International Journal of Recent Engineering Research and Development (IJRERD)
Volume No. 02 – Issue No. 01, ISSN: 2455-8761
www.ijrerd.com, PP. 68-74

Risk Based Spiral Review Approach (RB-SRA) for Software Quality Assurance

Nargis Fatima[1], Sumaira Nazir[2]

[1]*National University of Modern Languages, Department of Engineering, Faculty of Engineering and Computer Science, Sector H-9 Islamabad, Pakistan*

[2]*National University of Modern Languages, Department of Engineering, Faculty of Engineering and Computer Science, Sector H-9 Islamabad, Pakistan*

Abstract: Increasing demand of software is a source of promotion for the development of processes to ensure the quality of software. Software reviews are an effective technique for software quality assurance. Software review is static examination of software artifact to detect defects or any other deviations from the customer requirements. Although the defect detection is an essential technique, the defect correction is equally significant. This paper proposes a risk based spiral review approach to ensure software quality by effective defect detection, defect correction and risk analysis strategies. The proposed approach provides multiple benefits such as it helps to increase detect defection rate, resolve issues through multiple spirals, by risk based analysis strategy to identify and mitigate the risk before review execution. Moreover the proposed approach can also be implemented with any existing review process. Proposed approach is evaluated through experiment which proves its effectiveness for defect detection and defect correction.

Keywords: software review, software inspection, software artifact, risk, spiral approach

1. Introduction

Software Quality is highly needed to satisfy the customers. The poor quality software can be a root cause of huge disaster in critical system [1]. Software organizations must use quality improving procedures, right from the start of the software development. It is essential to control and monitor product quality throughout the software development phases [2]. The presence of defects in artifact can lead to poor quality rtifacts and software product. Defect discovery is foremost problems in software development [3].

Software reviews or inspection are static quality assurance technique which are used to validate and verify software artifact to accomplish quality in software products [4] [5] [6]. Both the terms inspection and review are used interchangeably in the literature [7]. These techniques can be used to validate and verify the quality of software by early defect detection prior to testing. They are effective procedures that can lessen the overhead of testing activity. In fact they are one of those software engineering revolutions that had the ability to influence real process change [8]. Software reviews are extensively used defect detection technique to develop high quality software by detecting and removing defects prior software delivery [9]. For almost thirty five years inspections and reviews have been considered best practice to verify software [10].

Review process was primarily proposed by Michael Fagan at IBM . Fagan's work provides the basis for all the future work done in the field of software reviews. After Fagan's review process the research regarding software reviews was performed in two sub fields such as software review procedure and computer support for software review [7], [10].

The software review is a process of defect detection from the artifacts produced during the software development life cycle. It can be applied at any phase of software development [11]. The fundamental elements of the software review process are goal, role, procedure and artifacts. The "goals" defines the purpose of conducting the review process. The "role" is the functional aspect associated with a pre-defined responsibility and it can vary depending upon review process. Roles are various such as author, presenter, leader, reviewer, reader etc. [7]. "Author" is a role who is assigned a responsibility to design or modify an artifact. Another example of role is a "leader" who's responsibility is to manage the complete review process besides verifying the changes made by the author. The "reviewer" role has the responsibility is to identify the defects in the artifact [12]. The other fundamental element of review process is "procedure" which includes the steps and activities such as planning, review, preparation, rework, follow-up etc. [8]. "Artifact" is a work product that is developed at the end of each software development phase. The artifact should be reviewed for its conformance to customer requirements and standards [10], [12].

Through literature survey it is analyzed that all existing review processes broadly contains three phases i.e. initiation phase, evaluation phase and closing phase [9]. The purpose of "initiation phase" is to do the necessary preparation of the review process. It includes a number of activities such as preparation of the artifact

for review, selection of review team members, distribution of artifact, review planning, presentation of artifact, checking artifact for entry criteria, informing the schedule and place for review meeting. "Evaluation phase" is the second most important phase of software review procedure and it is considered as the central part. The purpose of this phase is detection of defects to make the artifact defect free. The last major phase in review procedure is "closing phase" in which review process completion activities are performed. It includes the activities such as defect correction and its verification [13]. The aim of the research is to propose risk based spiral review approach (RB-SRA) for reviewing the artifacts produced during software development life cycle. The proposed approach is an effective way to identify and correct the maximum number of defects from the artifact which in turn enhance the quality of software product. This approach allows the software engineering team to save valuable resources by multiple spiral cycles and by reducing cost of rework.

