



SOFTWARE QUALITY ASSURANCE

Lecture 10

Instructor: Mr. Natash Ali Mian

Department of CS and IT
The University of Lahore



Switch off mobile phones during lectures, or put them into silent mode



TERM PAPER



- Finalize Group Members 26-Feb-2013
- Finalize Topic 12-Mar-2013
- Search Papers and Sort Selected (TODAY) 20-Mar-2013
- Go Through the Abstract and Introduction of Selected Papers 27-Mar-2013
- Submit a Summary and Comments on related papers 09-Apr-2013
- Present Your Work till Today 09-Apr-2013
- Submit Initial Draft 30-Apr-2013
- **Final Paper Submission 21-May-2013**
- Feedback on Final Submission + Plagiarism Report 28-May-2013
- Final Presentation 4-June-2013

Please note that Every Phase has Marks

CONTENTS

- Review
- Types of Reviews
- Roles and Responsibilities
- Introduction to Inspections



WHAT IS A REVIEW?

- A process or meeting during which a work product, or a set of work products, is presented to project personnel, managers, users, or other interested parties for comment or approval. Types include code review, design review, formal qualification review, requirements review, test readiness review
 - IEEE Std. 610.12-1990



OBJECTIVES OF REVIEWS - 1

- Identify required improvements in a product
- Assure that the deliverable is complete
- Assure that the deliverable is technically correct
- Measure the progress of the project



OBJECTIVES OF REVIEWS - 2

- Identify any defects early, thus resulting in cost and time savings
- Assure the quality of deliverable before the development process is allowed to continue
- Once a deliverable has been reviewed, revised as necessary, and approved, it can be safely used as a basis for further development



COLLEAGUES AS CRITICS

- There is no particular reason why your friend and colleague cannot also be your sternest critic
 - Jerry Weinberg



BENEFITS OF REVIEW

- A number of team members get an opportunity to provide their input
- Ownership of the work product is transferred from an individual to a group
- A (limited) training ground



KINDS OF REVIEWS

- Business reviews
- Technical reviews
- Management reviews
- Walk-throughs
- Inspections



OBJECTIVES OF BUSINESS REVIEWS

- The deliverable is complete
- The deliverable provides the information required for the next phase
- The deliverable is correct
- There is adherence to the procedures and policies



OBJECTIVES OF TECHNICAL REVIEWS - 1

- Point out needed improvements in the product of a single person or a team
- Confirm those parts of a product in which improvement is either not desired or not needed
- Achieve technical work or more uniform, or at least more predictable, quality than can be achieved without reviews, in order to make technical work more management



OBJECTIVES OF TECHNICAL REVIEWS - 2

- Software reviews are a “filter” for software engineering process
- Reviews are applied at several points during software development and serve to uncover errors and defects that can then be removed
- Software reviews “purify” the software engineering activities



OBJECTIVES OF TECHNICAL REVIEWS - 3

- Technical work needs reviewing for the same reason that pencils need erasers: To err is human
- Another reason we need technical reviews is that although people are good at catching some of their own errors, large classes of errors escape the originator more easily than they escape anyone else



OBJECTIVES OF TECHNICAL REVIEWS - 4

- They also ensure that any changes to the software are implemented according to pre-defined procedures and standards



WHAT TECHNICAL REVIEWS ARE NOT!

- A project budget summary
- A scheduling assessment
- An overall progress report
- A mechanism for reprisal or political intrigue!!



