

Software Inspections

Andy Podgurski

EECS Department

Case Western Reserve University

Inspections

- ❑ Involve developers carefully *reviewing documents* or *code* to identify:
 - Errors of omission or commission
 - Ambiguity and lack of clarity
 - Violations of standards
 - Other issues
 - ❑ They typically involve a *team* of developers
 - ❑ Studies have shown inspections are very *effective* for revealing problems
 - ❑ They *complement* other forms of validation
-

Summary of Inspection Results

Reference	Environment	Result
Fagan [40][41]	Aetna Life Casualty	38 defects from 46 detected
	IBM Respond, United Kingdom	93% of all defects were detected by inspections
	Standard Bank of South Africa	Over 50% of all defects detected by inspection
Weller [132]	Bull HN Information Systems	70% of all defect detected by inspection
Grady and van Slack [51]	Hewlett-Packard	60%-70% of all defects detected by inspection
Shirey [122]		60%-70% of all defects detected by inspection
Barnard and Price [4]	AT&T Bell Laboratories	30%-75% of all defects detected by inspection
McGibbon [92]	Cardiac Pacemakers Inc.	70% to 90% of all defects detected by inspection
Collofello and Woodfield [26]	Large real time software project	Defect detection effectiveness is 54% for design inspection, 64% for code inspection, and 38% for testing
Kitchenham et al. [71]	ICL	57.7% of all defects found by code inspection
Franz and Shih [43]	Hewlett Packard	19% of all defects found by inspection
A. Gately [46]	Raytheon Systems Company	The average number of defects found by inspection is 18.2.
Conradi et al. [27]	Ericsson	The average number of defects found by inspection is 3.41.

From *A Survey of Software Inspection Technologies* by Oliver Laitenberger

Inspection Roles

- ❑ *Author* – developer responsible for work product
 - ❑ *Inspectors* – inspect work product
 - ❑ *Scribe or recorder* – records issues
 - ❑ *Moderator* – directs preparation and inspection meeting; reports results to manager
 - ❑ *Manager* – schedules inspection, assigns moderator and team, manages follow-up
-

Inspection Meeting Rules

- ❑ Manager is *not present*
 - ❑ Inspectors *take turns* presenting issues
 - ❑ Inspectors are *tactful*
 - ❑ Producer does *not defend* work
 - ❑ Problems are *not solved* at the meeting
-

Inspection Process

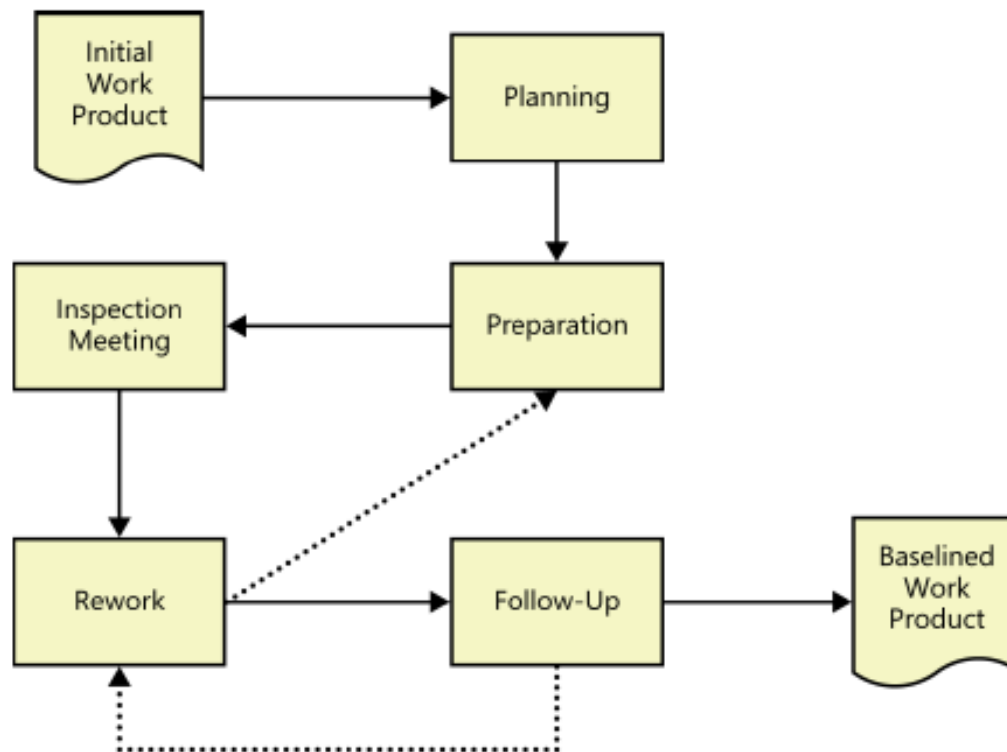


FIGURE 17-2 Inspection is a multistep process. The dotted lines indicate that portions of the inspection process might be repeated if reinspection is necessary because of extensive rework.

