Course Structure

1. Software Quality Assurance
2. Testing Fundamentals
3. Code-based Techniques
4. Specification-based Techniques
5. **Inspection Technique**
6. Test Tools
7. Measuring Software Quality
8. TDD

Inspection

Objectives

Other types of reviews

Case studies

Benefits & Cost

Team

Process

Results

"Given enough eyeballs, we can find most bugs"

Learning Objectives

- Identify the **different types of review / inspection**
- Discuss the **costs and benefits** of inspections
- Understand and follow the various stages of the **inspection process**
- **Conduct an inspection**
- Organize and run effective inspections
- Use the **reports** associated with the inspection process

Inspection

- A static analysis technique
- A form of **Peer Review**
- Based on AT&T Quality Control Methods, 1920's
- 1972 Michael Fagan implemented on software
  - ➔ Doubled productivity at IBM Federal Systems
- Highly *structured* meeting, forum for *independent* evaluation
- *Early*, *in-process* validation technique
- Each participant has specific roles
- Formal ***error seeking*** technique
- Formal reporting
- Formal follow-up on errors found

Inspection

A systematic **peer examination** that:

- Verifies that the software product satisfies
  - its specifications
  - specified quality attributes
- Conforms to applicable regulations, standards, guidelines, specifications, plans and procedures
- Identifies deviations from standards (coding and documentation standards) and specifications

Key Objectives

- ⌘ **Detect and correct defects** before they leak through subsequent development phases and into the field.
- Identify defects in *early* stages of life cycle
- Identify defects *cheaply* and *inexpensively*
- *Reduce* development and maintenance *costs*
- ⌘ *Shorten* development *cycle time*

The Need for Review/Inspection

"Engineers today, like Galileo three and a half centuries ago, are not superhuman. They make mistakes in their assumptions, in their calculations, in their conclusions. That they make mistakes is forgivable; that they catch them is imperative.

Thus it is the essence of modern engineering not only to be able to check one's own work, but also to have one's work checked and to be able to check the work of others."

Henry Petroski, 1985

What isn't it?

Inspection

- Not for *design alternative* evaluation
- Not used as a solution finding
- Not for reviewing programming style
- Not for software quality assurance group
- Not for *management* participation
- Not for *individual performance* evaluation

Inspections Can be Used On ...

- ⌘ Proposals
- ⌘ Requirements
- ⌘ Design
- ⌘ Code
- ⌘ Documentation
- ⌘ Engineering Change Requests
- ⌘ Statement of Work
- ⌘ Development Plan
- ⌘ Management Plan
- ⌘ Test Plan
- ⌘ Test Procedures
- ⌘ Quality Plan
- ⌘ White Papers
- ⌘ Newsletters

Other form of Reviews: Walkthrough

Purpose:

- Find defects
- Improve the software product
- **Consider alternative implementations**
- Evaluate conformance to standards and specifications
- **Exchange of techniques and style variations**
- **Educate an audience on the product (Training of the participants).**

The author leads and explains his code to a ***peer group*** in an informal meeting.

No preparation is needed.

Other form of Reviews: Audit

Purpose:

- **Independent evaluation of conformance** of software products and processes to applicable regulations, standards, guidelines, specifications, plans and procedures.

A third party (maybe from outside the organization) conducts an evaluation and examination of product or process.

Formal report to management.

Other: Management Reviews

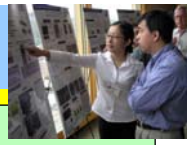
Purpose:

- **monitor progress**
- determine the status of plans and schedules
- confirm requirements and their system allocation
- evaluate the effectiveness of management approaches used to achieve fitness of purpose.

Management reviews

- carried out by, or on behalf of, the management.
- identify consistency with and deviations from plans, or adequacies and inadequacies of management procedures.
- support decisions about corrective actions, or changes to the scope of the project.

Other: Technical Reviews



Purpose:

evaluate a software product by a team of qualified personnel to determine its suitability for its intended use and identify discrepancies from specifications and standards.