OBJECTIVES OF MANAGEMENT REVIEWS - 1

- Validate from a management perspective that the project is making progress according to the project plan
- Ensure a deliverable is ready for management approval
- Resolve issues that require management's attention



OBJECTIVES OF MANAGEMENT REVIEWS - 2

- Identify if the project needs a change of direction
- Control the project through adequate allocation of resources



REVIEW ROLES

- Facilitator
- Author
- Recorder
- Reviewer
- Observer





RESPONSIBILITIES OF ROLES

RESPONSIBILITIES OF FACILITATOR

- Responsible for providing the background of the work and assigning roles to attendees
- Encourages all attendees to participate
- Keeps the meeting focused and moving
- Responsible for gaining consensus on problems



RESPONSIBILITIES OF AUTHOR - 1

- Responsible for the readiness and distribution of material to be reviewed
- During the meeting, the author paraphrases the document a section at a time
- Responsible for
 - scheduling the review
 - selecting the review participants
 - determining if the entry criteria for the review are met



RESPONSIBILITIES OF AUTHOR - 2

- providing information about the product during all stages
- clarifying any unclear issues
- correcting any problems identified
- providing dates for rework and resolution



RESPONSIBILITIES OF RECORDER

- Collects and records each defect uncovered during the review meeting
- Develops an issues list and identifies whose responsibility it is to resolve each issue
- Records meeting decisions on issues; prepares the minutes; and publishes the minutes, and continually tracks the action items



RESPONSIBILITIES OF REVIEWER

- Spends time prior to the meeting reviewing information
- Makes notes of defects and becomes familiar with the product to be reviewed
- Identifies strengths of the product
- Verifies that the rework is done
- Insists upon clarifying any issues that are not clear



RESPONSIBILITIES OF OBSERVER

- A new member to the project team, who learns the product and observes the review techniques



REVIEW GUIDELINES

- Preparation
- Discussions
- Respect
- Agenda
- Review Records
- Resources
- Attendees



REVIEW FREQUENCY

- At the beginning/end of the requirements phase
- At the beginning/end of the design phase
- At the beginning/end of the code phase
- At the beginning/end of the test phase
- Approval of the test plan



REVIEW PLANNING - 1

- Distribute review package one week in advance
 - Document to be reviewed
 - Review agenda
 - Identification of the individual who will manage the agenda and schedule
 - Exit and entrance criteria for the review
 - Objective of the review



REVIEW PLANNING - 2

- Names of attendees, their roles and responsibilities
- Review location
- Date and time of review
- List of classifications that will be used for defects discovered (defect type, defect origin, and defect severity)
- Procedures for handling issues raised during the review and escalation phase



REVIEW MEETING - 1

- Facilitator begins the meeting with an introduction of agenda, people, and description of their roles
- Author of the document proceeds to explain the materials, while reviewers raise issues based on advance preparation



REVIEW MEETING - 2

- When valid problems, issues, or defects are discovered, they are classified according to their origin or severity and then recorded
- These are accompanied with the names of individuals who are responsible for resolution and the time frame during which the item will be resolved
- Related recommendations are also recorded



GUIDELINES FOR REVIEWERS

- Be prepared - evaluate product before the review meeting
- Review the product, not the producer
- Keep your tone mild, ask questions instead of making accusations
- Stick to the review agenda
- Raise issues, don't resolve them
- Avoid discussions of style - stick to technical correctness



DECISIONS AT THE END OF A REVIEW MEETING

- All attendees must decide whether to
 - Accept the product without further modification
 - Reject the product due to severe errors
 - Accept the product provisionally
 - Hold a follow-up review session



REVIEW REPORT - 1

- Published by the recorder, with approval from all attendees, after a week of the review meeting
- Review report consists of
 - Elements reviewed
 - Names of individuals who participated in the review
 - Specific inputs to the review



REVIEW REPORT - 2

- List of unresolved items
- List of issues that need to be escalated to management
- Action items/ownership/status
- Suggested recommendations



REWORK

- It is the responsibility of project manager to ensure that all defects identified in the review are fixed and retested



FOLLOW-UP

- During the follow-up, that all discrepancies identified are resolved and the exit criteria for the review have been met
- Document lessons learned during the final report also



Inspections



INSPECTIONS - 1

- An inspection is a rigorous team review of a work product by peers of the producer of the work product
- The size of the team will vary with the characteristics of the work product being inspected; e.g., size, type



