International Journal on Perceptive and Cognitive Computing (IJPCC) Vol 9, Issue 1 (2023)
https://doi.org/10.31436/ijpcc.v9i1.375

Design and Development of the e-Review:
An Online Requirements Review Tool based on Reading Techniques

Nordin, Azlin*, Mohammad Musleh, Islah[1] and Ro Zac, Abdul

Department of Computer Science (DCS), Kulliyyah of Information and Communication Technology (KICT), International Islamic University Malaysia (IIUM), 53100 Gombak Kuala Lumpur, Malaysia,

*Corresponding author: azlinnordin@iium.edu.my
(Received: 8th January 2022; Accepted: 19th January 2023; Published on-line: 28th January 20230

Abstract— Requirements review is a formal review process in which several reviewers read all or part of software requirement specification (SRS) and search for defects. In order to ensure that the system requirements have been completely and clearly identified, reviewers use many different review techniques. Despite of the availability of many reading techniques such as perspective-based reading techniques, checklist-based reading techniques and a few more, we could not find any existing tool to support two or more reading techniques in order to guide reviewer while reviewing a document. Besides, currently, in order for the reviewers to review an SRS document, they need to gather physically, make a session and give their review feedback during the session. In such cases, the requirement engineers need to schedule the review session based on the availability of the reviewers. The review leader needs to manually organize the session and review outcomes also need to be manually consolidated. Thus, the objectives of this project are to develop a web-based tool in order to; (1) enable reviewers to review SRS document using two different reviewing techniques such as checklist-based reading (CBR) and perspective-based reading (PBR); (2) enable review leader to assign reviewer for requirement review; and (3) allow compilation of the review feedback to be generated. The project applied Iterative process model to manage the software development process. The implementation of the project is using PHP JavaScript, HTML, CSS, Bootstrap, JQuery, and Ajax.

Keywords— requirements validation, requirements error, requirements review, reading techniques, checklist-based reading, perspective-based reading

I. INTRODUCTION

Requirements engineering (RE) is the process of gathering requirements for a system, analyze and document them. The importance of RE in software development has been stated in many literatures including in [1]–[5]. In general, RE consists of five main activities that are (1) elicitation, (2) analysis and negotiations, (3) documentation and (4) validation and (5) management [6]. The requirements validation in RE activity is concerned with checking the requirements document for any requirements error such as consistency, completeness, and accuracy, in order to ensure that the requirements document defines the stakeholders' requirements.

One of the requirement validation techniques is requirements review, where a group of stakeholders checks the requirements document. In order to perform validation, the review 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. In addition, a review process is also considered as a learning and sharing knowledge instead of focusing on the main objective of the review process, which is finding faults [7].

Despite of the importance of requirements review, there has been insufficient published work, which specifically focus on requirements review and requirements review reading techniques. Based on our research, there is a number of requirements review reading techniques that have been used for detecting defects in the requirements review activity such as checklist-based reading, defect-based reading, perspective-based reading, usage-based reading, and scenario-based reading. In addition, in order to conduct the requirements review session, the organizer has to manually allocate time and check on the availability of each reviewers, which may takes a lot of effort and time. In addition, each reviewers' availability may change due to other appointments or other important events. Hence, re-arranging dates will also require more effort and time to the process.

The main objectives of this project are to design and develop a web-based requirements review tool in order to enable the stakeholders to review Software Requirements Specification (SRS) document using a combination of requirements reading techniques i.e., checklist-based

https://doi.org/10.31436/ijpcc.v9i1.375

reading (CBR) and perspective-based reading (PBR) techniques. In CBR technique, the reviewer gets a checklist that is expressed in the form of questions or statements in order to search for a special kind of faults in the specification [7]. The PBR technique concentrates on examining requirements from different perspectives of the users of software documents in order to improve efficiency by minimizing the overlaps among the faults found by the reviewers [7]. Both techniques use checklist, which is created and managed by the review leader.

The benefit of using multiple reading techniques for requirements review is that each technique can assist in identifying defects based on defect types and perspectives. Hence, a combination of reading techniques is proposed in this work to review a requirements document in order to further improve the defect detection in requirements. Furthermore, handling all the review reviews during validation is a time-consuming process that necessitates a significant amount of effort and time. Hence, it is evidence that the conventional (manual) requirements review process is a very challenging task for the review team especially the review leader. Therefore, an online requirement review tool shall be able to (1) assist in validating requirements documents, (2) help in systematically managing the requirements review feedback from different reviewers, and (3) provide a mechanism to allow multiple reading techniques to be adopted in a requirements review session.

