SE-3002
SOFTWARE QUALITY ENGINEERING

RUBAB JAFFAR
RUBAB.JAFFAR@NU.EDU.PK

Part III-Other Quality Assurance Techniques
Software review, walk through
Lecture # 34, 35
6, 7 Dec

TODAY'S OUTLINE

- Presentations
- Gilb inspection
- Desk check
- Review
- Walkthrough
- Comparison of different QA techniques w.r.t
  - Defect perspective
  - Problem type
  - Interpretation
  - Level of difficulty

OTHER INSPECTIONS AND RELATED ACTIVITIES

- Variations to Fagan inspection have been proposed and used to effectively conduct inspection under different environments.
- Some of them are direct responses to some of the general findings of Fagan inspection described above.
- We organize these inspection techniques and processes along two dimensions:
  - size and scope of the inspection,
  - formality of the inspection.

INSPECTIONS OF REDUCED SCOPE OR TEAM SIZE

- Fagan inspection teams typically consist of four members.
- However, some software artifacts are small enough to be inspected by one or two inspectors.
- Similarly, such reduced-size inspection teams can be used to inspect software artifacts of limited size, scope, or complexity.
- This so-called two-person inspection is the simplification form of Fagan inspection, with an author-inspector pair, but following essentially the same process for Fagan inspection.
- This technique is cheaper and more suitable for smaller-scale programs, small increments of design and/or code in the incremental or iterative development, or other software artifacts of similarly smaller size.

INSPECTIONS OF REDUCED SCOPE OR TEAM SIZE

- Another implementation of two-person inspection is the reversible author- inspector pair, that is, the individuals in the pair complement their roles by inspecting each other's software artifacts.
- easier to manage technique because of the mutual benefit to both individuals instead of the asymmetric relation in Fagan inspection, where the author is the main beneficiary while the inspectors are performing "service" to others or to the company.
- The idea of two-person inspection is also found in the new development paradigm called agile development and extreme programming, where the so-called paired programming resembles the author-inspector pair.
- Informal inspection
- Reduce cost

INSPECTIONS OF ENLARGED SCOPE OR TEAM SIZE

- A common extension to Fagan inspection is based on the observation that during Fagan inspection meeting, people tend to linger on discovered defects and try to both find the causes for them and suggest fixes.
- These additional activities in the meeting would interfere with the main task of defect detection and confirmation in Fagan inspection and tend to prolong the meeting.
- On the other hand, these activities do add valuable information to the feedback that can be used to improve the overall inspection process and product quality.
- A solution to this problem is proposed in the Gilb inspection.

GILB INSPECTION

- In Gilb inspection an additional step, called "process brainstorming", is added right after the inspection meeting in Fagan inspection.
- The focus of this step is root cause analysis aimed at preventive actions and process improvement in the form of reduced defect injections for future development activities.
- There are several other special features to Gilb inspection, as characterized below:
- The input to the overall inspection process is the product document, rules, checklists, source documents, and kin documents. The emphasis is that any technical documentation, even diagrams, can be inspected.
- The output from the overall inspection process is the inspected (and corrected) input documents, change requests, and suggested process improvements.
- The inspection process forms a feedback loop, with the forward part resembling Fagan inspection but with the added step for process brainstorming, and the feedback part consisting of inspection statistics and adjustment to inspection strategies.
- Multiple inspection sessions are likely through this feedback loop

GILB INSPECTION

- The Gilb inner inspection steps are as follows (with Fagan inspection equivalent given inside parenthesis):
  - 1. planning (same),
  - 2. kickoff (overview),
  - 3. individual checking (preparation),
  - 4. logging meeting (inspection),
  - 5a. edit (rework),
  - 5b. process brainstorming (),
  - 6. edit audit (follow-up).
- 5a and 5b are carried out in parallel in Gilb inspection.
- The team size is typically about four to six.
- Checklists are extensively used, particularly for step 3, individual checking.