INSPECTIONS - 2

- The primary purpose is to find defects, recording as a basis for analysis on the current project and for historical reference and for improvement for future projects, analyzing them, and initiating rework to correct the defects
- Direct fault detection and removal



INSPECTIONS - 3

- Inspections are most effective when performed immediately after the work product is complete, but they can be held any time the work product is deemed ready for inspection



INSPECTIONS - 4

- Inspections are critical reading and analysis of software code or other software artifacts, such as designs, product specifications, test plans, etc
- Inspections are typically conducted by multiple human inspectors, through some coordination process. Multiple inspection phases or sessions may be used



INSPECTIONS - 5

- Faults are detected directly in inspection by human inspectors, either during their individual inspections or various types of group sessions
- Identified faults need to be removed as a result of the inspection process, and their removal also needs to be verified



INSPECTIONS - 6

- The inspection processes vary, but typically include some planning and follow-up activities in addition to the core inspection activity
- Developed by Michael Fagan at IBM and were first reported in public domain in 1976



INSPECTIONS - 7

- Inspections remove software defects at reduced cost
- Inspections enable us to remove defects early in the software life cycle, and it always cheaper to remove defects earlier in than later in the software life cycle



- We know that defects are injected in every software life cycle activity
- We remove some of these defects in testing activities after code is completed
- We also know that all defects are not removed at shipment time, and these are known as latent defects



- We want to eliminate or at least minimize latent defects in the shipped software product
- It is expensive to find and remove defects in the testing phase, and even more expensive after shipment of the software
- We can use inspections to reduce these costs and improve the timelines also





**HOW DEFECT REMOVAL IS
CHEAPER FOR INSPECTIONS AS
COMPARED TO SOFTWARE
TESTING**

- During testing, defects are found, then the programmers are notified of their presence, who will recreate the defects under the similar circumstances, fix them, re-test the software and re-integrate the software module, which were affected



- While in inspections, the inspection process is executed in the same life cycle activity, and substantial amount of rework is avoided
- This results in the reduction of costs



- If and when defects are detected after the shipment of the software, then these costs are even higher
- Many times, original development team is disbanded after the completion of the project and new staff is looking after the maintenance activity
- These people are usually not fully aware about the project
- This can result in unplanned expenses for the software development company



