Peering Into the White Box: A testers approach to code reviews

Alan Page
Microsoft
self-promotion slide

http://angryweasel.com/blog
http://www.hwtsam.com
http://twitter.com/alanpage
Code Reviews...really?
The Tester’s Point of View

Code reviews *traditionally* a developer activity

- Developers write unit tests
- Developers ask “does this work?”
- Developers look at code one way, but...

- But testers still find functional bugs
- Testers ask “how can this fail?”
- Testers have a different perspective

**Experiment**: Have testers perform code reviews
Types of Reviews

Less formal
Less effective
Less cost

More formal
More effective
Greater cost

Ad hoc review

Formal Inspection

Based on Peer Reviews in Software by Wiegers, 2002
Approach

1. Kickoff meeting
2. Collect volunteers
3. Set expectations
4. Schedule meetings
5. Share checklist
Using a Checklist

The illusion of attention

If you look for everything, you’ll miss something
If you look for one thing, you’ll miss other things

Mitigation

Loop one at a time through a list of common errors
Selecting Code for Review

- Churn
- Complexity
- Intuition
- Convenience
Results

Four teams
Moderate engagement
Skill increase
Successful “Bug Hunting”
new Text-only

(text))

ter to read the

ved.

il failed,

Comment [WL17]: [XD] Should we have two functions to convert to rtf or text?

Comment [TJL18]: Should yes, but this was discovered too late in the game. Under the hood the HtmlToText() could use HtmlToRtf() to do its work, but that implementation detail would be abstracted from the rest of the code.

Comment [WL19]: [WL] Shouldn’t this be ERROR?

Comment [TJL20]: Yes.
Comment [XD31]: It seems to me that we update this length too early. If any operations below fail, we never restore the length to the original value.

Comment [TJL32]: That’s valid. Should move it below the switch statement where we know the insertion operation succeeded.
Comment [WL27]: [XD] Why is this variable here as it is never used

Comment [TJL28]: Was probably useful in debugging at some point, can be removed

Comment [WL29]: [XD] Should we put “all” instead of any?

Comment [TJL30]: OK by me
Lessons Learned

Not like other testing
Balance w/other activities
The Checklist
Develop “Experts”
Development team involvement
Bug Tracking
Future Plans

Continue with select team members

Inspection “SWAT”

Track comments and discussion
Are code reviews for you?
Resources

• Software Inspection – Tom Gilb & Dorothy Graham. Addison-Wesley Professional (January 10, 1994)
• Software Inspection (web article) - http://www.thesoftware-experts.de/e_dta-sw-test-inspection.htm
• Are input parameters validated and checked for null?
• All exceptions must not be caught and sunk (i.e. not rethrown)
• Do the comments accurately describe the code?
• Examine loops for performance issues
• There must be no hard-coded strings and magic numbers in the code
• All user-visible strings must be localizable (i.e. retrieved from a resource file)
• Verify correctness of operators
• Is arithmetic safe for overflow and 64-bit math?
• Are the functions or source files exceptionally large or complex?
• Are COM interfaces properly ref counted?
• Are untrusted inputs validated?
• Check for properly constructed switch statements
• Are there unused elements in data structures?
• Are string buffers big enough for localized content?
• Does the new code duplicate code that exists elsewhere in the project?
• Are there unnecessary global variables?
• Is the code overly complex? Could it be simpler?
• Do the function and variable names make sense?
• All memory allocations should be checked for failure?
• Are comments appropriate and useful?
• Does the code have performance implications – if so, are optimizations possible?
• Are there spelling errors in names or comments?
• Have boundary conditions been checked?