

Software Testing : REVIEW and INSPECTION





Systematic Software Reviews

Software reviews are a “quality improvement processes for written material”.



Systematic Software Reviews

Help support the objectives of:

- Project management
- Systems engineering
- Verification and validation
- Configuration management
- Quality assurance



Software Life Cycle

Reviews are applicable to software products throughout the software life cycle

- Requirements
- Design
- Coding
- Testing
- Maintenance



Common Attributes:

Systematic reviews have these attributes in common:

- Team participation
- Documented results of the review
- Documented procedures for conducting the review



Goal and Motivation:

By detecting defects early, and preventing their leakage downstream, the higher cost of later detection and rework is eliminated.



Basic Steps:

- Using a static analysis technique,
- Perform a visual examination of the software products
- Detect and correct:
 - Defects
 - Violation of design standards
 - Other problems



What is a Software Product

The term “software product” is used in a very broad sense to describe any document produced during the software lifecycle.



Examples of Software Products

Include:

- **Contracts**
- **Installation plans**
- **Progress reports**
- **Software design descriptions**
- **Release notes**
- **Software requirements specifications**
- **Source code**



What Is a Defect?

- Any occurrence in a work product that is determined to be incomplete, incorrect, or missing
- Any instance which a requirement is not satisfied(Fagan, 1986)
- Informal synonyms:bug, fault, issue, problem



Inspections vs. Reviews

The IEEE Standard for Software Reviews defines 5 types of review:

- Management Reviews
- Technical Reviews
- Inspections (Formal Peer Review)
- Walk-throughs
- Audits



Why 5 types?

Different types of reviews reflect differences in the goals of each review type



Origins: Fagan's Inspection

- Michael E. Fagan
- IBM, Kingston, NY laboratories
- Applied hardware statistical quality and process control methods to “ideas on paper”



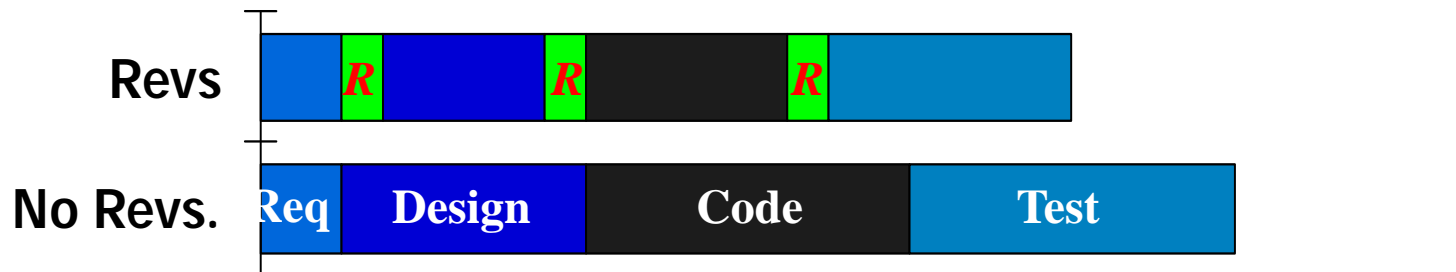
Origins: Continued

- “Design and code inspections to reduce errors in program development” (1976)
- Inspections = improved quality + less cost
- Scope of application expanded



Performance

Reviews improve schedule performance





Performance Continued

Reviews reduce rework.

- **Rework accounts for 44% of development. Cost!**
 - Requirements (1%)
 - Design (12%)
 - Coding (12%)
 - Testing (19%)

Reviews are *pro-active* tests.

- Find errors not possible through testing.

Reviews are training.

- Domain, corporate standards, group.



Quality Improvement

- Reviews can find 60-100% of all defects.
- Reviews are technical, not management.
- Review data can assess/improve quality of:
 - Work product.
 - Software development process.
 - Review process itself.



Quality Improvement Continued

- Reviews reduce total project cost, but have non-trivial cost (~15%).
- Early defect removal is 10-100 times cheaper.
- Reviews distribute domain knowledge, development skills, and corporate culture.



Industry Experience With Reviews

- **Aetna Insurance Company:**
 - FTR found 82% of errors, 25% cost reduction.
- **Bell-Northern Research:**
 - Inspection cost: 1 hour per defect.
 - Testing cost: 2-4 hours per defect.
 - Post-release cost: 33 hours per defect.
- **Hewlett-Packard**
 - Est. inspection savings (1993): \$21,454,000
- **IBM**
 - C system software
 - No errors from time of first compile.