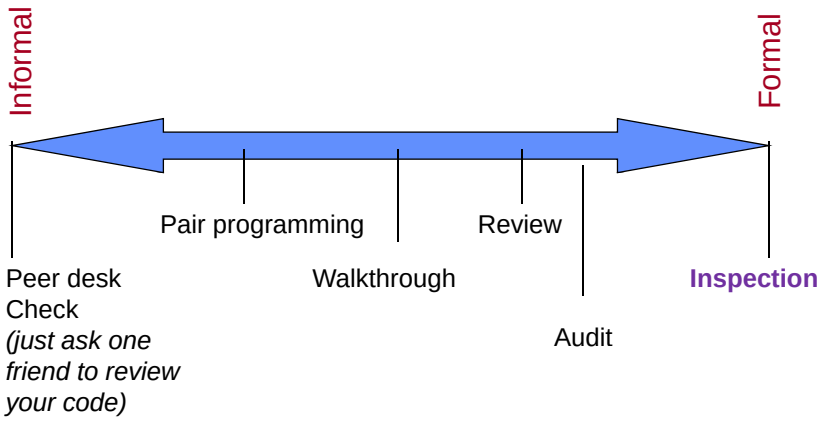
Technical review provides management with evidence to confirm whether:

- The software product conforms to its specifications
- The software product adheres to regulations, standards, guidelines, specifications, plans, and procedures applicable to the project.
- Changes to the software product are properly implemented and affect only those system areas identified by the change specification.

Comparison

STATIC REVIEW TECHNIQUE					
Feature \ Review	Inspection	Desk Check	Walkthrough	Audit	Phase Review
Purpose	Defect Identification	Defect Identification	Design Evaluation	Process Verification	Progress Evaluation
Timing	Very Early	Early	Early	Late	Very Late
When	Product Completion	Product Completion	Product Completion	Phase Completion	Phase Completion
People	4	1	5-15	5	10-200
Cost	Very Low	Low	High	Medium	Very High
Who	Engineers	Engineers	Project Manager	Quality Assurance	Customer
Independent	Very High	Low	Very Low	High	High
Rework	Mandatory	Not Mandatory	Not Mandatory	Not Mandatory	Not Mandatory
Pace	Slow	High	High	High	Very High
Efficiency	Very High	Average	Low	Low	Very Low
Measurable	Very High	High	Very Low	Low	Low
Metric	Defect	Defect	N/A	Deviation	Change Request

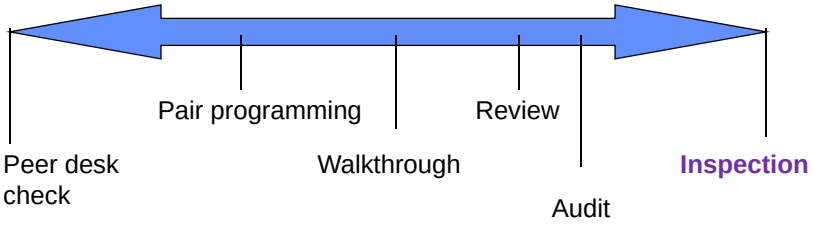
Formality of Different Types of Peer Reviews



Cost and Risk of Different Types of Peer Reviews

Lower ROI
Lower Cost
For Lower Risk software

Higher ROI
Higher Cost
For Higher Risk software



Who Has Used Inspection?

- IBM
- AT&T
- NASA's Space Shuttle program
- Jet Propulsion Laboratory
- American Express
- Shell Research
- Hewlett-Packard
- Standard Bank of South Africa
- Nortel/ BNR
- \+ many more companies

Bull HN Inform. Systems

System
- Operating system
- 11 million lines of code
- 600,000 lines of code added annually
- "C" programming language

Experience
- 7,413 inspections
- 11,557 "major" defects identified
- **98.7% defect removal efficiency achieved**
- 667,170 inspection data points (*in 3 years*)

Bell Northern Research

System
- Embedded, real-time digital switching systems
- 15 million LOC
- 312,500 LOC added quarterly
- Modern, high-level programming languages

Experience
- 2,778 inspections
- 240,000 defects identified
- **80% defect removal efficiency achieved**
- 250,020 inspection data points (*in 2 years*)

IBM AS/400

System
- Operating system
- 7.1 million LOC
- 2 million LOC added annually
- PL/1, Jovial, and RPG programming languages

Experience
- 7,889 inspections
- 681,600 defects identified
- **70% defect removal efficiency achieved**
- 710,010 inspection data points (*in 3.5 years*)

