Introduction to Formal Technical Reviews

Philip Johnson
Associate Professor
Department of Information and Comp. Sciences
University of Hawaii

johnson@hawaii.edu
http://www.ics.hawaii.edu/~johnson/
808 956-3489

Notkin's version of Johnson's overview
- I removed a lot
- I don't buy everything

What is Formal Technical Review?

A method involving a structured encounter in which a group of technical personnel analyzes or improves the quality of the original work product as well as the quality of the method.

Why review? We test!

Reviews improve schedule performance

Reviews reduce rework.
- Rework accounts for 44% of dev. cost!
- Reqs (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.

Why review? Who benefits?

Formal technical review provides:
- Defect information to the *author*.
- Information on work product and development to *peers*.
- Fault likelihood data to *testers*.
- Product status to *management*.
- Process status to *SPI group*.

True FTR is well-defined

Well-defined process
- Phases (orientation, etc.)
- Procedures (checklists, etc.)

Well-defined roles
- Moderator, Reviewer, Scribe, Author, etc.

Well-defined objectives
- Defect removal, requirements elicitation, etc.

Well-defined measurements
- Forms, consistent data collection, etc.

FTR is effective 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

Reviews reduce total project cost, but have non-trivial cost (~15%)

Upstream defect removal is 10-100 times cheaper.

Reviews disseminate domain knowledge, development skills, and corporate culture.

Industry Experience with FTR

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 (using Cleanroom)
- C system software
- No errors from time of first compile.

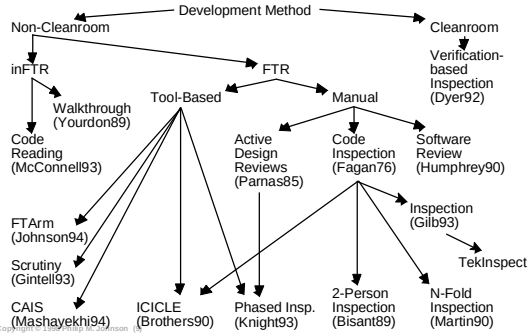
Who, What, and When

Who decides what should be reviewed?
- Senior technical personnel, project leader

What should be reviewed?
- Work products with high impact upon project risks.
- Work products directly related to quality objectives.
- "Upstream" work products have higher impact.

When should review be planned?
- Specify review method and target work products in software development plan/quality plan.

The range of review practice



Families of Review Methods

Method Family	Typical Goals	Typical Attributes
Walkthroughs	Minimal overhead Developer training Quick turnaround	Little/no preparation Informal process No measurement Not FTR!
Technical Reviews	Requirements elicitation Ambiguity resolution Training	Formal process Author presentation Wide range of discussion
Inspections	Detect and remove all defects efficiently and effectively.	Formal process Checklists Measurements Verify phase

Example Planning Data

Planning

1. Inspection ID _____ Date: _____

2. Team
   - Moderator _____
   - Authors _____ _____
   - Reviewers _____ _____
   - _____ _____
   - _____ _____

3. Documents
   - Work Product _____ _____
   - References _____ _____
   - Checklists _____ _____

4. Meetings

	Date	Location	Start	End
Orientation	_____	_____	_____	_____
Review Meeting	_____	_____	_____	_____

5. Planning Objectives
   - ❑ References obtained for work product.
   - ❑ Checklists obtained for work product.
   - ❑ Moderator is trained in TekInspect method.
   - ❑ Team members agree to proposed times/dates.
   - ❑ Moderator's quick review yields less than 5 major issues.
   - ❑ Reviewers understand responsibilities and are committed.

6. Plan. Effort _____ min

Example Orientation Data

Orientation

7. Prep. Goals _____ min/pg x _____ pgs. = _____ prep time/reviewer

8. Orient. Objectives
   - ❑ Reviewers understand scope and purpose of work product..
   - ❑ Reviewers understand checking process, checklists, and referenc
   - ❑ Work product, references, checklists, and checking forms provid

9. Orient. Effort _____ min. meet x _____ particip. = _____ min

Preparation

Planning ⇨ Orientation ⇨ **Preparation** ⇨ Review Mt. ⇨ Rework ⇨ Verify

Objectives
- Find maximum number of non-minor issues.

Procedure for reviewers:
- Allocate recommended time to preparation.
- Perform individual review of work product.
- Use checklists and references to focus attention.
- Note critical, severe, and moderate issues on Reviewer Data Form.
- Note minor issues and author questions on work product.

Example Issue Classification

Critical
- Defects that may cause the system to hang, crash, produce incorrect results or behavior, or corrupt user data. No known work-arounds.

Severe
- Defects that cause incorrect results or behavior with known work-arounds. Large and/or important areas of the system is affected.