Measuring Impact

Return on Investment:

$$\text{ROI} = \frac{\text{net savings}}{\text{Detection cost}}$$

- **Net savings = cost avoidance – cost to repair now**
- **Detection cost = cost of preparation + cost to conduct**

Details of the Five Types of Software Review





Management Reviews Overview

- Performed by those directly responsible for the system
- Monitor progress
- Determine status of plans and schedules
- Confirm requirements and their system allocation
- Or, evaluate management approaches used to achieve fitness or purpose



Management Reviews Overview Continued

Support decisions made about:

- Corrective actions
- Changes in the allocation of resources
- Or changes to the scope of the project.



Management Reviews Continued

Software products reviewed

- Audit Reports
- Contingency plans
- Installation plans
- Risk management plans
- Software Q/A



Management Reviews Roles

Required:

- Decision Maker
- Review Leader
- Recorder
- Management Staff
- Technical Staff



Management Reviews Outputs

Documented evidence that identifies:

- Project under review
- Review team members
- Review objects
- Software product reviewed
- Inputs to the review
- Action item status
- List of defects identified by the review team



Technical Reviews Overview

Confirms that product

- Conforms to specifications
- Adheres to regulations, standards, guidelines, plans
- Changes are properly implemented
- Changes affect only those system areas identified by the change specification



Technical Reviews Continued

Software products subject to technical reviews

- Software requirements specification
- Software design description
- Software test documentation
- Software user documentation
- Installation procedure
- Release notes



Technical Reviews Roles

The roles established for the technical review

- Decision maker
- Review leader
- Recorder
- Technical staff



Technical Reviews Outputs

Outputs, documented evidence that identifies:

- Project under review
- Review team members
- Software product reviewed
- Inputs to the review
- Review objectives and status
- List of resolved and unresolved software defects
- List of unresolved system or hardware defects
- List of management issues
- Action item status
- Recommendations for unresolved issues
- Whether software product meets specification



Inspection (Formal Peer Reviews)

Confirms that the software product satisfies

- Specifications
- Specified quality attributes
- regulations, standards, guidelines, plans
- Identifies deviations from standard and specification

Failure to do so results in logging a defect



Inspections Continued

Software products subject to Inspections

- Software requirements specification
- Software design description
- Source code
- Software test documentation
- Software user documentation
- Maintenance manual
- Release notes



Inspections Roles

The roles established for the Inspection

- Inspection leader
- Recorder
- Reader
- Author
- Inspector



Inspections Outputs

Outputs, documented evidence that identifies:

- Project under inspection
- Inspection team members
- Inspection meeting duration
- Software product inspected
- Size of the materials inspected
- Inputs to inspection
- Inspection objectives and status
- Defect list (detail)
- Defect summary list
- Disposition of the software product
- Estimate of the rework effort and completion date



Walk-throughs

- Evaluate a software product
- Sometimes used for educating an audience
- Major objectives:
 - Find anomalies
 - Improve the software product
 - Consider alternative implementations
 - Evaluate performance to standards and specs



Walk-throughs Continued

Software products subject to walk-throughs

- Software requirements specification
- Software design description
- Source code
- Software test documentation
- Software user documentation
- Maintenance manual
- Release notes



Walk-throughs Roles

The roles established for Walk-throughs

- Walk-through leader
- Recorder
- Author
- Team member



Walk-throughs Outputs

The outputs of the walk-through

- Walk-through team members
- Software product being evaluated
- Statement of objectives and their status
- Recommendations made regarding each anomaly
- List of actions, due-dates, responsible parties
- Recommendations how to dispose of unresolved anomalies
- Any proposal for future walk-throughs



Audits

The purpose of an audit is to provide an independent evaluation of conformance of software products and processes to applicable;

- Regulations
- Standards
- Guidelines
- Plans
- Procedures



Systematic Software Reviews

Comparison of Review Types

(see handout, Annex B)

IEE Std 1028-1997



Review & Inspection Process

Materials, Methods, and Roles



Review Materials

- Source Document
- Checklist
- Supporting Documents
- Invitation
- Master Plan
- Issue/Defect Log
- Data Summary



