



Traceability of Implementation to Design and Requirements Specifications: A Formal Technical Review Method (Reverse Engineering Tool)

RASHMI YADAV, RAVINDRA PATEL and ABHAY KOTHARI

Department of Computer Science Engineering and Application,
Rajiv Gandhi Technical University, Bhopal, India.

(Received: June 17, 2015; Accepted: August 11, 2015)

ABSTRACT

The software quality of a software product is challenging for the software industry. The reason that software industry demand of product in less time period so developer or team in on stress due to that they are missing something so software product not up to mark. The purpose of this paper viewing significance of formal technical review of requirement gathering and design any software, products or tools and reviews missing a thing and improve software product quality. This research paper elaborates how to perform requirement gathering and review that, for the reverse reverse engineering tool.

Key word: Formal technical review, Traceability ,Reverse Engineering.

INTRODUCTION

A technical review (TR) is the most effective filter from a quality control standpoint. Conducted by software engineers (and others) for software engineers, the TR (Technical Review) is an effective means for uncovering errors and improving software quality¹

Formal techniques are not necessarily mathematical specification languages, but can be graphical techniques as well, provided that the syntax and semantics of these techniques are precisely described. Object Oriented Analysis methods which primarily use graphical specification techniques. The purpose of this study is to look to

what extent these graphical specification techniques are formalized. Despite of several advances in automated verification and validation, human review of software artifacts is still a unique important method for software quality improvement. Formal technical review (FTR) is an umbrella term for review methods involving a structured encounter where a group of technical personnel analyzes an artifact in order to improve both the quality of the product and the review process.

In addition, the formal technical review serves as a training ground, enabling junior engineers to observe different approaches to software analysis, design, and implementation. The formal technical review also serves to promote

backup and continuity because a number of people become familiar with parts of the software that they may not have otherwise seen.

The formal technical review is actually a class of reviews that includes walk-through's, inspections, round-robin reviews and other small group technical assessments of software. Each formal technical review is conducted as a meeting and will be successful only if it is properly planned, controlled, and attended.

Literature Survey

A review process can be defined as a critical evaluation of an object. Although the term review process often has many connotations, particularly for those with industry experience, the intent of this module is to use this term in its most general sense.

Formal technical review (FTR) is an essential component of all modern software quality assessment, assurance, and improvement techniques, and is acknowledged to be the most cost-effective form of quality improvement when practiced effectively².

Senior technical personnel, project leader decides what should be reviewed. Work products with high impact upon project risks should be reviewed. Specify review method and target work products in the software development plan/quality plan. [Philip Johnson]

Boniface C. Nwugwo³ Formal Technical reviews are the examination of the software product to identify the faults in this work's author gives the defect Amplification model if we haven't done formal technical review the error is amplified and generates thirteen errors. If we do the formal technical review generates three errors if we detect an error early it is less costly rather than we found error later. Formal technical review is found defect early reduce the overall cost of the product. Formal techniques can be applied in all phases of software engineering like requirement specification, design, code, testing, user documentation, any other defined development product.

Objective of Formal technical review

According to¹ the basic objective of formal technical review is:

- (1) To uncover errors in functional, logic, or implementation for any representation of the software;
- (2) To verify that the software under review meets its requirements;
- (3) To ensure that the software has been represented according to predefined standards;
- (4) To achieve software that is developed in a uniform manner.
- (5) To make projects more manageable.

Dr. Jody paul⁴ gives the formal technical review through the walk-through and the checklist.

Walkthroughs

In the sections that follow, guidelines similar to those for a walkthrough are presented

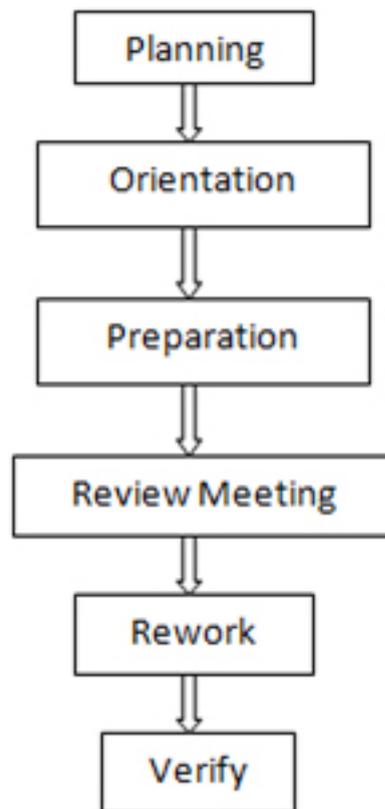


Fig. 1: The generic Inspection Review Process

as a representative formal technical review. Walk-throughs can be viewed as presenting reviews in which a review participant, usually the developer of the software being reviewed, narrates a description of the software and the remainder of the review group provides their feedback throughout the presentation. Features of walkthrough are less formal, producer presents or provides information

Checklist: Requirements Analysis

- Is information domain analysis complete, consistent, and accurate?
- Requirement satisfies the Tool Development objective.
- Is problem partitioning complete?
- Are external and internal interfaces properly defined?
- Does the data model properly reflect data objects, their attributes, and relationships?
- Are all requirements traceable to system level?
- Is performance achievable within the constraints imposed by other system elements?