Nortel Network

Inspection rate	Defect Found
150 line/hr	50/KLOC
450 line/hr	15/KLOC

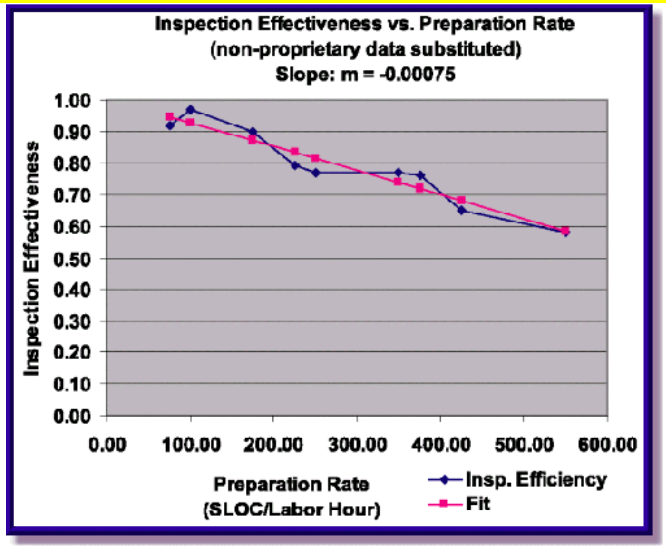
Inspection rate = size/ inspection time

- 1 staff hour of inspection time for each defect found.
- 2-4 staff hours of testing time for each defect found.

Key lessons:

1. Do not inspect at a fast rate. Slow down and we can find more defects.
2. Inspection is more efficient than testing

Effectiveness vs Preparation Rate



AT&T

System
- Embedded, real-time systems
- 111,600 lines of code
- 9,300 lines of code per project average
- "C" programming language

Experience
- 324 inspections
- 4,860 defects identified
- **70% defect removal efficiency achieved**
- 29,160 inspection data points (*in 7 years*)

Lockheed Martin

System
- Embedded, real-time system
- 2 million LOC
- 200,000 LOC added per year
- "C" programming language

Experience
- 23 inspections
- 324 defects identified
- **67% defect removal efficiency achieved**
- 2,070 inspection data points (*in 1 year*)

IBM Space Shuttle

System
- Man-rated spacecraft avionics system
- 500,000 LOC
- 25,467 LOC added/maintained per year
- HAL-S programming language (custom)

Experience
- 1,061 inspections
- 36,672 defects identified
- **90% defect removal efficiency achieved**
- 95,490 inspection data points (*in 15 years*)

HP

	Efficiency (defect found/hr)
Regular use	0.21
Black box testing	0.28
White box testing	0.32
Inspection	1.06

Inspection is more efficient than testing !!

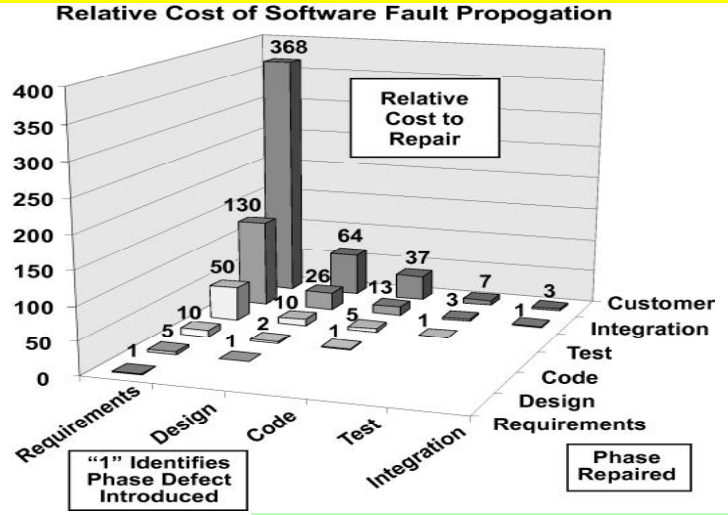
Relative Cost of Defect Correction

	Found at Stage						
Occurred at stage	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1. Contract	**1**	**1.3**	**2.4**	**3.3**	**6.8**	**26**	**96**
2. Requirements		**1**	**1.8**	**2.4**	**5.1**	**19**	**72**
3. Preliminary design			**1**	**1.3**	**2.8**	**11**	**39**
4. Detailed design				**1**	**2.1**	**8**	**30**
5. Unit test					**1**	**3.8**	**14**
6. System/integ test						**1**	**3.7**
7. Operation/usage							**1**