Name	Inspection Meeting script
Purpose	To run a moderated inspection meeting.
Summary	In an inspection meeting, a moderator leads a team of reviewers in reviewing a work product and fixing any defects that are found.
Work Products	Input Work product being inspected Output Inspection log
Entry Criteria	A moderator must be selected, as well as team of three to ten people. A work product must be selected, and each team member has read it individually and identified all wording which must be changed or clarified before he or she will approve the work product. A unique version number has been assigned to the work product.
Basic Course of Events	<p>1.Preparation. The moderator distributes printed version of the work product (with line numbers) to each inspector, along with a checklist to aid in the review. Each inspector reads the work product and identifies any defects to be brought up at the meeting.</p> <p>2.Overview. The inspection meeting begins. The moderator verifies that each team member is prepared.</p> <p>3.Page-by-page review. The moderator runs through the work product page by page. Inspectors indicate where there are defects. Each defect is either resolved or left as an open issue. The moderator adds each defect to the inspection log.</p> <p>4.Rework. The author repairs the defects identified in the inspection meeting.</p> <p>5.Follow-up. Inspection team members verify that the defects were repaired.</p> <p>6.Approval. The inspection team approves the work product.</p>
Alternative Paths	<p>1.During step 2, if any team member has not read the work product then the inspection is halted. The meeting is rescheduled and the script returns to step 1.</p> <p>2.During step 4, if an inspection team member discovers additional defects in the work product then the moderator calls another meeting and the process returns to step 1.</p>
Exit Criteria	The work product has been approved.

From www.stellman-greene.com/aspm/content/blogcategory/32/40/

Checklists

- ❑ Useful for reminding inspectors of *important issues* to check for
 - ❑ May *inhibit* them from discovering other issues, however
 - ❑ Perhaps best consulted *after first reading* of work product
-

Requirements Review Checklist

Completeness

- ☐ Do the requirements address all known customer or system needs?
- ☐ Is any needed information missing? If so, is it identified as TBD?
- ☐ Have algorithms intrinsic to the functional requirements been defined?
- ☐ Are all external hardware, software, and communication interfaces defined?
- ☐ Is the expected behavior documented for all anticipated error conditions?
- ☐ Do the requirements provide an adequate basis for design and test?
- ☐ Is the implementation priority of each requirement included?
- ☐ Is each requirement in scope for the project, release, or iteration?

Correctness

- ☐ Do any requirements conflict with or duplicate other requirements?
- ☐ Is each requirement written in clear, concise, unambiguous, grammatically correct language?
- ☐ Is each requirement verifiable by testing, demonstration, review, or analysis?
- ☐ Are any specified error messages clear and meaningful?
- ☐ Are all requirements actually requirements, not solutions or constraints?
- ☐ Are the requirements technically feasible and implementable within known constraints?

Quality Attributes

- ☐ Are all usability, performance, security, and safety objectives properly specified?
- ☐ Are other quality attributes documented and quantified, with the acceptable trade-offs specified?
- ☐ Are the time-critical functions identified and timing criteria specified for them?
- ☐ Have internationalization and localization issues been adequately addressed?
- ☐ Are all of the quality requirements measurable?

Organization and Traceability

- ☐ Are the requirements organized in a logical and accessible way?
- ☐ Are all cross-references to other requirements and documents correct?
- ☐ Are all requirements written at a consistent and appropriate level of detail?
- ☐ Is each requirement uniquely and correctly labeled?
- ☐ Is each functional requirement traced back to its origin (e.g., system requirement, business rule)?

Other Issues

- ☐ Are any use cases or process flows missing?
- ☐ Are any alternative flows, exceptions, or other information missing from use cases?
- ☐ Are all of the business rules identified?
- ☐ Are there any missing visual models that would provide clarity or completeness?
- ☐ Are all necessary report specifications present and complete?

FIGURE 17-4 A defect checklist for reviewing requirements documents.

Generic Checklist for Code Reviews

Structure

- ☐ Does the code completely and correctly implement the design?
- ☐ Does the code conform to any pertinent coding standards?
- ☐ Is the code well-structured, consistent in style, and consistently formatted?
- ☐ Are there any uncalled or unneeded procedures or any unreachable code?
- ☐ Are there any leftover stubs or test routines in the code?
- ☐ Can any code be replaced by calls to external reusable components or library functions?
- ☐ Are there any blocks of repeated code that could be condensed into a single procedure?
- ☐ Is storage use efficient?
- ☐ Are symbolics used rather than "magic number" constants or string constants?
- ☐ Are any modules excessively complex and should be restructured or split into multiple routines?

Documentation

- ☐ Is the code clearly and adequately documented with an easy-to-maintain commenting style?
- ☐ Are all comments consistent with the code?

Variables

- ☐ Are all variables properly defined with meaningful, consistent, and clear names?
- ☐ Do all assigned variables have proper type consistency or casting?
- ☐ Are there any redundant or unused variables?

