

Collaborative requirements review

Azlin Nordin¹, Nadzurah Zainal Abidin², Sara Hanum Mohd Zaini³

¹Department of Computer Science, Kulliyah of Information and Communication Technology
International Islamic University Malaysia
50728 Gombak, Kuala Lumpur, Malaysia

*Corresponding author E-mail : azlinnordin@iiu.edu.my

Abstract

Requirements review is a formal review conducted to ensure that system requirements have been completely and clearly identified. In the conventional requirements review technique, reviewers are required to physically attend the review session and give their review feedback during the session. In such a situation, there are potential problems of scheduling the review session based on the availability of the reviewers and issues with having to physically attend the entire review session. Furthermore, the review session needs to be manually organized by the review leader and the outcome of the session need to be manually compiled. In some occasions, more than one sessions are required to be organized. Hence, the objectives of this project are to: (1) create means for the reviewers to perform review anytime and anywhere; (2) facilitate collaborative review session; (3) support checklist management for the review guidance; and (4) allow compilation of the review feedback to be generated. As a case study, a web application for collaborative requirements review has been developed and tested. Finally, the features of the application are tested and issues are documented.

Keywords: : Requirements Review; Requirements Validation; Collaborative Review.

1. Introduction

In general, Requirements Engineering (RE) is concerned with understanding domain knowledge and producing Software Requirements Specification (SRS) for the system to be built. RE normally consists of five main activities i.e. (1) elicitation, (2) analysis and negotiations, (3) documentation, (4) validation and (5) management or evolution [1]. Requirements review is performed in the requirement validation activity, which mostly concerned with checking the quality of the requirements document. In short, it is concerned with answering the question of ‘have we got the requirements right?’ [2].

Based on IEEE Recommended Practice for Software Requirements Specification (SRS) [3], a good SRS should be correct, unambiguous, complete, consistent, ranked, verifiable, modifiable and traceable. In addition to that, a mapping study on empirical evaluation of SRS indicated that among the 31 aspects being investigated, the top-six most studied aspects are understandability, efficiency, correctness, defect rate, completeness and consistency [4].

Based on IEEE Standard for Software Review and Audit [5], there are five types of software reviews i.e. management review, technical review, walkthrough, inspection and audit. These reviews can be performed in any stage of the software development life cycle (e.g. requirements, design, implementation, testing) or for any software work products (e.g. requirements specification, design models, coding, test cases). Requirements review can be considered as technical review technique where a group of stakeholders check the requirements document. According to Sulehri [6], requirements review is the most widely used as a requirement validation technique.

There are three types of reviewing process i.e. 1) *free mode*, where there is no specific directive on how to perform the review, basi-

cally based on reviewer’s initiatives and expertise; 2) *checklist-based*, where the review process is done based on a predefined list of criteria as a guidance; and 3) *process-based*, where reviewers are expected to find defects based on different perspectives [7].

In general, the requirements review process can be summarized and visualized as shown in Figure 1 [2]. During the initial step, the review leader plans for the review by selecting the review team members, setting the time and venue. Next stage is followed by distributing documents, where requirements document and any relevant documents are distributed to the review team members.

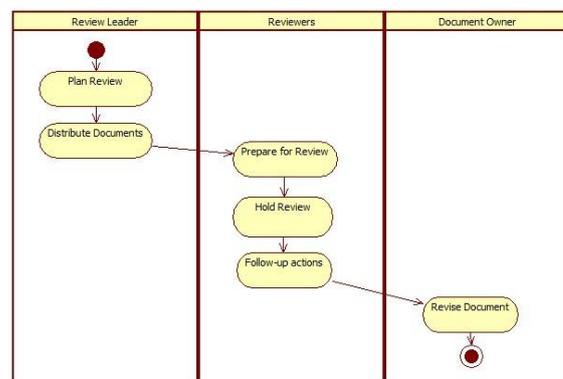


Fig. 1: Requirements review process

During the preparation for the review process, each reviewer is normally expected to inspect the requirements document to identify conflicts, omissions, inconsistencies, deviations from standards

and any other problems based on the team leader's decision. During the review session, the reviewer stakeholders (1) read and analyze requirements, (2) look for problems, (3) meet and discuss the problems and (4) agree on a set of actions to address the identified problems [7]. All the individual comments, problems and set of action are discussed and agreed in the review meeting. The process continues with implementing the agreed actions and documenting them. In the end, the reviewed document will be passed to the document owner to be revised.