- Are requirements consistent with schedule, resources, and budget?
- Are validation criteria complete?

In our work we frame the requirement set for re-engineering tool by the formal technical review. Various ways of formal technical review Here we choose the checklist for preparing the requirement of re-engineering tool using formal technical review.

The Formal Technical Review Process:

The Review Meeting

Regardless of the formal technical review format that is chosen, every review meeting should abide by the following constraints:

- Between three and five people (typically) should be involved in the review.
- Advance preparation should occur, but should require no more than two hours of work for each person.
- The duration of the review meeting should be less than two hours.

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

Fig. 2: Inspection review form for planning phase

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 references.
 Work product, references, checklists, and checking forms provided.

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

Fig. 3: Inspection review form for orientation phase

Given these constraints, it should be obvious that a formal technical review focuses on a specific (and small) part of the overall software.

The review meeting is attended by the review leader, all reviewers, and the producer. One of the reviewers takes on the role of the recorder; that is, the individual who records (in writing) all important issues raised during the review. The formal

technical review begins with an introduction of the agenda and a brief introduction by the producer. The producer then proceeds to “walk through” the work product, explaining the material, while reviewers raise issues based on their advance preparation. When valid problems or errors are discovered, the recorder notes each. At the end of the review, all attendees of the formal technical review must decide whether to (1) accept the product without

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.

Fig. 4: Inspection review form for checklist phase

Review Meeting	Aggregate Checking Data						Total	
	R1	R2	R3	R4	R5	R6		
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. Mect. Objectives <input type="checkbox"/> All reviewers present. List absent: _____ <input type="checkbox"/> All reviewers prepared sufficiently for meeting. <input type="checkbox"/> All issues noted by Scribe and understood by Author for rework <input type="checkbox"/> Any problems with inspection process have been noted.							
	17. R.M. Effort	_____ min. meet	x	_____ particip.				= _____ min

Fig. 5: Inspection review form for checklist phase

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.

Fig. 6: Inspection review form for rework phase

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: _____

Fig. 7: Inspection review form for verifying phase

Methods	Functional Req.	Non functional Req.	Support changes in Requirement	Design Technique Used	Support for Traceability	Principal underlying the method	Suitable Environment	Traceability in Context with Proposed tool
Feature oriented Domain Analysis [5] [6] [7]	Yes, Separate architecture model than nonfunctional	Yes. Separate architecture model	It emphasizes on users understanding of how the application will works on live domain so requirement are and support separation of concern then it easy to change	Architecture model is based on feature model Has some resemblance to object oriented techniques.	Feature model	Feature oriented	It is used in mature domain standard terminology domain expert and up-to-date documentation available	Reengineering tool have four basic components extractor, repository, and analyzer and visualize out of which extractor and analyzer are relatively complex and all together there is need for numerous quality or nonfunctional requirements, so here methods are recommended keeping these facts in view
Object oriented transition [8]	Yes	Yes	Yes	Object and class related diagram	Yes	Convert Object oriented Analysis model to object oriented Design Model	Commercial Application	√
Use Case Maps [9]	Yes	Yes ,as behavioral frameworks are used to evaluate and make architectural decision at higher level of design	Yes	Related use cases are shown on the map like diagrams, this notation is useful for capturing, elicitation and validation of use-case this helps in architectural design and test case generations	Yes	Scenario based, behavioral framework is used to evaluate and make architecture decision.	Object oriented and commercial application	√
Weaving together requirement into architecture [10]	Yes	Yes	Yes, Very flexible		Implicitly yes	Twins Peak model is used which supports changing requirements	This methods is suitable where we are not very confident for frozen the requirement or we development this type of software first time	-
Problem frame [12]	Yes	Yes	They works on frame formats and short delivery so it is unlikely as problem frames.	Real problems can be modeled as problem frames which describes architectural structures, services and artifacts as a part of problem domain.	Yes	A problem is collection of many simple sub-problems.	Need Early delivery	√