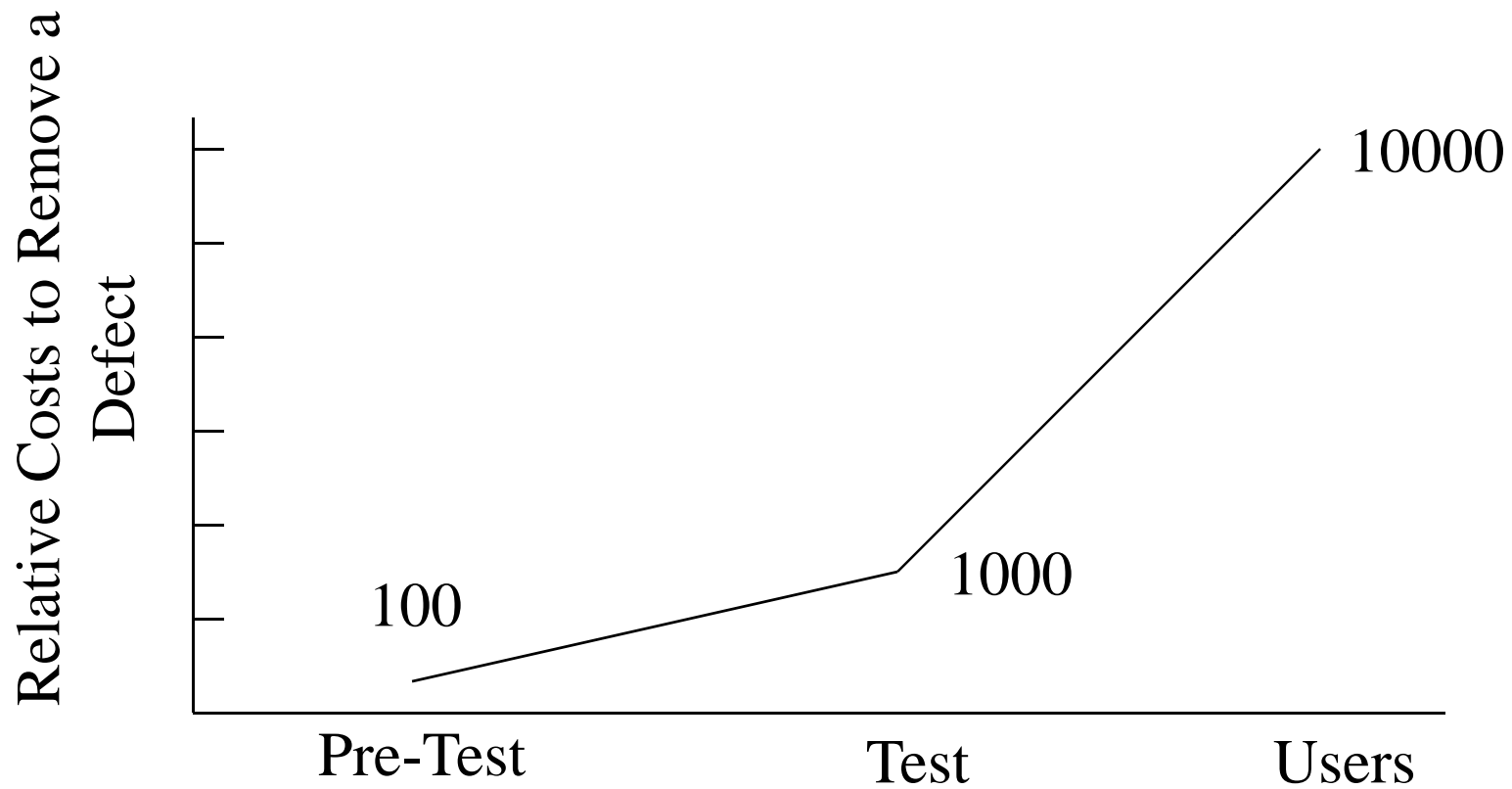
- On the other hand, if an effective software inspections process is in place, fewer defects enter the testing activity and the productivity of tests improve
- The costs of tests are lower and the time to complete tests is reduced



- Several studies have confirmed the reduction in project costs when defects were removed earlier



DEFECT COST RELATIONSHIP



Major Defect Removal Area

- It is interesting to note that this relationship has remain consistent in the last three decades – since the earliest studies when inspections were being first reported
- In addition to the costs on project, there are additional costs to the customer for downtime, lost opportunity, etc., when defects are detected in maintenance




- Let's look at the published data from different studies of companies in which comparison of inspection costs and testing costs have been made
- These were independent studies, and so they use different units to report their results
- However, the pattern repeats that the cost of inspections is much lower than that of software testing



REPORTED COST RELATIONSHIP - 1

Company	Cost in Inspections	Cost in Test	Cost With Customer Discovery
IBM	\$48/defect	\$61-\$1030 / defect	\$1770 / defect
AT&T	1 unit	20 units	--
ICL	1.2-1.6 hours/defect	8.47 hours/defect	--




REPORTED COST RELATIONSHIP - 2

Company	Cost in Inspections	Cost in Test	Cost With Customer Discovery
AT&T	1.4 hours	8.5 hours	--
JPL	\$105/defect	\$1700/defect	--
IBM	1 unit	9 times more	117 times more

REPORTED COST RELATIONSHIP - 3

Company	Cost in Inspections	Cost in Test	Cost With Customer Discovery
Shell	1 unit	30 units	--
Thorn EMI	1 unit	6.8-26 units	96 units
Applicon, Inc.	1 hour	--	30 hours
Infosys	1 unit	3 – 6 units	--