The Standish Group Chaos Report stated that 12.3% of a failure software development comes from the incompleteness of requirements and specifications [8]. Moreover, the report also highlighted three top reasons for the success of software projects are: (1) user involvement, (2) executive management support and (3) clear statement of requirements. These show that quality of requirements gives a significant impact to the quality of system being built.

The importance of requirements review has been widely acknowledged in the literature. The technique can be useful to improve the quality of SRS by discovering defects at the early stage of software development process. Nonetheless, despite the importance of requirements review, there are very little published work, which focuses specifically on requirements review.

Some of the issues and challenges of requirements review in the industry are issues with the review process itself, reviewer team, requirements specification, support tool, organizational culture and governance[9]. From the context of the review process, it is well acknowledged that reviewers had lack of guidance and direction in order to discover errors. Wiegers and Beatty [10] stated that the challenges during review process is due to lack of a standard checklist to be used as a guidance.

Whilst, from the perspective of a review team leader, it is a challenging task to organize and facilitate the validation of requirements in an efficient and effective manner [4]. In the conventional way, requirements reviews are organized by the review leader. During the review, all the review team members are required to attend the session to give feedback on the requirements document. The challenge with such a scenario is to ensure the availability of the reviewers to attend and participate during the whole review session[11].

In addition, the review process is normally ineffective due to reasons such as it is not held when intended, not achieve the expectations, reviewers only take superficial look and miss major defects of the reviewed document[12]. Review session has always been considered as expensive due to the time and effort required to plan and execute the session [13].

The main objective of this work is to provide means to allow requirements review sessions to be more flexible and efficient. This is even much more crucial as detecting errors at early stage of software development can prevent the errors from propagating to later stages [7]. Eventually, this strategy can help to reduce the review cost and assist the development team to achieve better software quality.

2. Related Work

In Software Engineering (SE), the early step towards a successful project development is defining project requirements, while the deliverable is a complete SRS. SRS is the most important artefact to be referred to for every project team member. In this project, we analyzed the existing requirements review tools based on its features. Based on the analysis, all the tools' features were listed in Table 1.

Table 1: Analysis of Requirements Review

FEATURES	STAND ALONE REVIEW TOOL				REQUIREMENTS MANAGEMENT TOOL	
	PLEASE REVIEW	PROOF HQ	FRONTIER	SDL.COM	SPARX	INNO-SLATE
Upload review artifacts	Y	Y	Y	Y	N	N
Checklist management	N	N	Y	Y	Y	Y
Comments on requirements	Y	Y	Y	Y	Y	Y
Check requirements quality	N	N	N	N	N	P ¹
Edit Requirements	N	N	N	Y	Y	Y
Insert multimedia & videos	N	Y	Y	Y	N	Y
Deadline Reminder	Y	N	Y	Y	N	N
Collaborative	Y	N	Y	Y	Y	N
Generate report	Y	Y	Y	N	Y	Y

Y=yes N=No P=Partial

We grouped the tools into two categories i.e. (1) stand-alone review tool and (2) requirements management tool. The first category i.e. *stand-alone*, defines tools that are designed to support generic requirements reviews, whereas the second category consists of requirements management tools that embed requirements review as a part of its features.

The main investigated features are upload review artifacts, checklist management, review requirement, track changes, edit requirement, collaborative, and generate review report. For the stand alone review tool category, all the investigated tools allow users to upload review materials. In contrast, in the other category, the same feature is not required as the requirements are stored and managed in the tools. The derivation of requirements for review purpose will be extracted from the requirement repository.

In order to organize review sessions, the review leader will normally decide on the review criteria. By doing so, review activity will be more focused and directed. To support this, checklist management feature could assist the leader to decide which criteria to be included or to be excluded for the review activity. Among the investigated tools, only *PleaseReview* and *ProofHQ* did not support this feature[14], [15]. As mentioned in the earlier section, without the checklist, the reviewers may have issues to perform the review process efficiently.

In *Sparx* and *SDL.com*, the upload SRS feature is not applicable as the tools allow requirements to be added and managed in the tool. Hence, the SRS can be generated from the stored requirements.

Normally, during requirements review, reviewers are not expected to edit requirements. That is why we can observe that all the standalone review tools do not support this feature. We believe that the editing requirement feature is a part of the RM tool and not a significant feature in review tool.

Based on the comparison in Table 1, we further analyzed the two review tools from the stand-alone category. This is because, these tools are in line with our product values. The chosen tools are *PleaseReview* [14] and *ProofHQ* [15]. None of the tools is observed to have complete features as recommended by the conventional requirements review. Nonetheless, *Frontier* [16] will not be discussed in detailed since the tool focuses on peer review technique.

