CS 4387/5387
SOFTWARE INTEGRATION AND V&V

LECTURE 2
SOFTWARE INSPECTIONS

Outline

- Short Quiz
- Lecture (Software Inspections and Reviews)
- Individual Inspection of DH
- Team Inspection Meeting
Quiz

Semester Topics

- Appraisal QA Techniques
  - Formal Reviews and Inspections
- Failure QA Techniques
  - Testing
- Formal Verification and Validation Techniques
  - Model checking, theorem proving, and runtime monitoring
- Developing and Monitoring Testing Plans
When are Defects Introduced?

- The majority of defects are introduced in earlier phases.
  - Requirements are the top factor in a project’s success or failure.

<table>
<thead>
<tr>
<th>Phase</th>
<th>% of defects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>56</td>
</tr>
<tr>
<td>Design</td>
<td>27</td>
</tr>
<tr>
<td>Code</td>
<td>7</td>
</tr>
<tr>
<td>Other</td>
<td>10</td>
</tr>
</tbody>
</table>

Cost of fixing defects

- Relative cost of fixing defects
  - benchmark: cost of fixing defect in requirements phase = 1

<table>
<thead>
<tr>
<th>Phase found</th>
<th>Cost ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>1</td>
</tr>
<tr>
<td>Design</td>
<td>3 - 5</td>
</tr>
<tr>
<td>Coding</td>
<td>10</td>
</tr>
<tr>
<td>Unit / integration testing</td>
<td>15 - 40</td>
</tr>
<tr>
<td>System / acceptance testing</td>
<td>30 - 70</td>
</tr>
<tr>
<td>Production</td>
<td>40 - 1000</td>
</tr>
</tbody>
</table>
Requirements Problem

- “The hardest single part of building a software system is deciding precisely **what to build**. No other part of the conceptual work is so difficult as establishing the detailed technical requirements, … No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later.”

Fred Brooks, “No Silver Bullet - Essence and Accident in Software Engineering”

The Requirements Process

- The **basic goal** of the requirements phase is to collect, analyze, establish, and specify **what** the software is do, **NOT how** to do it.
- Requirements Elicitation
  - collect requirements from the customers/clients/users
- Requirements Analysis
  - study, analyze and model the problem to be solved
- Requirements Specification
  - define and document the requirements in an organized and precise manner
- Requirements Verification
  - verify that the requirements are correct, complete, and consistent
Requirements Elicitation Principles

- Customers may not always be good at describing their needs.
  - Interviews
  - Prototype
  - Customer Stories (XP)

- There are two general types of requirements:
  - A functional requirement describes an interaction of a system with its environment.
    - “The program shall output (at the terminal screen) the names of the presidents (and their ages) in alphabetical order.”
  - A non-functional requirement describes a restriction that constrains how the problem is to be solved.
    - Maintainability, performance, reliability…
    - “The system must be maintainable”

Importance of Requirements

- Software requirements specifications play multiple roles:
  - Customers: Contractual basis for development
  - Managers: Scheduling and measuring progress
  - Designers: “Design-to” specification
  - Coders: Acceptable implementations
  - Quality Assurance: Basis for validation and test planning
What Makes Requirements Hard?

- Distinguishing between *What* and *How*
- Understanding customer needs
  - People don’t know what they want
- Communicating Properties
  - Software Properties are hard to visualize
- Multiple stakeholders
  - Mutual satisfaction of goals

Characteristics of Good Requirements Document

- Complete
- Implementation independent
- Unambiguous
- Consistent
- Precise
- Concise
- Verifiable
Easy to Use

Dilbert

YOUR USER REQUIREMENTS INCLUDE FOUR HUNDRED FEATURES.

DO YOU REALIZE THAT NO HUMAN WOULD BE ABLE TO USE A PRODUCT WITH THAT LEVEL OF COMPLEXITY?

GOOD POINT. I’D BETTER ADD “EASY TO USE” TO THE LIST.