The paper is structured as section 1 which gives the brief introduction of the software review process. Section 2 discusses the background study of the relevant area. Section 3 provides the details of proposed risk based spiral review approach. Section 4 consists of experimental evaluation of proposed approach. Last section provides the conclusion and future work suggestions.

2. Background study

Fagan introduced software review process in 1976 which served as the basis for the research in the area of inspections and review [4]. Fagan review process consists of five well defined phases such as planning, overview, meeting, rework and follow-up. [10] [14] [15]. After Fagan's review process, Parnas & Weiss introduced active design review process in which their focus was to reduce the number of reviewers. [16] [7]. Bisant & Lyle introduced two person review process especially for small companies in which they tried to reduce the large consumption of resources i.e. the reviewers so they suggested, two members to participate in a review i.e. author and reviewer [17]. In the same year Humphrey introduced new process similar to the Fagan's review process having "analysis" phase in it, the purpose of this phase was the collection of identified defects. He had shifted the goal of preparation from basic understanding to the defect detection [18].

Martin & Tsai proposed effective review process for requirement phase of the critical projects. The main theme of the process was to have N-sub teams to review the requirement document [19]. Phased inspection was proposed by Knight & Myers to review the code [20]. Gilb & Graham also proposed a review process having a phase called "brainstorming", its purpose was to find root causes of detected defects. Another modification done by them was the introduction of entry phase at the start of the process to check the worth of the artifact to be reviewed [21]. In the same year the criticism on review processes was started with respect to review meetings then Votta proposed meeting less review process to reduce the meeting overhead cost [22]. Johnson proposed offline individual defect detection [23].

Sauer et al. proposed reengineered review process with phases like "discovery" for individual defect detection, "collection" for defect collection and "discrimination" for finding unique true defects list [24]. Paul proposed a scenario based review process to review the design artifact [25].

Laura suggested structured review process with the aim of reducing the interpersonal communication overhead [26]. Torgeir et al. introduced Postmortem review process for review process improvement [27]. Mishra & Mishra proposed simplified software review process based on formal technical asynchronous review process [28]. Effective hybrid review process was introduced by Sumaira et al. to reduce meeting overhead [13].

Mishra & Mishra proposed global software inspection method to reduce the communication and coordination problem of distributed team members [29]. Tejal & Bhavesh introduced software requirement inspection model to review the requirement documents [30]. A simplified model for software inspection was developed by misra et al. for small scale software industries with the aim to inspect the artifact at low cost [31]. Another approach requirement specification review was proposed by Li which use animation to guide the reviewers while reviewing the artifact [32].

From literature it is analyzed that all existing review processes ends-up with the closing phase [13] [33]. In closing phase the author modifies the artifact and the leader verifies the changes. It is also analyzed that the process does not closes until all the changes are made by the author, due to this if author have any query regarding identified defects and the leader is unable to answer that query then the defects in the artifact will propagate to next development phase and cost double to remove. The other issue identified is that the change – verification cycles moves between leader and author until all the changes are performed due to this if leader during follow-up phase encounter more unidentified defects then these defects will also propagate to next development phase. It is also analyzed from literature that existing review techniques have not provided any risk identification strategy to meet any uncertain issue. There is lack of guidance for tackling any uncertain situation, due to this If any uncertain situation occurs during the conduction of review like unavailability of reviewer, team composition, reviewer expertise then how that situation will be handled [33].

All above issues motivated us to propose such a review approach that allow the leader to analyze review related risk before the start of the review and also take decision to plan subsequent review cycle.