Arithmetic Operations

- ☐ Does the code avoid comparing floating-point numbers for equality?
- ☐ Does the code systematically prevent rounding errors?
- ☐ Does the code avoid additions and subtractions on numbers with greatly different magnitudes?
- ☐ Are divisors tested for zero or noise?

Loops and Branches

- ☐ Are all loops, branches, and logic constructs complete, correct, and properly nested?
- ☐ Are the most common cases tested first in IF-ELSEIF chains?
- ☐ Are all cases covered in an IF-ELSEIF or CASE block, including ELSE or DEFAULT clauses?
- ☐ Does every case statement have a default?
- ☐ Are loop termination conditions obvious and invariably achievable?
- ☐ Are indexes or subscripts properly initialized, just prior to the loop?
- ☐ Can any statements that are enclosed within loops be placed outside the loops?
- ☐ Does the code in the loop avoid manipulating the index variable or using it upon exit from the loop?

Defensive Programming

- ☐ Are indexes, pointers, and subscripts tested against array, record, or file bounds?
- ☐ Are imported data and input arguments tested for validity and completeness?
- ☐ Are all output variables assigned?
- ☐ Are the correct data operated on in each statement?
- ☐ Is every memory allocation deallocated?
- ☐ Are timeouts or error traps used for external device accesses?
- ☐ Are files checked for existence before attempting to access them?
- ☐ Are all files and devices left in the correct state upon program termination?

Example: Inspection Comments

Reviewer's Name	Sophie (senior QA engineer)
Author's Name:	Dean (junior QA engineer)
Title:	Contract Certification – automated test script #TP-491-A
Review Date:	8/12/03
No. of Review hours:	2

Location	Comments
Global	Script does not adequately copy databases in when the data changes
Case 14	The test plan logs in as "Administrator", this script logs in as "Admin"
Case 52, 53	What exactly is printed? It's not clear, you should be looking for specific data.
Case 61	The test plan tests all of the preferences, but the script only tests the first five.
Global	Script does not adequately copy databases in when the data changes

INSPECTION ACTION LOG							
Name of Chairperson:		Margaret Pink		Name of Project:		ACCT System	
Name of Deliverable:		Functional Design Specification					
Name of Subset to be Inspected:		Maintain Accounts (Subset 7)					
Number of Hours:				Length of Subset			
Logging: 1.75 hours				11 pages			
Discussing: 10.27 hours							
Inspection Number:		FDS-7		Inspection Date:		January 31, 2001	
				Reinspection?		No	
Defect #	Location	Description	Severity*	Class**	Action Taken	Type of Defect	Time to Correct
1.	page 1, paragraph 1	Typo - "end-user" should be hyphenated.	Min	I			
2.	page 1, paragraph 2	Typo - extra space between select and the	Min	I			
3.	page 1, paragraph 3.	Typo - missing commas at the end of bullets 2 and 3.	Min	I			
4.	page 1 Footer	Incorrect format, Account Maintenance needs to be all caps.	Min	I			
5.	page 2, section 2.1, paragraph 1	Should state only a manager can delete an account.	Maj	M			

From
blogs.ittoolbox.com/eai/implementation/archives/sample-inspection-meeting-action-log-13595

Inspection Problems and Remedies

Work product not available on time	Moderator informs manager, who reschedules
Inspectors spend insufficient time reading document	Record reading time Moderator reports poor preparation to manager
Some inspectors dominate meeting	Inspectors take turns reporting issues
Scribe can't keep up	Moderator pauses
Scribe doesn't accurately record issues	Scribe reads back issues Participants review issue list afterward
Confrontation	Moderator intervenes
Producer defends product	Moderator intervenes

Inspection Follow-Up

Follow-Up

Task	Responsible
1. Confirm that the author has addressed every item on the Issue Log. Determine whether the author made appropriate decisions as to which defects not to correct and which improvement suggestions not to implement.	Verifier
2. Examine the modified work product to judge whether the rework has been performed correctly. Report any findings to the author, so rework can be declared complete, incorrect rework can be redone, or items that were not originally pursued can be addressed.	Verifier
3. Report the number of major and minor defects found and corrected and the actual rework effort to the moderator.	Author
4. Check whether the exit criteria for the inspection and for the peer review process have been satisfied. If so, the inspection is complete.	Moderator
5. Check the baselined work product into the project's configuration management system.	Author
6. Deliver Inspection Summary Report and counts of defects found and defects corrected to peer review coordinator.	Author

Additional Sources:

- ❑ Process Impact “Goodies for Peer Reviews”,
www.processimpact.com/pr_goodies.shtml
 - ❑ *A Survey of Software Inspection Technologies* by Oliver Laitenberger,
<ftp://cs.pitt.edu/chang/handbook/61b.pdf>
 - ❑ *Code Review for Teams Too Busy to Review Code*, Atlassian Summit 2010,
www.youtube.com/watch?v=1m3eRFeCInY.
 - ❑ *10 Tips for Effective Code Review*, Atlassian Summit 2016,
https://www.youtube.com/watch?v=fatTnX8_ZRk
-