Requirements Exercise
Quality: Testing, Inspection, and Analysis

- Testing is the most widely used approach to manage software quality
- Testing and inspection typically account for more resource use than actual design and coding
- Testing and inspection cannot find all defects
- Testing and inspection do not create quality. Development practices create quality
- Choices regarding testing and inspection are influenced by quality metrics visible to management
- There are emerging assurance techniques that complement test and inspection

Inspection – The Big Questions

1. What is inspection? And what are the benefits?
2. When are inspections better than testing? What kind of attributes? What is the typical experience of firms with inspection?
3. Are there different kinds of inspections? What are the relative benefits of each?
4. What gets inspected? And when to do inspections?
5. Who are the participants in inspection? What value is provided by each
6. How is the inspection process accomplished? What are summary guidelines for the meetings?
What is a Software Inspection?

- Meetings (real or virtual) during which a software artifact is reviewed by people other than the original developer
- Goal is to discover defects (not to fix them)
- Done as soon as there’s something to inspect

Benefits of Inspections - 1

- Using combined knowledge
  - Regarding designs and specific software artifacts and regarding defect detection practices
- Using different view points
  - Finding defects may be easier for people who haven’t seen the artifact before and don’t have preconceived ideas about its correctness

Improving the odds
Benefits of Inspections - 2

- Find flaws early
  - Can dramatically reduce cost of fixing them
  - During requirements and detailed design
  - Even before code is written or code that does not yet have a test harness
  - Or code in which testing has found flaws but root causes are not understood

- Reduce rework and testing effort
  - Can reduce overall development effort

Inspection – The Big Questions

1. What is inspection? And what are the benefits?
2. When are inspections better than testing? What kind of attributes? What is the typical experience of firms with inspection?
3. Are there different kinds of inspections? What are the relative benefits of each?
4. What gets inspected? And when to do inspections?
5. Who are the participants in inspection? What value is provided by each
6. How is the inspection process accomplished? What are summary guidelines for the meetings?
Inspections vs. Testing

- What attributes are well-handled by inspections but not testing?
  - “Fuzzy” non-functional properties Maintainability, reusability ...

- Other properties tough to test
  - Scalability, efficiency
  - Security, integrity
  - Robustness, reliability

- Requirements, architecture, design documents
  - Cannot execute these as tests

Industrial Experiences with Inspections - 1

- Raytheon:
  - Reduced "rework" from 41% of cost to 20% of cost
  - Reduced effort to fix integration problems by 80%

- Paulk et al.: cost to fix a defect in space shuttle software
  - $1 if found in inspection
  - $13 during system test
  - $92 after delivery

- IBM
  - 1 hour of inspection saves 20 hours of testing
  - Saved 82 hours of rework on defects in released product
Industrial Experiences with Inspections - 2

- **IBM Santa Teresa Lab**
  - 3.5 hours to find a bug with inspection, 15-25 through testing

- **C. Jones**
  - Design/code inspections remove 50-70% of defects
  - Testing removes 35%

- **R. Grady, efficiency data from HP**
  - System use: 0.21 defects/hour
  - Black box: 0.28 defects/hour
  - White box: 0.32 defects/hour
  - Reading/inspect: 1.06 defects/hour

---

Inspection – The Big Questions

1. What is inspection? And what are the benefits?
2. When are inspections better than testing? What kind of attributes? What is the typical experience of firms with inspection?
3. Are there different kinds of inspections? What are the relative benefits of each?
4. What gets inspected? And when to do inspections?
5. Who are the participants in inspection? What value is provided by each
6. How is the inspection process accomplished? What are summary guidelines for the meetings?
Kinds of Inspections (Reviews) - 1

- **Walkthroughs**
  - No advance preparation
  - Author leads discussion in meeting
  - No formal follow-up
  - Low cost, valuable for education

Kinds of Inspections (Reviews) - 2

- **Inspections / Formal Technical Reviews**
  - Participation defined by policy
    - Developers
    - Designated key individuals – peers, quality team, review board, etc.
  - Advance preparation by participants
    - Typically based on checklists
  - Formal meeting to discuss artifact
    - Led by moderator, not author
    - Documented process followed
    - May be virtual or conferenced
  - Formal follow-up process
    - Written deliverable from review
Kinds of Inspections (Reviews) - 3