Ref.: British Standard

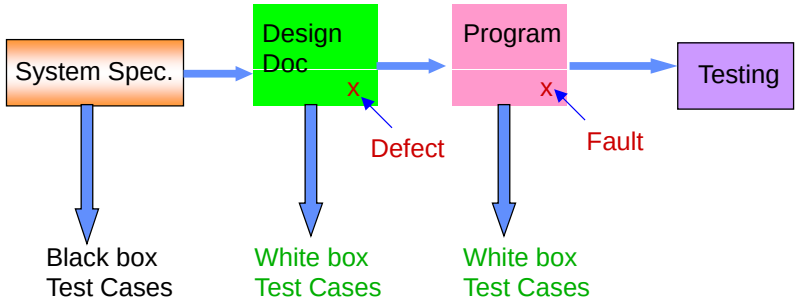
Early defect detection save $

Statistics for embedded systems



Early defect detection save $

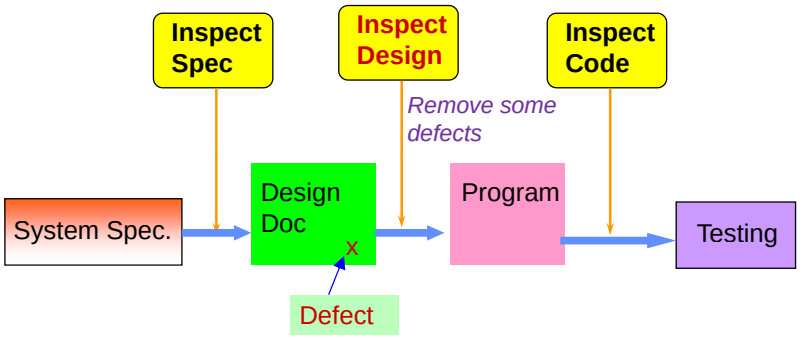
Inspection can Detect Design Error



- If the design has a defect, then the program will include the corresponding fault.
- When verifying the program, the design will be used as the ***oracle***, thus, cannot detect this defect!!

Detecting the Design Error

Add a step to inspect the design.



In fact, we should have an inspection step after every stage!

Return on Investment

Average return on Investment: **133:1**

	ROI
DACS	72:1
AT&T	234:1
Rico	160:1
BNR	114:1
Gilb	113:1
HP	104:1

Cycle Time Reduction

Average cycle time reduction: **5.5x**

	CT reduction
DACS	1.6x
Fagan	6.7x
AT&T	8.4x
BNR	5.2x
Gilb	5.1x
HP	4.8x

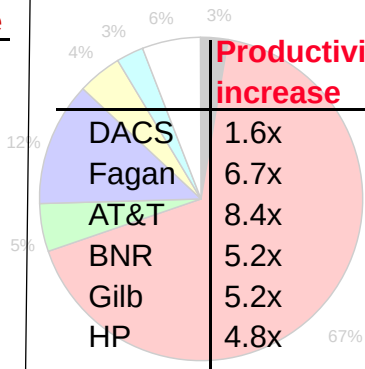
Quality Increase

Average quality increase: **16.4x**

	Quality increase
Bull HN	76.9x
Aetna	5.6x
IBM	14.3x
BNR	5.0x
AT&T	3.4x
Fagan	6.7x

Productivity Increase

Average productivity increase: **6x**



	Productivity increase
DACS	1.6x
Fagan	6.7x
AT&T	8.4x
BNR	5.2x
Gilb	5.2x
HP	4.8x

Defect Removal Efficiency (DRE)

	Defects detected
Immediately after proper training	50%
Within 12-18 months	60-90%

	DRE
Bull HN	98.7%
Aetna	82%
IBM	83-93%
BNR	80%
AT&T	70-92%
Fagan	85%

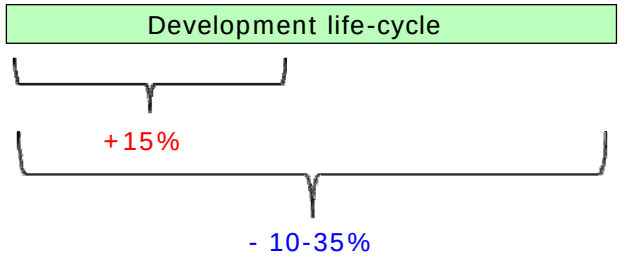
Average defect removal efficiency of 82.2%

Other BENEFITS of Inspection

- ☺ Reduction in user-reported defects
- ☺ Increased customer satisfaction
- ☺ Increased development productivity, evidenced by more function shipped in a given time, or reduced development cycle time
- ☺ Improvements in meeting committed schedules
- ☺ Rapid cross-training of developers and maintainers on new products
- ☺ Team building
- ☺ Raise people's awareness of quality issues, accountability, and effectiveness

Inspection Costs

- ⌘ Investment of 15% of the total development cost early in the process
- ⌘ Reduced spending later in the testing phases
- ⌘ Overall reduction: 10-35%



Inspection Costs

On-going cost:

- $ for minor improvements,
- $ for maintaining a statistical database,
- $ for training inspectors

Cost Factors

- ⌘ administering
- ⌘ training
- ⌘ developing procedures
- ⌘ preparing requirements, design, code, and test documents for review
- ⌘ reviewing the documents
- ⌘ conducting inspection
- ⌘ recording data
- ⌘ compiling metrics

Why Inspections are Efficient? (1)

In testing

- we start with a failure
- then we must find the defect
- next, we devise a fix
- finally, we implement and test the fix

Output is wrong !