2.1. PleaseReview

PleaseReview simplifies the review process by providing the necessary tools and functionalities such as collaborative review environment, review control, management and reporting. *PleaseReview* enables parallel and simultaneous detailed review by multiple reviewers and also caters for off-line review. Besides, it also allows the reviewers to make comments and changes within the document, and see and reply to each other's feedback as can be shown in Figure 2. *PleaseReview* handles the document review with Word document, PDFs, Excel, PowerPoint, images and text file formats.

In addition, PleaseReview also provides automatic email reminder on due date of a document review. Review owner can easily assimilate and view all the proposed changes and comments and decide to accept, consolidate or reject. PleaseReview also offers a comprehensive reconciliation report, which acts as an audit trail and valuable record of the review. Besides, PleaseReview provides complete status information at all times detailing user progress.

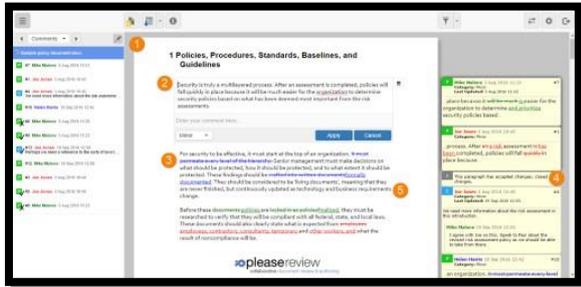


Fig. 2 : PleaseReview

2.2. ProofHQ

ProofHQ provides a collaboration in real time. Users are allowed to give feedback and comment directly onto the document to be reviewed. Comments become alive as real-time discussion threads, allowing for specific requirement. See Figure 3. ProofHQ also allows intuitive commenting and annotation tools on over 150 file types, including print, web, interactive, email, and video.



Fig. 3: ProofHQ

Besides, ProofHQ enables the user to perform the review process anywhere and anytime on any selected mobile platform. Users can experience the efficiency of ProofHQ review and get approval from any mobile devices.

Although both the tools allow the relevant features to accommodate and support the review process, both the tools allow comments to be made based on the document as a whole, and not for each individual requirement. Comments can be linked to a specific requirement as annotation, however, each requirement is not a focus for the review. We foresee that by allowing each requirement to be as an entity of the review activity, reviewers will be

able to justify their feedback based on specific checklist feature and later on changes to the document can be reflected in the next review cycle. Hence, this work proposes on providing users with the ability to support and conduct requirements review activities in a more efficient and flexible manner.

3. Methodology

Based on the analysis, the required features were gathered to be developed as a web application in order to support the requirements review process. First, we gathered and listed all the 29 requirements. These requirements were then summarized and represented in a use case diagram. See Figure 4.

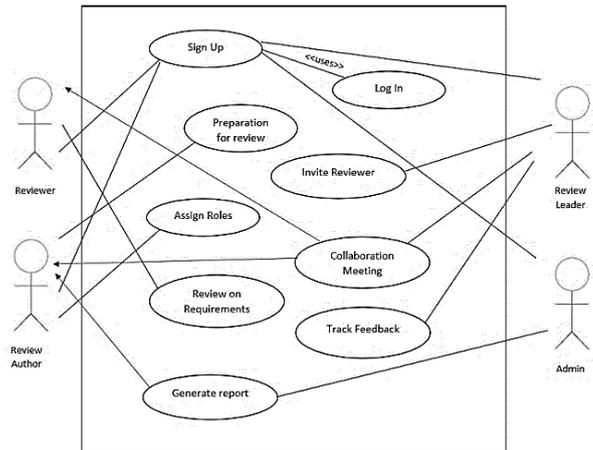


Fig 4: Collaborative Requirements Review

The figure shows that there are four (4) main types of actors involve in the project i.e. reviewer, review author (owner), review leader and administrator. Selecting the right team for requirement review process is a very crucial task. Ideally, the project review leader may propose reviewers that are not necessarily from the same background. The review team should include people from different backgrounds that are responsible on system design, implementation and requirements engineers.

The requirements were grouped into nine use cases that are sign up, login, preparation for review, invite reviewer, assign roles, collaboration meeting, review on requirements, track feedback and generate report. The detailed of the requirements are provided in a technical report. The review process begins when the Review Author uploads the SRS document into the system. The Review Author should be able to set up the review details with the file name, a short description of the review and desired completion date. See Figure 5.

Then, the Review Author selects the review team and specify each team member's role. The Review Leader will then invite reviewers for a specific review session. As for the reviewers, they will receive an invitation from the Review Leader. The Reviewers can review the requirements based on the provided checklist and they can provide feedback on the requirements. Conventionally, the checklist is distributed to all review team members to use as a guidance while reviewing the requirements. See Figure 6.