Moderate
- Defects that affect limited areas of functionality that can either be worked around or ignored.

Minor
- Defects that can be overlooked with no loss of functionality.

Example checklist

Checklist for Software Quality Plans

❑ 1. Does the plan reference the Tektronix Test Plan process document to be used in this project?

❑ 2. Does the plan list the set of measurements to be used to assess the quality of the product?

❑ 3. Is a rationale provided for each feature to be tested?

4. According to this document, what features won't be tested? Are any missing? List all below:

❑ Does the plan provide a rationale for why each of these features will not be tested?

5. How well does the plan describe how tests will be traced back to requirements?
Check one of the following:
❑ Very well ❑ Fairly well ❑ Poorly ❑ No Traceability

6. Refer to the corresponding software development plan. Does the quality plan discuss each of the test milestones and test transmittal events from this document?
Check all that apply:
❑ I cannot access the software development plan.
❑ The software development plan has no test milestones.
❑ The software development plan has no test transmittal events.
❑ The quality plan has no test milestones.
❑ The quality plan has no test transmittal events.
❑ Both documents include the same set of test milestones and test transmittal events.

Example Preparation Data

1. Inspection ID _____ **2. Document:** _____ **3. Name:** _____

4. Critical, Severe, and Moderate Issues

Num	*Location*	*Severity*	*Chk/Ref*	*Description*

5. Effort: ____ min **6. Issue Totals** _____ critical _____ severe _____ moderate _____ minor _____ author Q's

7. Preparation Objectives
❑ Work product has been completely checked.
❑ All critical, severe, and moderate issues are noted on this form.
❑ All minor issues and author questions are noted on the work product.

Why not write on the work product?

Advantages of Reviewer Data Sheet:
- Minor issues are "pre-filtered" from review meeting, saving meeting time.
- Reviewers articulate issues clearly during preparation, saving meeting time.
- Preparation statistics gathering simplified.
- Preparation effectiveness (% true defects, % redundancy) and checklist effectiveness is measurable.
- Issues can be presented in order of importance.
- Data sheet indicates effectiveness of checklists.

Why not write on the work product?

Disadvantages of Reviewer Data Sheet:
- Requires extra time (15 minutes?)
- Discourages last minute preparation.
- Makes quality of preparation more visible.

Review Meeting

Planning ⇨ Orientation ⇨ Preparation ⇨ **Review Mt.** ⇨ Rework ⇨ Verify

Objectives
- Create consolidated, comprehensive listing of non-minor issues.
- Provide opportunity for group synergy.
- Improve reviewing skill by observing others.
- Create shared knowledge of work product.

Procedure
- Moderator requests issues sequentially.
- Reviewers raise issues.
- Scribe notes issues on Scribe Data Sheet.
- Scribe Data Sheet is visible to everyone.

Example Review Meeting Data

Review Meeting

	Aggregrate Checking Data R1	R2	R3	R4	R5	R6	Total
10. Prep. Effort	____ +	____ +	____ +	____ +	____ +	____ =	_____ min
11. Critical Iss.	____ +	____ +	____ +	____ +	____ +	____ =	_____ iss.
12. Severe Iss.	____ +	____ +	____ +	____ +	____ +	____ =	_____ iss.
13. Moderate Iss	____ +	____ +	____ +	____ +	____ +	____ =	_____ iss.
14. Minor Iss.	____ +	____ +	____ +	____ +	____ +	____ =	_____ iss.
15. Author Q's.	____ +	____ +	____ +	____ +	____ +	____ =	_____ Qs

Review Meeting (cont.)

16. Rev. Meet. Objectives
- ❑ All reviewers present. List absent:_____
- ❑ All reviewers prepared sufficiently for meeting.
- ❑ All issues noted by Scribe and understood by Author for rework
- ❑ Any problems with inspection process have been noted.

17. R.M. Effort _____ min. meet x ___ particip. = _____ min

Example Rework Data

1. Inspection ID _____ **2. Document** _____ **3. Author** _____

4. Issue Disposition

Num	*Fixed*	*Type*	*Explanation*
____	____	_____	_____
____	____	_____	_____
____	____	_____	_____

5. Effort _____min

6. Rework Objectives
- ❑ Outcome of all Review Meeting Data Sheet issues are noted on this form.
- ❑ All minor issues have been addressed.
- ❑ No known defects remain in the work product.

Verify

Planning ⇨ Orientation ⇨ Preparation ⇨ Review Mt. ⇨ Rework ⇨ **Verify**

Objectives
- Assess the (reworked) work product quality.
- Assess the inspection process.
- Pass or fail the work product.

Procedure for moderator:
- Obtain reworked product and Author Data Sheet.
- Review work product/data sheet for problems.
- Provide recommendation for work product.
- Perform sign-off with reviewers.
- Compute summary statistics for inspection.
- Generate any process improvement proposals.
- Enter review data into quality database.