Where is the fault?

With inspections

- we see the defect
- then we devise a fix
- finally, we implement and review the fix

Why Inspections are Efficient? (2)

In **testing**, we must

- detect that the system result was unusual
- figure out what the program was doing
- find where it was in our program
- figure out what defect could cause such behavior

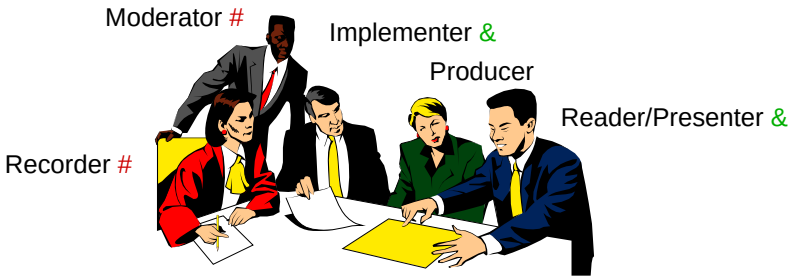
We are searching for the unplanned and unexpected.

This can take a large effort!

With **inspections**

- we follow our own logic
- when we find a defect, we know exactly where we are
- we know what the program should do and did not do
- we thus know why it is a defect
- we are in a better position to devise a fix

Inspection Team



may be the same person
& may be the same person

Other inspector: Tester

- *No User*
- *No Management*

Moderator

- Often from outside group
- Must be competent at level of the document being inspected
- Skilled in
  - coaching
  - sensitivity
  - act as 'invisible leader'
- Handles administrative details (arrange and chair the meeting)
- Ensure that the meeting is successful, follow up on action taken
- Not a supervisor

Multiple Roles of Moderator

- ⌘ Run the inspection meeting
- ⌘ Induce optimum performance from the inspection team
- ⌘ Ensure adherence to the inspection process

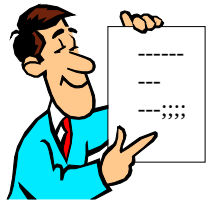
- ⌘ If a moderator is unprepared, he may leave out crucial steps of the process, misapply process rules, or do the inspection with the wrong team.
- ⌘ This causes wide variation in inspection results

Recorder

- Takes 'minutes'
- Records errors found, preparation time, etc.
- Provides written report
- Frequently performed by moderator

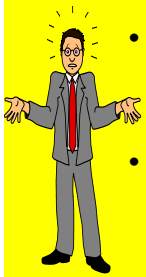
Reader

- 'Lead Inspector'
- May be the primary implementer
- Paraphrases product, present the product
- Very familiar with material



Producer

- Author of the "document" to be inspected
- Answers questions about product
- Does not justify approach
- Does not seek solutions to problems
- At the end of meeting, provide comment on the inspection
- Performs the follow-up actions.



Other Members

Implementer

- Takes product through its next phase
- May be designated to be Reader

Tester

- Obtain glass box view
- Look for subtle defects

Who are Producer & Implementer?

When inspecting the design doc.

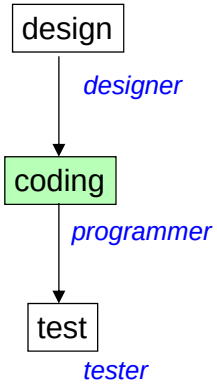
- Producer is the designer
- Implementer is the programmer

When inspecting the source code

- Producer is the programmer
- Implementer is the tester

When inspecting the test plan

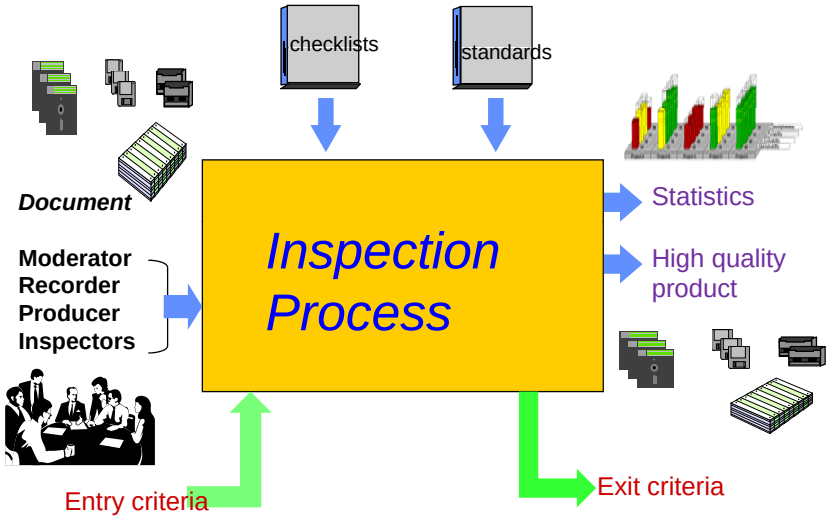
- Producer is the tester
- Implementer is the maintainer