Review Methods

- Synchronous
 - Traditional Approach
 - Meeting-based
- Asynchronous
 - Relatively new area
 - Meeting replaced with email (or other electronic) communication



Synchronous Review

- Most popular is the Fagan method
 - Review is separated into 5/6 phases
 1. (Planning)
 2. Overview
 3. Preparation
 4. Inspection
 5. Rework
 6. Follow-up



Planning/Overview

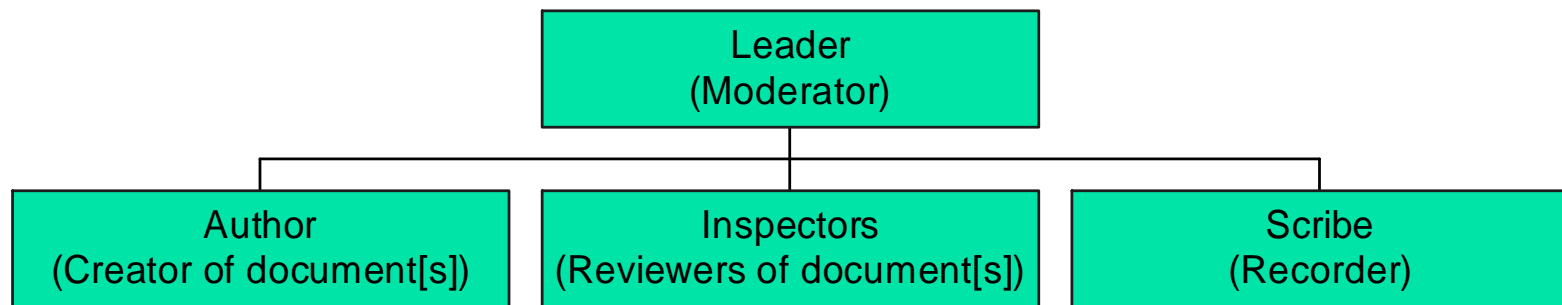


- Reviewers are selected
- Roles are assigned
- Documents are distributed
- General review task is discussed



Review Roles

Roles for a Review





Roles: Leader

- Manages inspection
- Acts as moderator
- Determines document worthiness
- Identifies/invites reviewers
- Assigns roles
- Distributes documents
- Schedules meeting times/locations



Roles: Author

- Creates the document for review
- Assists with answering questions
- Typically not directly involved in review
- Makes corrections to document if necessary



Roles: Inspector/Reviewer

- Complete familiarization of document on time
- Review document(s) for defects
- Look for assigned defects (if appropriate)
- Make use of checklists or other supporting documents
- Contact leader early if problems arise or if the review might be a waste of time

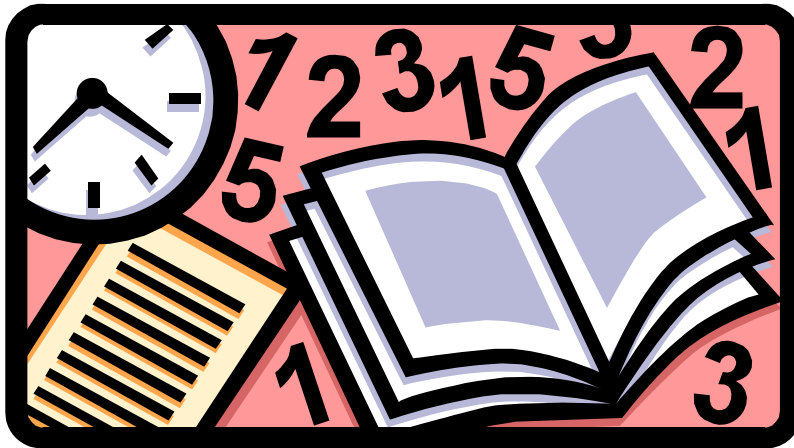


Roles: Scribe/Recorder

- Records issues as they are raised
- Ideally not the moderator or reviewer
- Record information legibly