- These studies clearly report data from different companies that it is cheaper to detect and remove data using software inspections as compared to software testing
- There is evidence in the literature that inspection offer significant return on investment even in their initial use

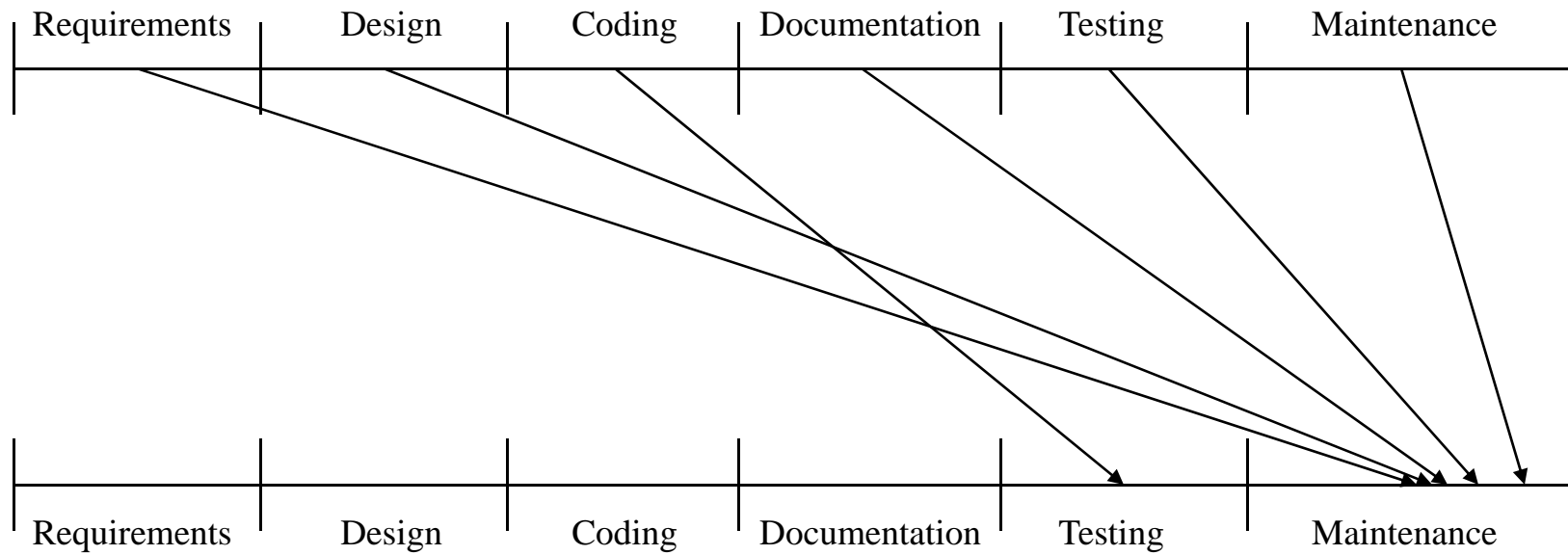


- Let' now look at inspections from another point of view
- Relating defect origin points and defect discovery
- In a project with no software inspections, defects are typically injected in the earlier activities and detected in later stages
- As a result, we get a chaos zone