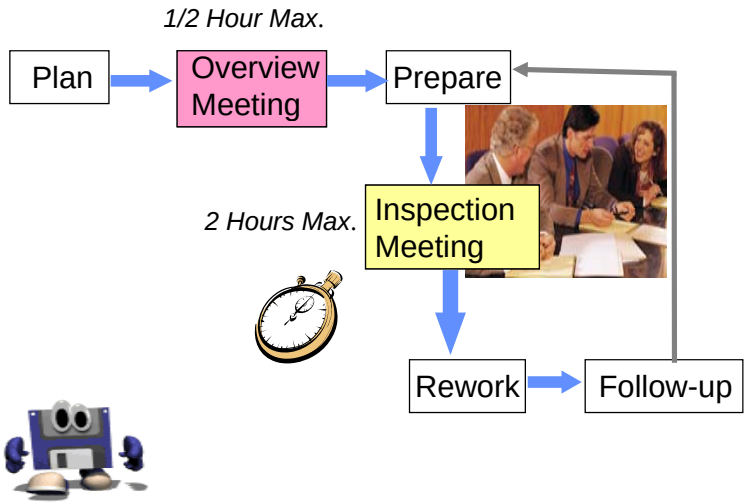
All Inspectors

- Identify and report defects
- Make process improvement suggestions
- Learn, teach by example
- Work as a team
- Help the producer
- Do not criticize product, or suggest changes (**just look for defects**)

How to conduct an inspection



Inspection Process



Planning

- Ensure the right people are involved
- Work out logistics, prepare materials and invite team members
- Distribute agenda and product for inspection
- Adequate preparation time is ideally 3 days

PLANNING	
Purpose	✓Plan Inspection
Input	✓Product (Draft)
Activity	✓**Moderator Verifies Entry Criteria** ✓**Moderator Selects Participants** ✓**Moderator Schedules Inspection** ✓**Moderator Prepares Notices**
Output	✓**Product (Entry Criteria Verified)** ✓**Inspection Notice**

Overview

OVERVIEW	
Purpose	✓Introduce and Describe Product
Input	✓Product (Entry Criteria Verified)
Activity	✓Moderate Facilities Overview ✓Producer Introduces Product ✓Moderator Assigns Roles
Output	✓Product Description

Preparation

PREPARATION	
Purpose	✓Prepare for Inspection
Input	✓Product (Entry Criteria Verified)
Activity	✓Inspectors Analyze Specifications ✓Inspectors Analyze Checklists ✓Inspectors Analyze Defect History ✓Inspectors Note Potential Defects
Output	✓Defect List (Preliminary)

Preparation phase

Moderator
- Prepares agenda
- Defines approach
- Defines expectations
- Arranges meeting

Reader
- Prepares product for presentation
- Works with the recorder to define the approach to documenting deficiencies
- May prepare meeting documents with the Producer

Documents

Inspection team
- Previews documents
- May review in writing if unable to attend

Preparation rate=200 line/hr

Inspection Meeting

- ⌘ Present product and follow agenda
- ⌘ Get acceptance, define issues and prioritize
- ⌘ Document findings

INSPECTION	
Purpose	✓Identify Defects
Input	✓Product (Entry Criteria Verified) ✓Defect List (Preliminary)
Activity	✓Moderator Facilities Inspection ✓Reader Paraphrases Product ✓Producer Answers Questions ✓Inspectors Identify Defects ✓Recorder Transcribes Defects ✓Moderator Reviews Defect List ✓Moderator Summarizes Defects
Output	✓Defect List (Final) ✓Defect Summary

At the Meeting

Moderator
- ➔ States objective
- ➔ Confirms time limit
- ➔ Keep defect logging rate high
- ➔ Be diplomatic
- ➔ Remind team to avoid extra talk
- ➔ Remind team to look for defects
- ➔ Ask junior team members to start first
- ➔ contributes defect last

Inspection team
- General comments
- Major concerns

Reader
- Walks through product

Reader
- Brief overview of product
- Purpose of product
- Objectives/constraints
- Who will use the product

Follow-up action list
Project issues
List of errors
Annotated product

Entry Criteria
Checklist

Before the inspection meeting:
1. Has the preceding life cycle activity been concluded?
2. Are inspection members in place and briefed?
3. Have all members received all the review materials?
4. How many hours of preparation did each member perform?
5. Are there any changes to the baseline?