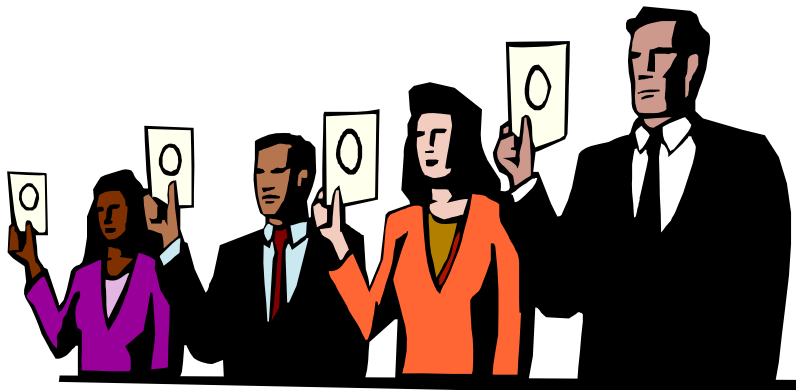
Preparation



- Reviewers acquaint themselves with the documents to be reviewed
- Need to be familiar with material in time for review meeting



Inspection/Review Meeting



- Review team attempts to locate defects
- Defects are not fixed at this point
- Meeting < 2 hours long!



Inspection/Review (cont.)

- Round-robin approach or Reader approach
- Scribe records all issues
 - Where defect was located
 - Why is it a defect (cite requirement or checklist)
 - Suggested severity level (Major, minor)
 - Do Not record names of reviewers with defect
 - Try to make visible to all participants (avoid duplication)



Rework



- Author receives defect log
- Identifies true defects vs. "false positives"
- Fixes defects, provides justification for false positive