Example Verify Data

Verify

18. Total Effort

+ _____	Planning	(Line 6)
+ _____	Orientation	(Line 9)
+ _____	Preparation	(Line 10)
+ _____	Review Meeting	(Line 17)
+ _____	Rework	(See Rework Data Sheet)
+ _____	Verify	
= _____	**Total Inspection Effort**	

19. Total Defects Removed

_____	Critical	(All from Rework Data Sheet)
+ _____	Severe	
+ _____	Moderate	
+ _____	Minor	
= _____	**Total Defects Removed**	

20. Method Variations
- ❑ Reviewer forms were not filled out completely.
- ❑ Review meeting involved issue discussion and resolution.
- ❑ Checklists did not appear to be helpful.
- ❑ References did not appear to be helpful.
- ❑ Other: _____

21. Verify Objectives
- ❑ Moderator's quick review yields less than 2 major issues.
- ❑ Moderator has collected all TekInspect forms for filing.
- ❑ Moderator has entered data into quality engineering database.

22. Process Improvement _____

23. Inspection Status
- ❑ Pass
- ❑ Conditional Pass: _____
- ❑ Fail: _____

Moderator signature:_____ Date: _____

I agree/disagree with the moderator's decision:

❑ Agree ❑ Disagree _____Date: _____

What is PSP?

PSP is the "Personal Software Process"

PSP was invented by Watts Humphrey and is currently promoted by the Software Engineering Institute.

PSP is a technique to support individual, empirically-guided process improvement.

For more details, see the authoritative reference: "A Discipline for Software Engineering", Watts Humphrey, Addison-Wesley, 1995.

PSP Review Characteristics

PSP reviews are:

- Conducted on designs and code.
- Have a well-defined process.
- Always conducted prior to first compile.
- Based upon checklists created from personal defect history.
- Measured and evaluated for process improvements.

PSP reviews satisfy all requirements for FTR except that they are not a *group* process.

PSP reviews start at PSP2

PSP Review Process

Checklist Generation:

- Use defect history to generate checklist.

Design/Code review:

- Apply checklist at end of design/code phases to eliminate defects.

Evaluation/Improvement:

- Calculate time-savings of reviews.
- Calculate effectiveness of checklist items.
- Improve checklist to track defects created.

PSP Reviews vs. Generic Inspection

Individual process	Group process
Bottom-up driven	Top-down and bottom up
No "roles"	Moderator, Scribe, etc.
No group synergy	Group synergy/learning
Producer responsible for process improvement	Software engineering process group (SEPG) responsible for process improvement.

Critical Success Factor: Checklists

Checklists guide reviewers to areas prone to defects.

Checklists may be stated as a yes/no question:

- "Are all strings null terminated?"

Checklists can also stimulate mental modelling:

- "After a fork, what happens if a child exits immediately?"

Checklists should be combined with general analysis.

- Don't trust checklists to be comprehensive!

Checklists are specific to work product type and development phase.

Critical Success Factor: Effective Preparation

Effective preparation requires both:

- Comprehension: the nature of the entire document.
- Analysis: inter-document consistency and adequacy.

Focus on:

- What is present but not adequate.
- What is missing but should be there.
- What unique skills and experiences can you bring to bear on the work product?

Allocate enough time to prepare!

- Make multiple passes over document.
- Let it "sit overnight".
- Don't prepare right before the review.

Critical Success Factor: Measurement

The goal of Inspection is to detect and remove all defects efficiently and completely.

We measure:

- Time spent on each phase.
- Number of issues of each type discovered.
- Utility of review meeting, checklists, etc.

Analysis over time suggests:

- New and better checklist items.
- Improvements to inspection process,by identifying poor quality review.
- Improvements to software development process, by identifying poor quality work products.
- Improvements to standards.

Critical Success Factor: The moderator

Indicators of effective inspection moderators:

- Work products are inspected when ready.
- Meeting dates are aggressive but do-able.
- Author overviews are useful or omitted.
- Checklists and reference materials are useful.
- Review meeting focuses on issue detection.
- Author does not feel threatened.
- Rework is verified carefully.
- Improvements to inspection and software development process are discovered.
- Participants feel the method effectively improved quality.
- Everyone wants to do it again!

Further references

Software Inspection, Tom Gilb and Dorothy Graham, Addison-Wesley, 1993.

The WWW FTR Archive,
- http://www.ics.hawaii.edu/~johnson/FTR/

Software Inspection: An Industry Best Practice,
David Wheeler, Bill Brykczynski, and Reginald Meeson.

(For PSP) *A Discipline for Software Engineering*,
Watts Humphrey, Addison-Wesley, 1995.