PHASED INSPECTION

- Another variation to the above is the phased inspection, where the overall inspection is divided into multiple phases with each focusing on a specific area or a specific class of problems.
- These problems not only include the defects (correctness problems), but also issues with portability, maintainability, etc.
- This inspection is typically supported by some form of checklist and related software tools.
- The dynamic team make-up reflects the different focus and skill requirements for individual phases.

INFORMAL DESK CHECKS, REVIEWS, AND WALKTHROUGHS

- **Desk check** typically refers to informal check or inspection of technical documents produced by oneself, which is not too different from proofreading one's own writings to catch and correct obvious mistakes.
- Advance software tools can detect things as mis-spelling, format, syntactical errors.
- Instead, desk checks should focus on logical and conceptual problems, to make effective use of the valuable time of software professionals.
- **Review** typically refers to informal check or inspection of technical

REVIEW

- Review typically refers to informal check or inspection of technical documents, but in this case, produced by someone else, either organized as individual effort, or as group effort in meetings, conference calls, etc.
- The focus of these reviews should be similar to desk checks, that is, on logical and conceptual problems.
- The differences in views, experience, and skill set are the primary reasons to use some reviews to complement desk checks.
- In most companies, the completion of a development phase or sub-phase and important project events or milestones are typically accompanied by a review, such as requirement review, design review, code review, test case review, etc.

WALKTHROUGH

- A special form of review is called walkthrough,
- a more organized review typically applied to software design and code.
- Meetings are usually used for these walkthroughs.
- The designer or the code owner usually leads the meeting, explaining the intentions and rationales for the design or the code, and the other reviewers (meeting participants) examine these design code for overall logical and environmental soundness and offer their feedback and suggestions.
- Defect detection is not the focus.
- Typically, these meetings require less time and preparation by the participants except for the owners.
- In practical applications, these informal checks, reviews, and walkthroughs can be used in combination with formal inspections.

EFFECTIVENESS COMPARISON

- Different QA alternatives can be compared by examining the specific perspectives of defect they are dealing with, what kind of problems they are good at addressing, their suitability to different defect levels and pervasiveness, and their ability to provide additional information for quality improvement.

EFFECTIVENESS COMPARISON : DEFECT PERSPECTIVE

- Among the different defect related perspectives and concepts, the QA alternatives can be compared by examining whether they are dealing with error sources, errors, faults, failures, or accidents.
- This examination can be broken down further into two parts:
- Detection or observation of specific problems from specific defect perspectives during the performance of specific QA activities.
- Types of follow-up actions that deal with the observed or detected problems in specific ways as examined from the defect perspectives.

EFFECTIVENESS COMPARISON: DEFECT PERSPECTIVE

QA Alternative	Defect Perspective	
	At Observation	At Follow-up (& Action)
testing	failures	fault removal
defect prevention	errors & error sources	reduced fault injection
inspection	faults	fault removal
formal verification	(absence of) faults	fault absence verified
fault tolerance	local failures	global failures avoided
failure containment	accidents	hazards resolution & damage reduction

EFFECTIVENESS COMPARISON: PROBLEM TYPES

- Different QA alternative might be effective for different types of problems, including dealing with different perspectives of defects, ranging from different errors and error sources, various types of faults, and failures of different severity and other characteristics.

Main problem types dealt with by different QA alternatives

QA Alternative	Problem Types
testing	dynamic failures & related faults
defect prevention	systematic errors or conceptual mistakes
inspection	static & localized faults
formal verification	logical faults, indirectly
fault tolerance	operational failures in small areas
failure containment	accidents and related hazards

EFFECTIVENESS COMPARISON: PROBLEM TYPES

- Defect prevention works to block some errors or to remove error sources to prevent the injection of related faults. Therefore, it is generally good at dealing with conceptual mistakes made by software designers and programmers.
- Once such conceptual mistakes can be identified as error sources, they can be effectively eliminated.
- One key difference between inspection and testing is the way faults are identified: inspection identifies them directly by examining the software artifact, while failures are observed during testing and related faults are identified later by utilizing the recorded execution information.
- This key difference leads to the different types of faults commonly detected using these two techniques.
- Inspection is usually good at detecting static and localized faults which are often related to some common conceptual mistakes, while testing is good at detecting dynamic faults involving multiple components in interactions.

