Software Inspections

Reading Assignment

  • Read pp. 387-393 Sections 3.1-3.3
**Manual Reviews**

- Manual static analysis methods
  - Reviews, walkthroughs, inspections, cleanroom
- Most can be applied at any step in the lifecycle
- Have been shown to improve reliability, but
  - often the first thing dropped when time is tight
  - labor intensive
  - often done informally, no data/history, not repeatable

**Different Kinds of Manual Reviews**

- **Reviews**
  - author or one reviewer leads a presentation of the artifact
  - review is driven by presentation, issues raised

- **Walkthroughs**
  - usually informal reviews of source code
  - step-by-step, line-by-line review
Different Kinds of Manual Reviews (continued)

- **Software inspections**
  - formal, multi-stage process
  - significant background & preparation
  - led by moderator
  - Many variations of this approach

- **Cleanroom**
  - Incremental development
  - formal review process
  - Plus, statistical-based testing

Software Inspections

- Developed by Michael Fagan in 1972 for IBM
- 3–5 participants
- 5 stage process with significant preparation
Inspections participants (4 to 6 people)

- MODERATOR - responsible for organizing, scheduling, distributing materials, and leading the session
- AUTHOR - responsible for explaining the product
- SCRIBE - responsible for recording bugs found
- PLANNER or DESIGNER - author from a previous step in the software lifecycle
- USER REPRESENTATIVE - to relate the product to what the user wants
- PEERS OF THE AUTHOR - perhaps more experienced, perhaps less
- APPRENTICE - an observer who is there mostly to learn

Inspection Process

- Planning
  - Done by author(s)
    - Prepare documents and an overview
      - explain content to the inspectors
  - Done by moderator
    - Gather materials and insure that they meet entry criteria
    - Arrange for participants
      - assign them roles
      - insure their training
    - Arrange meeting
Fagan Inspection Process (cont.)

• Preparation
  • Participants study material

• Inspection
  • Find/report faults (Do NOT discuss alternative solutions)

• Rework
  • Author fixes all faults

• Follow-Up
  • Team certifies faults fixed and no new faults introduced

Fagan Inspection-general guidelines

• Distribute material ahead of time
• Use a written checklist of what should be considered
  • e.g., functional testing guidelines
• Criticize product, not the author
**Experimental Results**

- software inspections have repeatedly been shown to be cost effective
- increases front-end costs
  - ~15% increase to pre-code cost
- decreases overall cost
IBM study

- Doubled number of lines of code produced per person
  - some of this due to inspection process
- Reduced faults by 2/3
- Found 60-90% of the faults
- Found faults close to when they were introduced
  - The sooner a fault is found the less costly it is to fix

Why are inspections effective?

- Knowing the product will be scrutinized causes developers to produce a better product
  - Hawthorne effect
- Having others scrutinize a product increases the probability that faults will be found
- Walkthroughs and reviews are not as formal as inspections, but appear to also be effective
  - Hard to get empirical results
**What are the deficiencies?**

- Tend to focus on error detection
  - what about other "ilities" -- maintainability, portability, etc.
- Not applied consistently/rigorously
  - inspection shows statistical improvement
- Human intensive and often makes ineffective use of human resources
  - e.g., skilled software engineer reviewing coding standards, spelling, etc.
  - Lucent study .5M LOCS added to 5M LOCS required ~1500 inspections, ~5 people/inspection
  - No automated support

**Experimental Evaluation**

- There have been many studies that have demonstrated the effectiveness of inspections
- Indirect effect--developers involved in inspections improve their skills by observing superior artifacts and skilled reviewers
- Recent studies trying to determine what aspects of inspections are effective
  - Provide insight into
    - Ways to improve the process
    - Ways to reduce the cost
Experimental evaluation of inspections


Experimental Design

- Lucent compiler project for 5ESS telephone switching system, 1994
  - 55K new lines; 10K reused lines
- Inspectors chosen from 11 professionals
  - At least 5 yrs. experience
  - Inspection training
- Modified inspection process and measured effect
  - Defects found
  - Interval: time from when artifact is ready to be reviewed until it is repaired
- 88 inspections overall
**Variants considered**

- **Team size**
  - Differences between teams of 1, 2, or 4 on # defects found

- **Inspection interval**
  - Calendar time to complete an inspection

- **Single or multi-session inspections**
  - N-fold --N teams doing N independent inspections
  - Multiple phased inspections focus on different concerns at each phase

- **Individual or group centered**
  - Is it necessary to actually have a meeting?

**Alternatives**

- N sessions(s), with M people(p), repairing defects (R) between sessions or not (N)
  - Ns x Mp {R|N}

  - E.g., Considered
  - 1sX4p
  - 2sX2pN
  - 2sX2pR
  - 1sX2p
  - 2sX1pN
  - 2sX1pR
Hypotheses

• Large teams ==> 
  • No increase in defects found 
  • Increase in interval 
• Multiple-session inspections ==> 
  • Increase in defects found 
  • Increase in interval 
• Correcting defects between sessions ==> 
  • Increase in defects found 
  • Increase in interval 
    • Terminated this process early since it was too costly

Results from the experiment

• Can use 2 person teams 
  • Can use a small team w/o jeopardizing the effectiveness 
  • 1sX1p < 1sX2p, but 1sX2p ≈ 1sX4p 
• Number of sessions did not impact effectiveness 
  • More sessions increase interval but not defects found 
  • Can use one session 
• Repairs between sessions did not significantly improve defect detection but did increase time interval

Use single sessions inspections with 2 person teams
Other results from the experiment

- Effort increases with the number of people, independent of the process
- Only 13% of reviewer issues were real defects
  - Meetings suppressed 26% of the superfluous issues
- Meetings lead to the detection of 30% of all the defects found
  - Others found by individuals before the meeting

Current trend: Xtreme programming advocates pairs of programmers working closely together

Concluding Remarks on Manual Reviews

- Some form of careful manual inspection seems to improve the quality of a s/w system and to improve productivity
  - Not clear if the benefits of cleanroom are from the inspection aspects of the process or other aspects or some combination
  - When deadlines are tight, it is very hard to commit the resources for such a labor-intensive task
- Some automated support could help to reduce the manual effort involved
  - Would this be effective or counter-productive?