Goal based transition[11]	Yes	Same functional architecture is recursively refined to accommodate non functional requirements.	Yes	Goal based transition method uses architectural specification language.	Yes	Requirement architectural from system goals.	Non functional requirement highly required	√
Rule Based decision making[13]	Yes	-	Rule base can be easily updated so, yes,	Reasoning and organization module	Yes	Automated rule based reasoning	Application domain need flexibility	-
Architecturing Requirement [14]	Yes	Yes, as refactoring of requirements is there	Yes	Requirements elicitation, architectural requirement, design, implementation phase phase.	Implicitly implemented, as architectural requirements phase replaces architecture design phase.	Implicit analysis	Requirement set not confidently design software design first time	√
Patterns[15]	Yes	Yes on priority	Yes	Design patterns	Poor	Non functional requirements, then functional	Highly desirable for nonfunctional requirement	

Fig. 8: Traceability of Metrics to design and requirements specifications

further modification, (2) reject the product due to severe errors (once corrected, another review must be performed), or (3) accepts the product provisionally (minor errors have been encountered and must be corrected, but no additional review will be required). The decision made, all formal technical review attendees complete a sign-off, indicating their participation in the review and their concurrence with the review team’s findings.

Review Reporting and Record Keeping

During the formal technical review , a reviewer (the recorder) actively records all issues that have been raised. These are summarized at the end of the review meeting and a review issues list is produced. In addition, a formal technical review summary report is completed.

A review summary report

Answers three questions:

1. What was reviewed?
2. Who reviewed it?
3. What were the findings and conclusions?

The review summary report is a single page form (with possible attachments). It becomes part of the project historical record and may be distributed to the project leader and other interested parties.

The review issues list serves two purposes:

- (1) To identify problem areas within the product.
- (2) To serve as an action item checklist that guides the producer as corrections are made. An issues list is normally attached to the summary report.

It is important to establish a follow-up procedure to ensure that items on the issues list have been properly corrected. Unless this is done, it is possible that issues raised can “fall between the cracks.” One approach is to assign the responsibility to follow up to the review letter.

Review Guidelines

Boniface C. Nwugwo [3] gave Guidelines for the conduct of formal technical reviews must be established in advance, distributed to all reviewers, agreed upon, and then followed. A review that is uncontrolled can often be worse than no review at all. The following represents a minimum set of guidelines for formal technical reviews:

Review the product, not the producer

A formal technical review involves people and egos.. Errors should be pointed out gently; the tone of the meeting should be loose and constructive; the intent should not be to embarrass or belittle.

Set an agenda and maintain it

A malady of any meetings is drift. In a formal technical review must be kept on track and on schedule.

Limit debate and rebuttal

When an issue is raised by a reviewer, there may not be universal agreement on its impact. Record the issue for further discussion off-line, rather than spend time debating the question.

Don't attempt to solve every problem noted

A review is not a problem-solving session. Problem solving should be postponed until after the review meeting.

Take written notes

Sometimes it is a good idea for the recorder to make notes on a wall board, so that wording and priorities can be assessed by other reviewers as information is recorded.

Limit the number of participants' preparation

Two heads are better than one, but 14 are not necessarily better than 4.

Insist upon advance preparation

All review members must prepare in advance. The written command should be solicited by the review leader.

Develop a checklist for each product that is likely to be reviewed

A checklist helps the review lead to structure the formal technical review meeting and helps each reviewer to focus on important issues.

Allocate resources and schedule time for the formal technical reviews

To be effective formal technical review scheduled be scheduled as a task during the software engineering process. In addition, time should be scheduled for the inevitable modifications that will occur as the result of a formal technical review .

Review your early reviews

Debriefing can be beneficial in uncovering problems with the review process itself. The very first product to be reviewed should be the review

guidelines themselves and your development standard.

Restrict a Design Review to Reviewing one design

Don't use a design review to compare two or more designs, but use two or more designs at once, the review may turn into a yelling contest for the advocates of the various alternatives.

Conduct meaningful training for all reviewers

To be effective, all reviews, participants should receive some formal training.

Inspection Review

Inspection review process in six phases in figure 1 they are: Planning, orientation, preparation, review meeting, rework and verify and in inspection/ review following function in phase wise, Choose a team, materials, dates. Present product, process, goals. Check product, note issues. Consolidate issues. Correct defects. Verify product/process quality and details are discussed in below section

Following phases of Inspection Review

1. Planning: In planning phase -Gather review package, work product, checklists, references, and data sheets. Form inspection team and determine dates for meetings. Procedure for establishment planning -Moderator assembles team and review package, moderately enhances checklist if needed, moderator plans dates for meetings, moderator checks work product for readiness and moderator helps author prepares overview. Figure 2 shows in the details of the inspection, review form in which mention inspection id., team member etc.
2. Orientation: In this phase first, the author provides an overview, Reviewers obtain review package, Preparation goals established and Reviewers commit to participate. Procedure for establishment Orientation -Moderator distributes a review package, the author presents an overview, if necessary, scribe duty for review meeting assigned and moderator review preparation procedure. Figure 3 shows in the details of the orientation review.