THE REASONS OF DIFFERENCES BETWEEN THE TWO TYPES OF QA ALTERNATIVES:

- Inspection involves static examination while testing involves dynamic executions. Therefore, static problems are more likely to be found during inspection, while dynamic problems are more likely to be found during testing.
- It is hard for human inspectors to keep track of multiple components and complicated interactions over time, while the same task may not be such a difficult one for computers. Therefore, testing is generally better at detecting interaction problems involving multiple components.
- Human inspectors can focus on a small area and perform in-depth analysis, leading to effective detection of localized faults.

EFFECTIVENESS COMPARISON

- Formal verification deals with logical (or mathematical) correctness, and can be interpreted as extremely formalized inspection. Therefore, it shares some of the characteristics of inspection in dealing with static and logical problems.
- Problem identification is only a side-effect of failing to produce a correctness proof.
- Fault tolerance and failure containment are designed to work with dynamic operational problems that may lead to global failures or accidents.
- Fault tolerance techniques are good at isolating faults to only cause local failures but not global ones, while failure containment works to contain failures that may lead to accidents by dealing with hazards or reducing damage related to accidents.

EFFECTIVENESS COMPARISON: DEFECT LEVEL AND PERVASIVENESS

- Different QA techniques may be suitable for different defect levels or pervasiveness.

Defect levels where different QA alternatives are suitable

QA Alternative	Defect Level	
testing	low – medium	
defect prevention	low – high	(particularly pervasive problems)
inspection	medium – high	
formal verification	low	
fault tolerance	low	
failure containment	lowest	

EFFECTIVENESS COMPARISON: RESULT INTERPRETATION AND CONSTRUCTIVE INFORMATION

- Ease of result interpretation plays an important role in the application of specific QA techniques. A good understanding of the results is a precondition to follow-up actions.
- For example, both inspection and testing are aimed at defect removal. However, inspection results are much easier to interpret and can be used directly for defect removal. Testing results need to be analyzed by experienced software professionals to locate the faults that caused the failures observed during testing, and only then can these faults be removed.
- Result interpretation for formal verification, fault tolerance, and failure containment is harder than that for inspection and testing. A significant amount of effort is needed to analyze these results to support follow-up actions.
- For example, in a fault tolerant system using recovery blocks, repeated failures need to be dealt with off-line by analyzing the dynamic records. Much information related to unanticipated environment and usage not covered in the pre-planned testing activities may be included in these records.
- Similarly, failure containment results typically need additional analysis support.

EFFECTIVENESS COMPARISON: RESULT INTERPRETATION AND CONSTRUCTIVE INFORMATION

Ease of result interpretation for different QA alternatives and amount of constructive information/measurements

QA Alternative	Result Interpretation	Information/Measurement
testing	moderate	executions & failures
defect prevention	(intangible)	experience
inspection	easy	faults, already located
formal verification	hard	fault absence verified
fault tolerance	hard	(unanticipated) environments/usages
failure containment	hard	accident scenarios and hazards

COMPARISON SUMMARY

General comparison for different QA alternatives

QA Alternative	Applicability	Effectiveness	Cost
testing	code	occasional failures	medium
defect prevention	known causes	systematic problems	low
inspection	s/w artifacts	scattered faults	low – medium
formal verification	formal spec.	fault absence	high
fault tolerance	duplication	rare-cond. failures	high
failure containment	known hazards	rare-cond. accidents	highest

That is all