- Other review approaches
  - Pass-around – preparation part of an inspection
  - Peer desk check – examination by a single reviewer (like pair programming)
  - Ad-hoc – informal feedback from a team member

Inspection – The Big Questions

1. What is inspection? And what are the benefits?
2. When are inspections better than testing? What kind of attributes? What is the typical experience of firms with inspection?
3. Are there different kinds of inspections? What are the relative benefits of each?
4. What gets inspected? And when to do inspections?
5. Who are the participants in inspection? What value is provided by each
6. How is the inspection process accomplished? What are summary guidelines for the meetings?
What to Inspect?

- **What do inspectors inspect?**
  - Requirements, design documents
    - Difficult to validate in other ways
    - May have high associated risk
      - Especially important to get right
      - Cheaper to fix earlier on in process
    - Many different perspectives are helpful
    - Need involvement of multiple stakeholders
  - Critical or uncertain pieces of code
    - Security-critical code
    - Safety-critical code

When to Inspect?

- **Start inspections at the earliest stages of process**
  - Catch mistakes early, when easy to fix
  - Allow rest of system to be built with knowledge gained
Inspection – The Big Questions

1. What is inspection? And what are the benefits?

2. When are inspections better than testing? What kind of attributes? What is the typical experience of firms with inspection?

3. Are there different kinds of inspections? What are the relative benefits of each?

4. What gets inspected? And when to do inspections?

5. Who are the participants in inspection? What value is provided by each?

6. How is the inspection process accomplished? What are summary guidelines for the meetings?

Who are the participants in inspection? - 1

- **Moderator**
  - Organizes review
  - Keeps discussion on track
  - Ensures follow-up happens

- **Key characteristics**
  - Good facilitator
  - Knowledgeable
  - Impartial and respected
  - Can hold participants accountable and correct inappropriate behavior

- **Separate role from Recorder**
  - Who captures a log of the inspection process
Who are the participants in inspection? - 2

- **Author**
  - Describes rationale for work
  - Not moderator or inspector
    - Conflict between objectivity required of moderator/reader and advocacy for the author's own work
  - Not recorder
    - Temptation to not write down issues the author disagrees with

- Significant benefits to attending
  - Gain insight from others' perspectives
  - Can answer questions
  - Can contribute to discussion based on knowledge of artifact

- Potential downside: meeting may be confrontational

Who are the participants in inspection? - 3

- **Inspector**
  - Not the author
  - Inspect whole or part of artifact
  - Previous preparation
    - Based on previously established checklist
  - Unbiased and objective
  - Preferably more than two inspectors

- Alternative
  - Get comments section by section
  - Faster, but does not capture differing perspectives as effectively
Inspection – The Big Questions

1. What is inspection? And what are the benefits?
2. When are inspections better than testing? What kind of attributes? What is the typical experience of firms with inspection?
3. Are there different kinds of inspections? What are the relative benefits of each?
4. What gets inspected? And when to do inspections?
5. Who are the participants in inspection? What value is provided by each?
6. How is the inspection process accomplished? What are summary guidelines for the meetings?

Process: Planning

- **Planning**
  - Determine objectives
  - Choose moderator
  - Identify inspectors
  
  - Good to involve people with connection to artifact
    - e.g. depends on, interfaces with, author...

- **Schedule meeting(s)**
  - General guideline: 150-200 LOC/hour, or 6 pages/hour

- **Prepare and distribute inspection package**
  - Deliverable, supporting docs, checklists
Process: Preparation

- Preparation
  - Inspectors examine deliverable
  - Defects: cause an error in the product
  - Non-defects: improvements, clarification, style, questions
  - May want to list typos/spelling/format/style separately and not discuss during the meeting

- General guideline
  - prep time ~ meeting time

Process: Meeting

- Moderator describes one segment at a time

- Inspectors respond: defects, questions, suggestions

- Recorder writes down each defect, suggestion, issue
  - This is the primary deliverable

- Moderator
  - Avoid problem solving, inappropriate behavior, lack of participation