Exit Criteria
Checklist

At the end of the inspection meeting:
1. Have all product elements been inspected?
2. Have all checklists been processed?
3. Have the inspection results been recorded?
4. Have data/metrics been collected?
5. Has the producer commented on the identified issues and defects?

Rework

REWORK	
Purpose	✓Correct Defects
Input	✓Product (Entry Criteria Verified) ✓Defect List (Final)
Activity	✓**Producer** Reviews Defect List ✓**Producer** Corrects Defects
Output	✓Product (Reworked)

Follow-up & Re-inspection

- ⌘ Establish target dates for resolution of issues
- ⌘ Make changes as dictated by the inspection
- ⌘ Major re-work:
  - ➔changes to more than 10% of the documents
  - ➔re-inspect with full team
- ⌘ Minor re-work: review with moderator
- ⌘ Moderator ensures that follow-up takes place

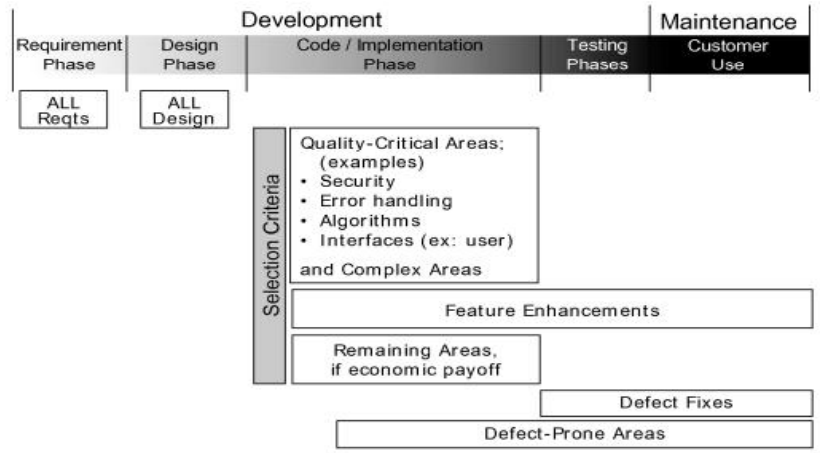
FOLLOWUP	
Purpose	✓Verify Rework
Input	✓Product (Reworked) ✓Defect List (Final)
Activity	✓**Moderator** Verifies Rework ✓**Moderator** Summarizes Inspection
Output	✓Product (Inspected) ✓Inspection Report

Involvement of team members

Stage	Moderator	Producer	Reader	Recorder	Inspector
Planning	x	x			
Overview	x	x	-	-	-
Preparation	x	x	x	x	x
Inspection meeting	x	x	x	x	x
Rework		x			
Follow-up	x	x	(x)	(x)	(x)

- do not hold any roles, but are present
(x) participate only when a work product must be reinspected

What to inspect?



Don't inspect everything, as it takes too much effort!
Just inspect the critical components/parts!

Results

Data that can be collected from inspection:

- Time spent (preparation time, meeting time)
- Defect found
- Size of document (page, LOC, etc)

Compute the following metrics:

- Inspection rate (size/hour)
- Efficiency (defect found/hour)
- Yield (defect removal effectiveness)
- Defect density (defect found/size)
- Estimate remaining defects