3. Proposed Risk Based Spiral Review Approach (RB-SRA)

Risk based spiral review approach proposed in this paper is for improving the software review process activities. The major distinctive feature of this approach is that it creates a risk-driven approach to the software review process. It integrates the strengths of spiral activities with the linear activities of the existing review processes. As the spiral activities are iterative in nature therefore it helps to detect and fix the defects in multiple iterations. The proposed approach shown in Figure 1 consists of four quadrant i.e. Review Planning and Risk Analysis, Review Execution, Rework/Defect Correction, Verification and planning for next spiral

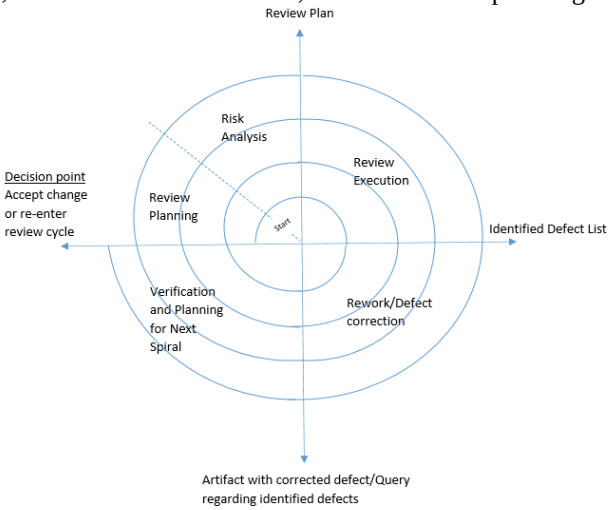

Figure 1: Proposed Risk Based Spiral Review Approach (RB-SRA)

3.1 Review Planning and Risk Analysis

The planning phase of the approach involves the review planning activities like preparation of artifact for review, selection of review team member, team composition, distribution/presentation of artifact, planning for review execution and identification of risk factors.

The Software Engineering Institute (SEI) defines risk as the probability of great loss. They should be well controlled in the software development projects [34]. The purpose of the risk analysis in reviews is to identify the associated risks that can affect the successful execution of the review process. The associated risks can be availability of the reviewers, reviewer's expertise, length of artifact to be reviewed etc [33]. Risk is a problem that may cause the failure of the review process. Risk analysis also helps to decide whether individual review, group review or just discussion with experts is required. Risks that can affect the results of the review process are given in Table 1.

Table 1: Expected Risk During The Conduction Of Review Process

Expected Risks
Review Team members availability
Review Team members at distributed location
Reviewer's expertise
Length of artifact to be reviewed
Understanding of review artifact
Review Team composition

Based on the risk analysis the alternative solutions to mitigate the identified risk must be defined. Figure 2 shows the input and output of the review planning and risk analysis quadrant.

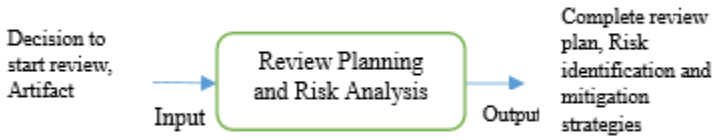

Figure 2: Input and Output of Review Planning and Risk Analysis Quadrant

3.2 Review Execution

The purpose of this phase is the actual execution of review process with the intent to find the defects. It involves the activities like individual review of the artifact, group review, discussion about defects and collection of defects. More details of the review activities can be found in [9]. The risk identified in the planning phase can affect the interaction mode of the reviewer in review execution phase. Like if review team members are distributed at different location than asynchronous tool assisted interaction must be performed. If the length of artifact is too large than the best strategy is to divide the artifact and conduct individual review of the allotted specific part of the artifact. Figure 3 shows the input and output of the review execution quadrant.

Figure. 3: Input and Output of Review Execution Quadra

3.3 Rework/Defect Correction

The purpose of this phase is actually to perform rework on the identified defects. The author makes changes to the artifact based on the identified defect lists. The author can query the leader if there is any issue regarding the understanding of the identified defect. Figure 4 shows the input and output of the rework/defect correction quadrant.