DEFECT ORIGINS AND DISCOVERY POINTS WITHOUT USAGE OF FORMAL INSPECTIONS

Defect Origins



Defect Discovery

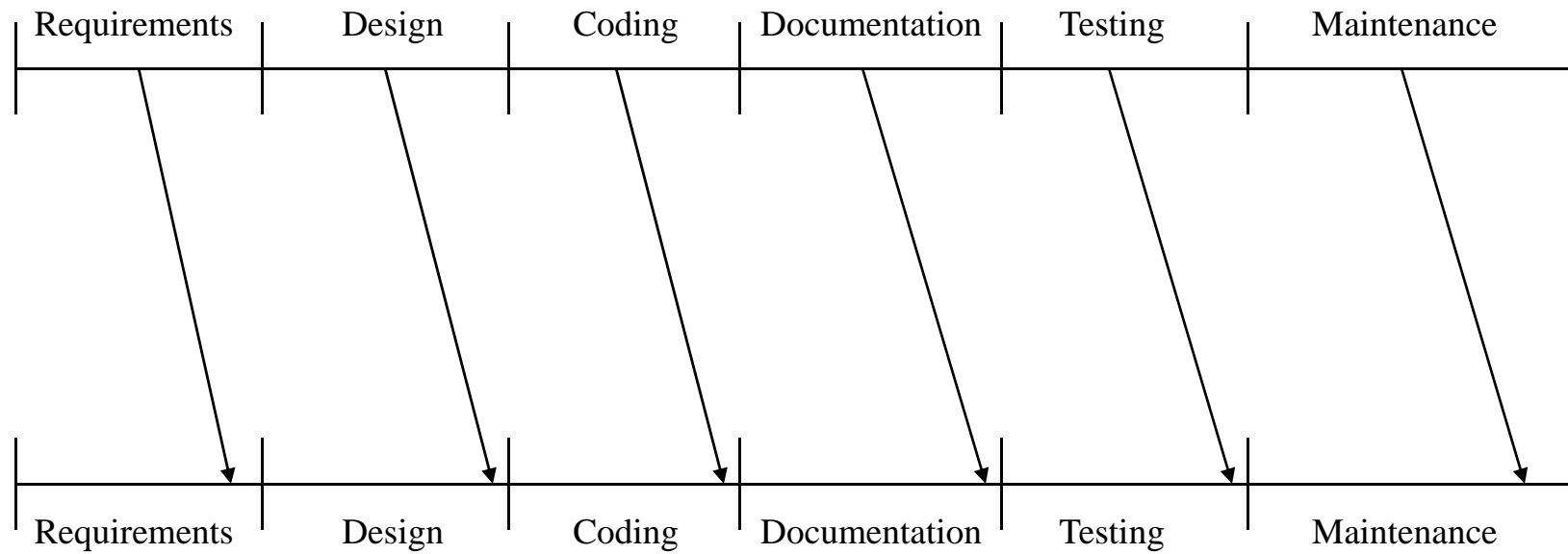
Chaos Zone

- This situation is a mess
- If only we were able to detect defects in the same life cycle activity, we can eliminate the chaos zone, and bring some sanity back to the project team and project management
- If we introduce software inspections, we can do that



DEFECT ORIGINS AND DISCOVERY POINTS WITH USAGE OF FORMAL INSPECTIONS

Defect Origins



Defect Discovery

- Here you can see that the chaos zone has been eliminated
- This is achieved by performing inspections on work products before leaving that life cycle activity, and as a large number of requirements defects will be detected and removed during the requirements activity, design and coding defects will be detected and removed during those activities, and so on



WHY ISN'T EVERYONE USING INSPECTIONS?

- Now we are convinced that inspections have a clear value independent of any model or standard for software development, so why isn't everyone using it?



REASONS FOR NOT USING INSPECTIONS - 1

- There is resistance to Inspections because people view them as if they are not easy to do well
- Management often views Inspections as an added cost, when in fact Inspections will reduce cost during a project
- Development of new tools and environments



REASONS FOR NOT USING INSPECTIONS - 2

- Inspections are not the most enjoyable engineering task compared to designing and coding
- Inspections are labor intensive and low-tech
- Programmers/designers are possessive about the artifacts they create



INSPECTION PRECONDITIONS

- Clear and visible management support
- Defined policy
- Good training for all
- Effective procedures
- Proper planning
- Adequate resources



REFERENCES

- Inroads to Software Quality by Alka Jarvis and Vern Crandall, PH 1997 (Ch. 7)
- Software Engineering: A Practitioner's Approach by Roger S. Pressman (Ch. 8)



Thanks!