Fig. 5: User interface to Add New Project

An alternative way we proposed is the Review Leader should be able to manage and define the checklist prior to inviting the review team. By doing so, it can avoid the need to memorize all the items to look for in the document.

Fig. 6 : Review Requirements based on Checklist

Next, all the feedback made will be stored and can be tracked later by the Review Leader. Once finished, the Review Leader is able to display all the review activities. In addition, the Review Leader will be able to accept or reject any purposed changes in the SRS document.

Review leaders are able to generate report on the review activity. This report could assist the document owner to work on the review feedback. By doing so, it could help the team to improve the SRS quality. See Figure 7.

Notification Button Clicked (3) - Generate Report

Requirement Number	Requirement	New Requirement
1.1	Requirement	Feedback
		Feedback
1.2	Requirement	Feedback
1.3	Requirement	Feedback
		Feedback
1.4	Requirement	Feedback
		Feedback
		Feedback

Fig. 7: User Interface to Generate Report

A few test cases were generated and executed to test the requirements. The implementation of this project is using the basic JavaScript, HTML, CSS and MySQL.

4. Conclusion

Requirements review is crucial in any RE project as to assure software quality. In this work, in addition to all the basic needs of the requirements review process, we provide a platform to support

and facilitate requirements review by allowing collaboration, providing check list and also managing for the preparation of the review process. In the future, we seek to integrate the review process with Natural Language Processing (NLP) technique to venture into semi-automated review process.

Acknowledgement

This work is sponsored by Research Initiative Grant (RIGS-16-344-0508). We would also like to thank the KICT faculty and IIUM for the opportunity to conduct this research.

References

- [1] I. Sommerville, *Software Engineering*, 10th ed. Boston: Pearson, 2015.
- [2] G. Kotonya and I. Sommerville, *Requirement Engineering: Process and Techniques*. John Wiley & Son, 1998.
- [3] "IEEE Recommended Practice for Software Requirements Specifications," *IEEE Std 830-1998*, pp. 1–40, Oct. 1998.
- [4] N. Condori-Fernandez, S. España, K. Sikkil, M. Daneva, and A. González, "Analyzing the Effect of the Collaborative Interactions on Performance of Requirements Validation," in *Proceedings of the 20th International Working Conference on Requirements Engineering: Foundation for Software Quality - Volume 8396*, New York, NY, USA, 2014, pp. 216–231.
- [5] "IEEE Standard for Software Reviews and Audits," *IEEE Std 1028-2008*, pp. 1–52, Aug. 2008.
- [6] L. Sulehri, *Comparative Selection of Requirements Validation Techniques Based on Industrial Survey*. 2010.
- [7] K. Pohl, *Requirements Engineering: Fundamentals, Principles, and Techniques*, 1st ed. Springer Publishing Company, Incorporated, 2010.
- [8] "Project Management White Papers | Project Smart." [Online]. Available: <https://www.projectsmart.co.uk/white-papers/chaos-report.pdf>. [Accessed: 18-Aug-2017].
- [9] S. O. Mokhtar, R. Nordin, Z. A. Aziz, and R. M. Rawi, "Issues and Challenges of Requirements Review in the Industry," *Indian J. Sci. Technol.*, vol. 10, no. 3, Jan. 2017.
- [10] K. Wiegers and J. Beatty, *Software Requirements*, 3rd ed. Redmond, Washington: Microsoft Press, 2013.
- [11] S. B. Saqi and S. Ahmed, *Requirements Validation Techniques practiced in industry: Studies of six companies*. 2008.
- [12] K. Wiegers, *More About Software Requirements: Thorny Issues and Practical Advice*. Microsoft Press, 2005.
- [13] Y. K. Lee, H. P. In, and R. Kazman, "Customer Requirements Validation Method Based on Mental Models," in *2014 21st Asia-Pacific Software Engineering Conference*, 2014, vol. 1, pp. 199–206.
- [14] "PleaseReview: collaborative document review, editing and co-authoring." [Online]. Available: <http://www.pleasetech.com/pleasereview.aspx>. [Accessed: 02-May-2017].
- [15] "Online Proofing Software Tools | Review & Approval ProofHQ." [Online]. Available: <https://www.proofhq.com/>. [Accessed: 02-May-2017].
- [16] "About Frontiers | Academic Journals and Research Community." [Online]. Available: <http://home.frontiersin.org/about/review-system>. [Accessed: 02-May-2017].