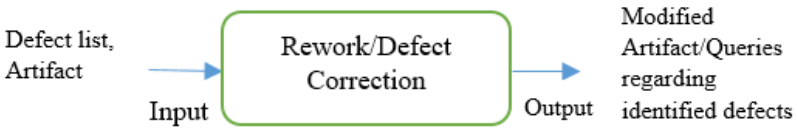

Figure. 4: Input and Output of Rework/Defect Correction Quadrant

3.4 Verification and Planning of Next Spiral.

The modified artifact is submitted by the author to the leader for verification. If all the defects are corrected then the leader closes the review process by formally signing off the review artifact. If some of the defects are not corrected by the author due to some misunderstanding or any other issue the leader can plan next review cycle if unable to answer that issue. The leader can also take decision of next review cycle if the leader finds that there are more unidentified defects in the artifact. Figure 5 shows the inputs and outputs of the verification and planning for next spiral quadrant.

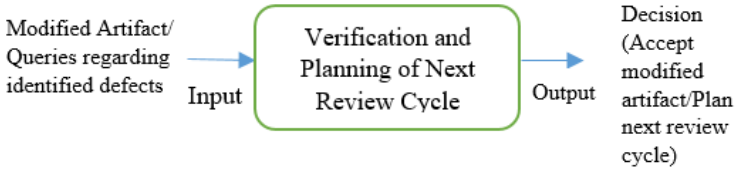

Figure 5: Inputs and Outputs of Verification and Planning of Next Review Spiral Quadra

4. Experimental evaluation of proposed review approach (RB-SRA)

Risk based spiral review approach is evaluated through experiment. The experiment evaluate the effectiveness of proposed review approach (RB-SRA) with the Fagan's inspection process.

4.1 Experiment Design

The experimental design consists of 12 subjects and two teams. Graduate level students of Software Engineering, National University of Modern languages are selected to perform the experiment.

Table 2: Experimental Design

Team #	Review Process	# of Subjects
Team 1	FRP	6
Team 2	RB-SRA	6

The subjects were given requirement document of Automated Teller Machine with 20 known defects. Table 2 demonstrate the experimental design.

4.2. Experiment Results

Results of experiment indicates that defects identified and corrected by Team 2 who executed review using iterative Risk-Based Spiral Review Approach (RB-SRA) is greater than Team 1 who had performed review with linear Fagan Review Process. Figure 6 shows the results of experimental evaluation of the defect detection effectiveness of both the approaches. The graph shows that Team 2 who have performed review with Risk-Based Spiral Review approach had detected large number of defects as compared to Team 1 who had performed review with Fagan Review Process.

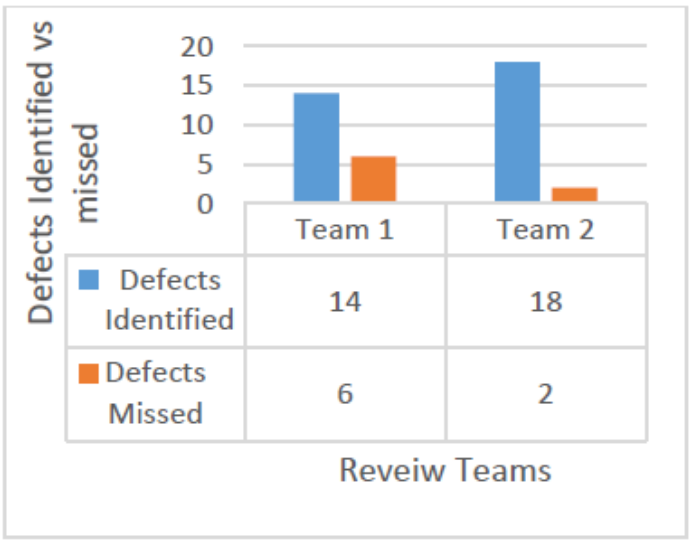

Figure 6: Comparison of defect detection between two teams

Figure 7 shows that Team 2 who had performed review with risk based spiral review approach had corrected maximum identified defects while Team 1 who applied Fagan review process could found only few defects so the approach appeared comparatively ineffective.