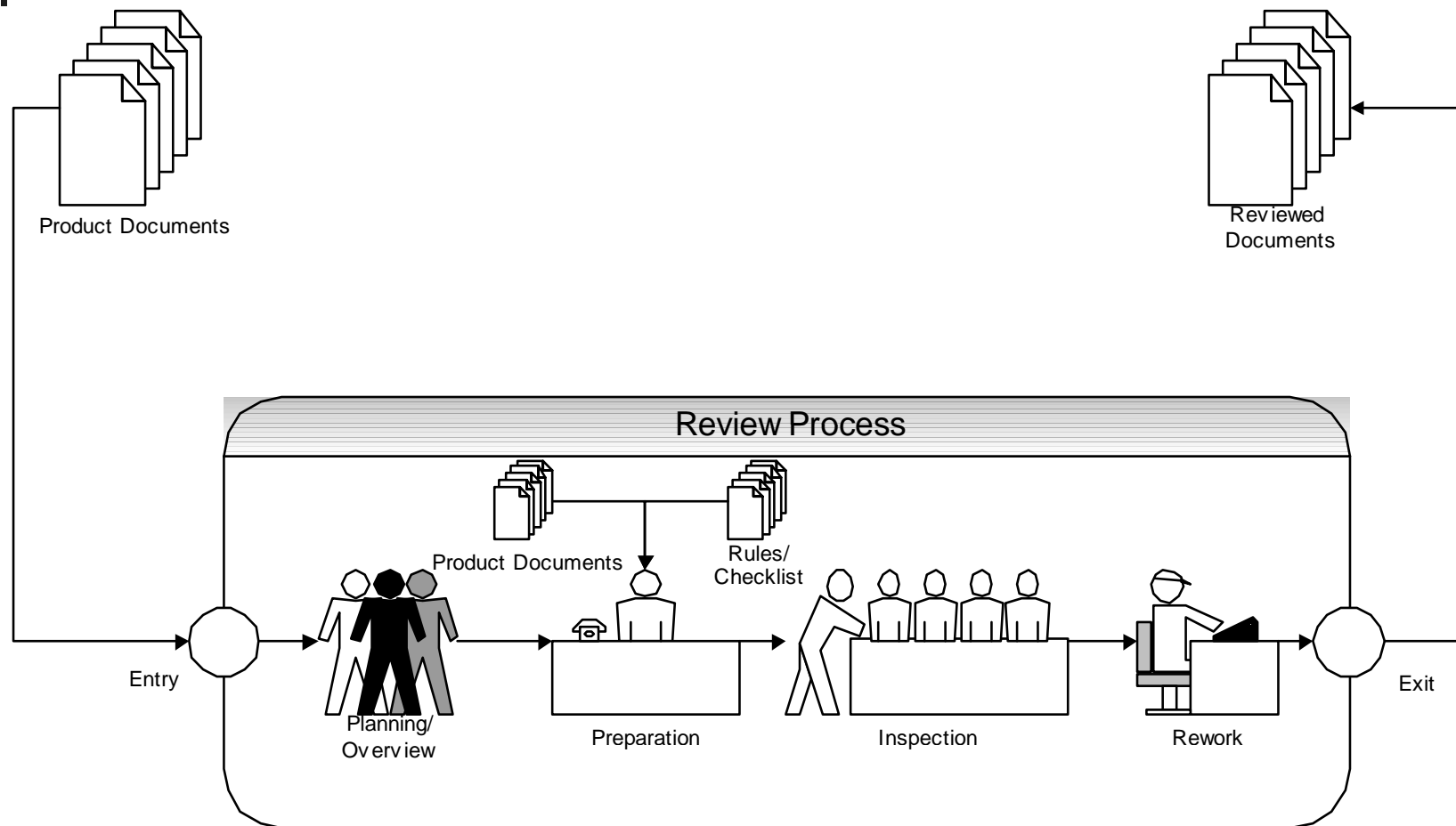


Follow-Up



- Leader verifies all defects have been addressed
- Decides if document passes review or if another review is necessary

Synchronous Review Process





Synchronous/Meeting Review

■ Pros

- Synergy
- Education
- Scheduled Deadline
- Competition
- Minimize “false positives”

■ Cons

- Cost (lost production time vs. cost of defect)
- Difficult scheduling of time/location for wide-spread reviewers



Asynchronous Review

- Formal, Technical, Asynchronous Review Method (FTArm)
- Developed by Philip Johnson at Univ. of Hawaii
 - Phase 1: Select Personnel and Organize Documentation
 - Phase 2: Orientation of Participants to Assigned Task
 - Phase 3: Private Review
 - Phase 4: Public Review
 - Phase 5: Consolidation
- Communication not performed in traditional meeting
 - Email
 - Bulletin Board



