Lightweight Peer Code Review

- Why code reviews work
- Why traditional inspections fail
- The modern view: Lightweight Reviews
- Cisco Systems: The largest-ever case study of peer review
- Managing social issues
- Final Thoughts
If you like it so much, 
why don’t you marry it?

- Developer time is expensive.

- Is it worth it?
Code Review Economics, Part I

Before Code Review

<table>
<thead>
<tr>
<th>Step</th>
<th>Bugs Remaining</th>
</tr>
</thead>
<tbody>
<tr>
<td>After Development</td>
<td>463</td>
</tr>
<tr>
<td>After QA/Test ($200/fix)</td>
<td>321</td>
</tr>
<tr>
<td>After Customer ($1000/fix)</td>
<td>194</td>
</tr>
</tbody>
</table>

Cost of fixing bugs: $174k

Total Cost: $368k
Code Review Economics, Part I

Before Code Review

Bugs Remaining

463

Bugs Remaining

321

Bugs Remaining

194

Cost of fixing bugs: $174k

+ Cost of 194 latent bugs: $194k

Total Cost: $368k

What if we had Peer Review?
Code Review Economics, Part II

Bugs Remaining

After Development 463
After Code Review 180
After QA/Test 113
After Customer 32

Cost of fixing bugs: $120k + Cost of 32 latent bugs: $32k = Total Cost: $152k

$152k, down from $368k

32, down from 194
What is a Code Review?

- Over-the-shoulder
  - Easy, but unenforceable and unmeasurable

- Email pass-around
  - Works remotely, but still unenforceable and unmeasurable

- Tool-assisted
  - More on this later…

- Formal Inspection
  - The “industry standard”
Traditional Code Inspections

- Michael Fagan, IBM, 1976

- Planning, Introduction, Reading, Meeting, Rework, Verification, Improvement

- Tom Gilb  Karl Wiegers
Motivation for New Methods

- **Time:** Impractical for most code
  
The 7% rule

- **Things have changed:**
  
  “Restore register values before returning from subroutine.”

- **Agile Methodology:**
  
  Lightweight process can be effective and fun.
Are Inspection Meetings Necessary?

- Lawrence Votta, 1993

- Top benefit of inspection meetings: Synergy: the Phantom Inspector

- Do we find more defects in meetings than we do by ourselves?
Inspection Meetings Don’t Find Defects

Defects Found By Inspection Phase

- Defects from Reading
- Defects from Meeting

4%
Are Meetings Good for Anything?

- Lawrence Votta, 1993, design reviews: 20% fewer false-positives; 4% new defects
- Diane Kelly, 2003, code reviews: 25% fewer false-positives; 10% new defects
- Reidar Conradi, 2003, architecture reviews: 20% fewer false-positives; 15% new defects
- False-positives often trivial
- Meeting time adds 7 man-hours per review
Cisco Systems: The Largest Case Study

- Cisco® MeetingPlace® product group
- 2500 reviews
- 3,200,000 lines of code
- 10 months
- 50 developers
- San Diego, Bangalore, Budapest
- Used Smart Bear’s Code Collaborator software
Cisco: Review Process Goals

- All code reviewed before checked into version control (Perforce)
- No in-person meetings
- Process is enforced by tools
- Metrics collected automatically
- Review-level and summary-level reporting
Cisco: Typical Review: Preparation

- Author uploads files from Perforce GUI client
- Author decides on 1-4 reviewers
Cisco: Typical Review: Inspection

- Review files using before/after side-by-side view

```java
if (!foundInLocalList) {
    // This might be because we don't have the right
    // also be because the changelist has no files
    // in particular. Cases 7191, 11015.
    IScmLocalCheckout[] localCheckouts = changelist.
    if (localCheckouts == null || localCheckouts.
```
```java
    System.err.println();
    System.err.println( "ERROR: Changelist "
    return false;
}

// Currently the only other explanation is a
System.err.println();
System.err.println( "ERROR: Pending changelist";
System.err.println( "Current: " + scmCl
```
Cisco: Typical Review: Inspection

- Review files using before/after side-by-side view
- Comments, defects threaded by file, line number
Cisco: Typical Review: Rework

- Authors fix defects
- Fixes uploaded for verification
Cisco: Typical Review: Verification

- Fixes verified before review is complete

```java
if ( ! foundInLocalList ) {
    // This might be because we
    // also be because the chang
    // in particular. Cases 719
    IScmLocalCheckout[] localCheckouts = crs.getLocalCheckouts();
    if ( localCheckouts == null ) {
        System.err.println();
    }
}
```
Cisco: Conclusions, Part I

- Lightweight reviews just as effective as Formal Inspections at uncovering defects.

- Average defect density: 40 defects / kLOC

- ...but lightweight review took 6.5 man-hours less time!
Cisco: Conclusions, Part II

- LOC under review should be <200 with a hard limit of 400.
Cisco: Conclusions, Part III

- Inspect slower than 500 LOC/hour for best defect detection results.
Cisco: Conclusions, Part IV

- Total review time <60 minutes.
Cisco: Conclusions, Part V

- Author-preparation results in few defects

![Diagram showing the effect of author preparation on defect density. The diagram compares defect density (Defects/kLOC) with and without preparation. The x-axis represents the number of author prep comments, while the y-axis shows defect density. The graph indicates a decrease in defect density with preparation.]
Cisco: Wrap-Up

- Lightweight reviews are effective and efficient.
- Review fewer than 200-400 LOC at a time.
- Leave enough time for proper review, but not more than 90 minutes.
- Author-preparation is good.
Social Aspects of Code Review

- The "Ego Effect"
- Personal Growth
- Hurt Feelings
- The "Big Brother Effect"
The “Ego Effect”

- Peer review invokes the ego
- “Jerry always forgets to check for NULL.”
- Immediate improvement in code quality
- Senior developers stop being lazy

- Still works with 33% review coverage
Personal Growth

- if ( "integrate".equals( s ) )

- Surprising Effects
  - Learning not related to the goal of the review.
  - Reviewer learns too.

- Fun!
Negative Emotions

- “Big Brother” Effect

- Hurt Feelings
Dealing with Negative Emotions

- Don’t single out a developer
- Defects are good!
- Defects ➔ Bugs not delivered to a customer
- Defects ➔ Reviewer is doing job

- No review metric can be used in performance evaluations
Final Thoughts

- Code reviews are cost-effective because defects found earlier are cheaper to fix.

- Lightweight reviews can be effective, efficient, measurable with tool support.

- Review: <200 LOC, <60 minutes, not too fast.

- Self-review or spot-check review is better than nothing
The End

- **Smart Bear:**
  Experts in lightweight peer review tools and techniques.

- **Code Collaborator:**
  Software assistance for agile peer review.

http://codecollaborator.com

(877) 501-5651