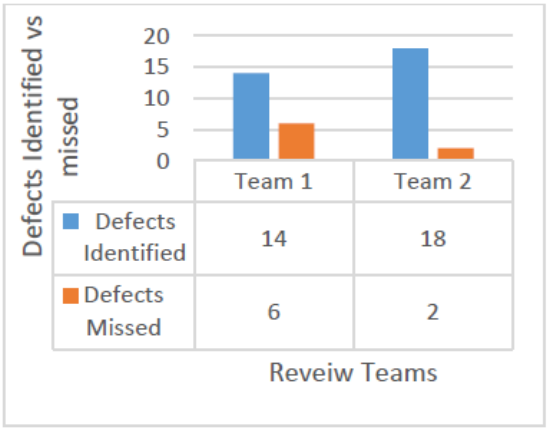

Figure 7: Comparison of defect detection between two teams

The outcome of the conducted experiment shows that the risk based spiral approach is more effective with reference to defect detection and correction than linear Fagan review process. The Iterative nature of the proposed approach is effective for achieving the quality goals.

5. Contribution and Significance.

The study has presented A Risk Based Spiral Review Approach that helps to handle two main issues of the review processes i.e. stagnation of the review process at the closing phase and mitigation of risk for the successful conduction of review process.

6. Conclusion and Future work suggestions

Software reviews are very important to ensure the quality of the software. This paper has contributed in the field of software quality assurance by introducing risk based spiral approach to review the artifacts developed in software development life cycle. This paper has provided comprehensive background of the existing software review process of last three decade. A risk based spiral strategy has been introduced with multiple benefits i.e. identification and mitigation of risks in planning phase that can negatively affect review process results. The introduced spiral strategy does not only help in effective defect detection but it also facilitates in correction of maximum possible defects. The proposed approach can be applied to any existing review process. The proposed approach is evaluated through controlled experiment. The experimental results verifies the effectiveness of the risk based spiral review approach in defect detection and defect correction. In future more experiments can be performed in multiple setting.

Acknowledgement

We wish to thank Engineering department, National University of Modern Language, Islamabad for providing support to our research work. We are also thankful to the valuable comments by our colleagues Ms. Farhat Sajida and Dr. Huma Hayat Khan as they have provided assistance and positive criticism that helps to conduct and improve the study.

References

[1]. S. J. Roohullah , S. T. U. Shah, Z. U. Johar, Y. Shah and F. Khan, "An Innovative Approach to Investigate Various Software Testing Techniques and Strategies," *International Journal of Scientific Research in Science, Engineering and Technology*, vol. 2, no. 2, pp. 682-689, March 2016.

[2]. P. Vitharana, "Defect propagation at the project-level: results and a post-hoc analysis on inspection efficiency," *Empir Software Eng*, p. 23, 20 November 2015.

[3]. G. Viviani and G. C. Murphy, "Removing Stagnation from Modern Code Review," in *SPLASH Companion'16*, 2016.

[4]. R. M. d. Mello, R. C. Motta and . G. H. Travassos, "A Checklist-Based Inspection Technique for Business Process Models," 2016.

