	Room: L 21	High	Ann	Unused imports need to be removed		
2	Adventure: L 52	High	Bob	Maze structure needs to allow variation - not be hard coded		
3	Adventure: L 52	High	Cherese	Treasure also needs to be variable (after you play it a few times you'd know what treasure was where)		
...						

4