- 3 Checklist: In this phase, Find the maximum number of non-minor issues, procedure for reviewers: Allocate recommended time for preparation, perform an individual review of work product, use checklists and references to focus attention, note critical, severe, and moderate issues on reviewer data form and note minor issues and author questions on work product. . Figure 4 shows in the details of the checklist review
- 4 Review Meeting: In this phase, Create consolidated, comprehensive listing of non-minor issues, provide opportunity for group synergy, improve is reviewing skill by observing others and create a shared knowledge of work product. The procedure for reviewing mean-Moderator requests, issues sequentially, reviewers raise issues, scribe notes issues on Scribe Data Sheet and scribe data sheet is visible to everyone. Figure 5 shows in the details of the review meeting form.
5. Rework: In this phase, Assess each issue, determines if it is a defect, and remove it if necessary ,produce written disposition of non-minor issue and resolve minor issues as necessary.
6. Verify: following function mention in verify phases ,assess the (reworked) work product quality, assess the inspection process, Pass

or fail the work product. Procedure for moderator: Obtain reworked on 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. Figure 6 shows in the details of the verify phase

Formal technical review Confirm traceability of implementation to design and requirements specifications. The figure 8 shows how traceability apply in requirement specification phases.

CONCLUSION

In this paper concentrate on formal technical review, which are help to us for design requirement set which are validate using different phases of posses of formal technical review and after finding requirement , filtering them on the bases of the available feature set in the different method sets and next to filtering for design qualitative requirement set for reverse engineering tool and the final, formal technical review Confirm traceability of implementation to design and requirements specifications.

REFERENCES

1. Rogers S. Pressman "Software Engineering A practioners Approach" Fifth Edition. 2006
2. Philip M. Johnson supporting technology transfer of formal technical review through a computer supported collaborative review system, in *Proceedings of the Fourth International Conference on Software Quality*, Reston, VA., October 1994. [Philip M. Johnson] Philip M. Johnson "Reengineering Inspection" *Communications of the ACM*, 41(2): (1998).
3. Boniface C. Nwugwo" Implementing a Formal Technical Review Process in a Software Development Environment. Kodak software Quality Mini Conference November 12, 1993
4. Dr. Jody paul structured walkthroughs and Formal Technical review. <http://www.jodypaul.com/SWE/WT/walkthroughs.html#top>
5. Kang, Kyo C. "Feature-Oriented Domain Analysis Feasibility Study "(CMU/SEI-90-TR-21, ADA235785) CMU-SEI 1990.
6. C. Reid Turner "A conceptual basis for feature engineering", *The Journal of Systems and Software* 49: 3-15 ((1999)).
7. Dongyun Liu Hong Mei "Mapping requirements to software architecture by feature-orientation" *Requirements Engineering*, 25(2): 69-76 (2003).
8. H. Kandl, "Difficulties in the Transition from OO Analysis to Design," *IEEE Software*, : pp. 94-102(1999).
9. J.A.Buhr,"Use Case Maps as Architectural

- Entities for Complex Systems," *IEEE Transactions on Software Engineering*, **24**: 1131-1155, (1998).
10. Nuseibeh, "Weaving together requirements and architectures," *IEEE Software*, **34**: 115-11, (2001).
 11. M. Bradozzi and D. E. Perry, "From Goal-Oriented Requirements to Architectural Prescriptions: The Preskriptor Process," The Second International Workshop on Software Requirements and Architectures (STRAW '03) at ICSE'03, Portland, OR, 2003.
 12. J. G. Hall, M. Jackson, R. C. Laney, B. Nuseibeh, and L. Rapanotti, "Relating Software Requirements and Architectures using Problem Frames," IEEE Joint International Requirements Engineering Conference (RE'02), Essen, Germany, 2002.
 13. W. Liu and S. Easterbrook, "Eliciting Architectural Decisions from Requirements using a Rule-based Framework," The Second International Workshop on Software requirements and Architectures (STRAW '03) at ICSE'03, Portland, OR, 2003.
 14. W. Liu, "Architecting Requirements," Doctoral Consortium at RE'04, Kyoto, Japan, 2004.
 15. Lihua Xu, Hadar Ziv "An Architectural Pattern for Non Functional Requirements" Elsevier Science Direct 2006.