[5]. C. M, L. O and B. S, "Software reviews: The state of teh practice," vol. 20, no. 6, pp. 46-51, 2003.
[6]. B. H. Barhate , "Software Inspection Improves Quality of Software Product," *International Journal of Computer Science & Informatics*, vol. 2, no. 1, pp. 44-48, 2016.
[7]. A. Aurum, H. Petersson and C. Wohlin, "State-of-the-art: software inspections after 25 years," *Software Testing, Verification and Reliability*, vol. 12, no. 3, pp. 133-154, September 2002.
[8]. F. Shull and . C. Seaman, "nspecting the History of Inspections: An Example of Evidence-Based Technology Diffusion," *IEEE Software*, vol. 25, no. 1, pp. 88-90, 2008.
[9]. P. Thongtanunam, "Studying Reviewer Selection and Involvement in Modern Code Review Processes," 2016.
[10]. M. Fagan, "Design and code inspections to reduce errors in program development," *IBM System Journal*, vol. 15, no. 3, pp. 182-211, 1976.
[11]. M. S. Iqbal, M. Sadiq and T. Khan, "Framework for the Development of Automated Inspection Tools," *International Journal of Computer Science and Network Security*, vol. 16, no. 1, January 2016.
[12]. A. Egorov , *Tool Support for Inspections*, 2013.
[13]. S. Nazir, N. Fatima and S. Malik, "Effective Hybrid Review Proess," in *International Conference Computer Science and Software Engineering Wuhan, China*, Wuhan, China, 2008.
[14]. P. Thongtanunam, "Studying Reviewer Selection and Involvement in Modern Code Review Processes," 2016.
[15]. E. P. Doolan, "Experience with Fagan's Inspection Method," *SoftwarePractice And Experience*, vol. 22, no. 3, pp. 173-182, 1992.
[16]. D. L. Parnas and D. M. Weiss, "Active Design Reviews: Principles and Practices," in *Proceedings of ICSE'85*, 1985.
[17]. D. B. Bisant and J. R. Lyle, "A Two-Person Inspection Method to Improve Programming Productivity," *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, vol. 15, no. 10, pp. 1294-1304, October 1989.
[18]. W. S. Humphrey, Managing the software process, Addison-Wesley Longman Publishing Co., Inc, 1989, p. 494.
[19]. J. Martin and W. T. Tsai, "N-fold Inspection: A Requirements Analysis Technique," *Communications of the ACM*, vol. 33, no. 2, pp. 225-232, 1990.
[20]. J. C. Knight and E. A. Meyers, "An Improved Inspection Technique," *Communications of the ACM*, vol. 36, no. 11, pp. 51-61, 1993.
[21]. T. Gilb and D. Graham, Software Inspection, Addison Wesley, 1993, p. 471.
[22]. L. G. Votta, "Does Every Inspection Need a Meeting?," vol. 18, pp. 107-114, 1993.
[23]. P. M. Johnson, "An Instrumented Approach to Improving Software Quality through Formal Technical Review," in *Proceedings of 16th International Conference on Software Engineering*, Sorento , Itly, 1994.
[24]. C. Sauer, D. R. Jeffery, L. Land and P. Yetton, "The Effectiveness of Software Development Technical Reviews: A Behaviorally Motivated Program of Research," *IEEE Transactions on Software Engineering*, vol. 26, no. 1, pp. 1-14, 2000.
[25]. C. C. Paul, "Active Reviews for Intermediate Designs," 2000.
[26]. F. Laura, "Applying Small-Investment, High-Return Review Techniques for Fast-Paced Teams," 2011.
[27]. D. Torgeir , D. T. T. Stalhane and N. M. Brede, "Light Weight Postmortem Reviews," 2002.
[28]. D. Mishra and A. Mishra, "Efficient software review process for small and medium enterprises," *IET Software*, vol. 1, no. 4, pp. 132-142, 2007.
[29]. D. Mishra and A. Mishra, "A Software Inspection Process for Globally Distributed Teams," pp. 289-296, 2010.
[30]. G. Tejal and C. Bhavesh, "Software Requirement Inspection Tool for Requirement Review," in *International Conference on Advances in Computing and Management*, 2012.
[31]. S. Misra, . L. Fernández and . R. C. Palacios, "A simplified model for software inspection," *JOURNAL OF SOFTWARE: EVOLUTION AND PROCESS*, vol. 26, pp. 1297-1315, 2014.
[32]. M. Li , "A Systematic Inspection Approach to Verifying and Validating Formal Specifications based on Specification Animation and Traceability," 2014.
[33]. D. Winkler, J. Musil and A. Musil, "Collective Intelligence-Based Quality Assurance: Combining Inspection and Risk Assessment to Support Process Improvement in Multi-Disciplinary Engineering," Springer, 2016.
[34]. M. F. Rabbi and K. O. B. Mannan, "A Short Review for Selecting the Best Tools and Techniques to Perform Software Risk Management," *European Journal of Advances in Engineering and Technology*, vol. 3, no. 6, pp. 1-7, 2016.