FTArm Method

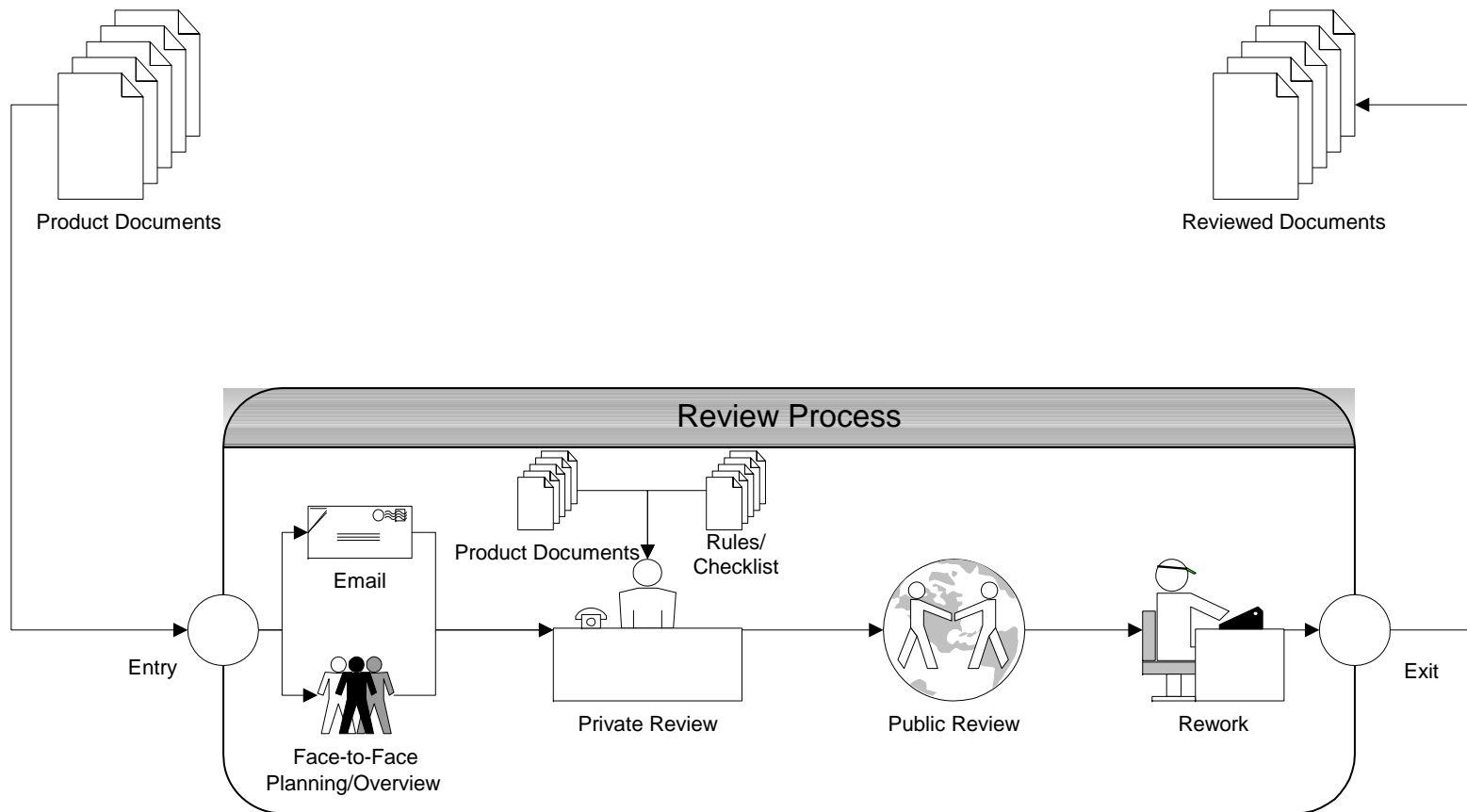
■ Pros

- Reviewers formulate opinions in private
- Opinions are discussed in public and voted on
- During public voting, less experienced reviewers can learn from more experienced reviewers
- Additional defects can be uncovered during public phase
- Compromise can be reached on opposing opinions
- Suitable for wide-spread reviewers

■ Cons

- All ideas must be voted on
- If compromise can not be reached, synchronous meeting should be used to reach one

Asynchronous Review Process





Review Pitfalls

- Insufficient Preparation
- Moderator Domination
- Incorrect Review Rate
- Ego-involvement and Personality Conflict
- Issue Resolution and Meeting Digression
- Recording Difficulties and Clerical Overhead



References/Resources

- Collofello, James S.: "The Software Technical Review Process", SEI Curriculum Module SEI-CM-3-1.5, 1998
- Carnegie Mellon Software Engineering Institute, (visited 3/31/2001), http://www.sei.cmu.edu/str/descriptions_body.html
- Ferguson, John D.: "Groupware Support for Asynchronous Document Review", Proceedings of the 17th International Conference on Computer Documentation, 1999, pp. 185-192
- Gilb, Tomas & Graham Dorothy: Software Inspection, Addison Wesley Longman Ltd, 1996, pp 2-13, 15-25, Glossary
- IEEE Standard for Software Reviews, IEEE Std 1028-1997, 1998 pp 1-26, Annex B
- Johnson, Philip M. & Tjahjono, Danu: "Assessing software review meetings: A controlled experimental study using CSRS", Proceedings of the 1997 International Conference on Software Engineering, 1997, pp. 118-127
- Johnson, Philip M.: "An Instrumented Approach to Improving Software Quality through Formal Technical Review", Proceedings of the 16th International Conference on Software Engineering, 1994, pp. 113-122



References/Resources Continued

- Johnson, Philip M.: The WWW Formal Technical Review Archive, (visited 2/9/2001), <http://www2.ics.hawaii.edu/~johnson/FTR/>
- Johnson, Philip M.: "Introduction to Formal Technical Reviews, A PowerPoint presentation" The WWW Formal Technical Review Archive, <http://www2.ics.hawaii.edu/~johnson/FTR/>
- Knight, John C. & Myers, E. Ann: "An Improved Inspection Technique", Communications of the ACM, 1993, Vol. 36 No. 11, pp. 51-61
- McConnell, Steve: Software Project Survival Guide, Microsoft Press, 1998
- Ranganathan, Kala: "How to Make Software Peer Reviews Work", Quality Process, Bell & Howell Information and Learning Company, American Society for Quality, 2/01/2001
- Rigby, Ken: Design Reviews, (visited 3/6/2001), <http://sparc.airtime.co.uk/users/wysywig/desrev.htm>
- Weiss, Alan R. & Kimbrough, Kerry: Weiss and Kimbrough Inspection Materials, (visited 3/15/2001), <http://www2.ics.hawaii.edu/~johnson/FTR/Weiss/weiss>