Defect Removal Effectiveness

Defect removal effectiveness

$$= \frac{\text{Defects found}}{\text{Estimated total defects}} * 100\%$$

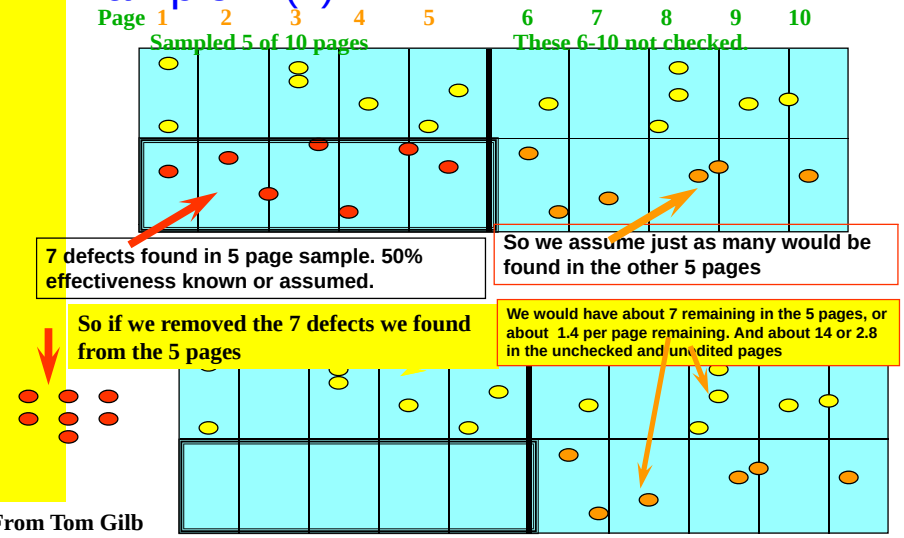
Defect Density

Defect density = Defects found/ Size

Example. If 11 defects found on 15 pages

Defect density = 11/15=0.73 defects/page

Calculating Remaining Defects Example 1 (1)



Calculating Remaining Defects Example 1 (2)

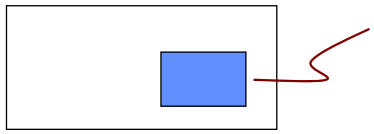
How many remaining defects in the 10 pages?

7 defects from the inspected 5 pages

Assuming the non-inspected pages have the same defect density as the inspected pages, then for the non-inspected pages, there are 14 defects.

Total defects remaining = 7 + 14 = 21 defects

Calculating Remaining Defects: Example 2



Inspect just 2 KLOC
Detect 50 defects

System size: 10 KLOC

What can we conclude?

Assume 50% inspection effectiveness

Thus, the inspected code (2KLOC) should have 50x2 = 100 major defects.

Product defect density = 100/2 = 50/KLOC

=> product contains 50*10KLOC=500 defects

500-50=450 **defects remaining**

Calculating Remaining Defects

In the previous two examples, we only consider the **defect detection effectiveness**.

But, it is often the case that when we try to fix a defect, we are not able to really fix the defect. There is a chance that we may not fix the defect or introduce additional defects.

Fix-fail-rate is the percentage of defects that we try to fix but in fact not fixed.

Calculating Remaining Defects
Adding the fix-fail-rate

Assume

fix-fail-rate = 20%

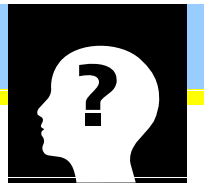
Failed fix=0.2*50= 10 defects *not really fixed*

Defects remaining= 450 + 10 = 460
Remaining defects/KLOC = (460)/10=46

Our formula:

Remaining defects=
[defects((1-effectiveness)/effectiveness + fix-fail-rate)]*

Review Questions



1. Inspection should be used for code only. True/False?
2. No manager should participate in the inspection meeting. True/False?
3. The inspection team should focus on the source document. True/False?

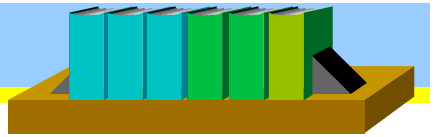
In the inspection process,

4. Who is responsible to the planning?
5. Who is responsible to rework?
6. Who is responsible to follow up?
7. Name the 3 key elements of inspections.

Summary

- Key benefits of inspection are:
  - more cost-effective than testing in detecting defect.
  - Good return on investment.
- Inspection process consists of the following stages: planning, overview, preparation, meeting, rework and follow-up stages.
- Each member of inspection team performs a specific role. The role can be moderator, recorder, reader, producer or inspector. A team member may perform more than one role.
- Key success factors of inspection are:
  1. Members must be prepared for the inspection
  2. The right people should attend the inspection meeting.

References



Ebenau, Robert G. and Strauss, S. H., *Software Inspection Process*. New York: McGraw-Hill, 1994. (ISBN 0-07-062166-7)

Software Engineering Standards Committee of the IEEE Computer Society. *IEEE Standard for Software Reviews, IEEE Std 1028-1997*. The Institute of Electrical and Electronics Engineers, Inc.

Wheeler, David A., Brykczynski, B., and Meeson, R.N. (eds.). *Software Inspection: An Industry Best Practice for Defect Detection and Removal*. IEEE Computer Society Press, 1996. (ISBN 0-8186-7340-0, 293 pages)

Tools

- CodeCollaborator: http://smartbear.com/products/development-tools/code-review/features/

Supplementary Notes

Reflect

- Think of something we discussed in class that can be useful to you?
- Anything you can do better at work from what we learn?

Supplementary Exercises

1. Explain the relationship between V&V, testing and inspection.
2. Inspection follows a well-defined process and assigns a specific role to each team member. Describe in details the four key roles that must be included in an inspection team.