- At conclusion: recorder prepares report with appraisal and data
Meetings: Review Guidelines - 1

- Build reviews into your schedule
  - Otherwise unexpected and viewed as intrusion
- Recognize that reviews can accelerate schedule by reducing other quality assurance activities
- Keep review team small
  - General guidelines: 3-7 participants
  - 3 is minimum for formal process to work
  - Below 3, too few perspectives besides author
  - Above 7, work may be slowed by process, scheduling
- Smaller groups for code, larger groups for other artifacts
  - Knowledge is spread around more stakeholders
  - Particular for requirements

Meetings: Review Guidelines - 2

- Find problems, but don’t try to solve them
  - Guideline: halt discussion after 1-3 minutes
- Limit meetings to 2 hours maximum
  - Attention span gets lost beyond this
- Require advance preparation
  - Provides much of the value of a (formal) review
Meetings: Checklists

Benefits
- Focus on likely sources of error
- Form quality standard that aids preparers
- Can bring up issues specific to a product

Should be short
- About seven items
- If more, group and do multiple passes

Focus
- Priority issues
- Issues unlikely to be found other ways
- Historical problems
- Issues specific to the document
- Issues specific to the author

Start with checklist from well-known source
- Refine based on experience

Process: Rework and Follow-up

Author addresses each item
- Ensure understanding of issue
  - Is it a defect or not? Is it a feature request or requirement change?

- Fixes defects and makes improvements
  - Uncorrected/unverified defects go into defect tracking system

- Deliverables
  - Corrected work product
  - Response to each issue and rationale for action

Moderator (or verifier) meets with author
- Check resolution of issues
- Examine corrected deliverable
People: Social Aspects of Reviews

- Reviews are challenging
  - Authors invest self-worth in product
  - Encourages you to avoid letting others find errors

- For Authors
  - Recognize value of feedback
  - Place value in making material (code, requirement statements…) easy to understand
  - Don’t take criticism of work personally

- For reviewers
  - Don’t show off how much better/smarter you are
  - Be sensitive to colleagues
    - Bad: “you didn’t initialize this variable”
    - Good: “I didn’t see where this variable was initialized”

Inspection of the DH SRS Document

1. Everybody inspects:
   1. functional requirements and non-functional requirements
      - Follow inspection checklist
      - Record individual findings in individual defect log
      - Should take < 40 min

2. Team meets for team inspection meeting
   - Moderator goes over sections of the document to elicit responses
   - Recorder logs findings into the inspection summary list
Inspection Teams and Roles

<table>
<thead>
<tr>
<th>Student</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angel Garcia</td>
<td>Moderator/Inspector</td>
</tr>
<tr>
<td>Vince Fonseca</td>
<td>Inspector/Recorder</td>
</tr>
<tr>
<td>Andres Olivas</td>
<td>Inspector</td>
</tr>
<tr>
<td>Raul Jimenez</td>
<td>Inspector</td>
</tr>
<tr>
<td>Mark Smith</td>
<td>Inspector</td>
</tr>
<tr>
<td>Jennifer Weand</td>
<td>Moderator/Inspector</td>
</tr>
<tr>
<td>Daniel Coronel</td>
<td>Inspector/Recorder</td>
</tr>
<tr>
<td>Upama Rahman</td>
<td>Inspector</td>
</tr>
<tr>
<td>Elvijs Ostrovskis</td>
<td>Inspector</td>
</tr>
<tr>
<td>Mark Eby</td>
<td>Inspector</td>
</tr>
</tbody>
</table>

Student  

| Jose Duarte  | Moderator/Inspector |
| Juan Soto    | Inspector/Recorder  |
| Matthew Melendez | Inspector   |
| Catalina Miranda | Inspector   |
| Chris Mckye  | Inspector           |
| Walter Copenhaver | Moderator/Inspector |
| Marco Lopez  | Inspector/Recorder  |
| Jorge Meza   | Inspector           |
| Jesus Tellez | Inspector           |
| Damien Valencia | Inspector   |

Deliverables

- Five Inspector Defect Logs
- One Inspection Summary Report