This paper begins with an explanation of requirements review methods and then moves on to the research methodology. The findings and discussion will be discussed after that, and eventually, the conclusion and future work are presented at the end.

II. Review of Previous Work

According to the IEEE Standard for Software Review and Audit, there are different types of requirements validation techniques such as 1) technical review 2) management review 3) walkthrough, 4) review, and 5) audit. Each type of the reviews can be carried out at any stage of the software development life cycle such as requirements, design, implementation. In management review for example, the reviewers check the project progress, identify the status of schedules, and plans as well as confirm requirements, and system allocation for the requirements. The aim of technical review is to evaluate a software product by a qualified personal team to find out its suitability for its usage and determine differences from standard and specification. Inspection aims at identifying and detecting faults in a software product. Furthermore, systematic walk-through is used for evaluating software product. It can also be held to educate an audience about a software product. Besides, software audit is used to evaluate a software products independently (IEEE Std, 1997). In the requirements review, several stakeholders have a look at the requirements documentation which is considered as a technical review.

One of the most frequently used requirements validation techniques is requirements review [8]. In requirements review, the review stakeholders go through the requirements, search for any requirements issues, gather and discuss the found issues and agree upon a set of actions to resolve the detected issues [9]. Furthermore, it is stated that a more effective requirements review and defect identification adopt several requirements reading techniques such as a combination of defect-based reading, checklist-based reading, perspective-based reading, usage-based reading, and scenario-based reading[10]. The aim of this study are to (1) apply the Checklist-based Reading (CBR) and the Perspective-based Reading (PBR) reading techniques, (2) design and develop a requirements review tool support. The reason for choosing these two techniques is because the CBR and PBR are the best-known reading techniques among others[11].

In CBR, a list of various question-based items is provided to use for reviewing requirements. Team members need to address these items that are linked to the consistency of the requirements specifications. These reading techniques are very useful for reviewers to remove loop entire requirements specification by addressing the numerous items which they missed during the review process [8].

PBR provides instruction to assist reviewers in following the view of the main stakeholders of the item under review. Tester (T), Designer (D), and End-user (U) are identified as the main perspectives. Reviewers create abstractions that are important to their point of view. For example, a designer generates high-level preliminary diagrams, a set of test cases is created by tester, and a user generates a series of use cases. Reviewers use a variety of questions to help spot flaws when constructing abstractions. The items are normally on a standard categories' basis of defects. This is not a static collection of types of defects and can be modified as needed [11].

A. *Existing tool*

The Quality Analyzer for Requirement Specifications (QuARS) tool was designed to analyze, and validate NLR in an automated systematic way[12]. Another tool i.e., the Automated Requirements Measurements (ARM) evaluates natural language requirements (NLR) during the early software development life cycle. It was created by the Goddard Space Flight Center's (GSFC) Software Assurance Technology Center (SATC) [13]. The tool searches for each of

https://doi.org/10.31436/ijpcc.v9i1.375

the quality primitives in the requirements document that was established by SATC including completeness, correctness, ranked, unambiguous, consistent, customizable, traceable, and verifiable. In [14], the authors stated that the implementation of automation tool in requirements defects detection is scarce and many still rely on manual way of conducting reviews. However, these automated tools rely on the automation by the tool itself rather than being built to complement the requirements review activity, and requirements reading techniques.

III. Methodology

In this project, the iterative process model was adopted for managing the project development (see Fig. 1). The iterative model is well suited to this project, particularly in terms of tracking progress and creating some working functionalities in the early project life cycle. The iterative process starts with a simple implementation of a small set of the software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed. This makes it easier to find functional or design flaws. Furthermore, making adjustments in any step of our project life cycle is less expensive because each iteration moves through all phases from planning to evaluation before the whole system is ready to be deployed. Testing and evaluation processes are less complicated as the processes are focused for smaller iterations.

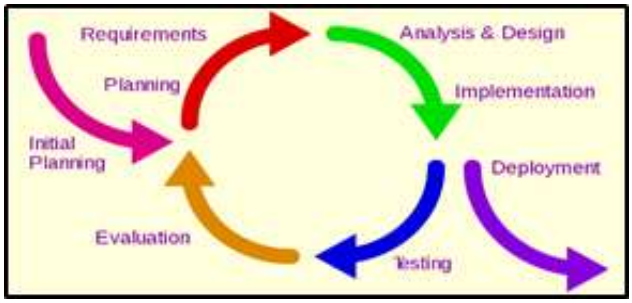


Fig. 1 Iterative Software Process Model

A. *Requirements Engineering Activities*

This section includes the requirements engineering phase for the e-Review tool project. In this section, the system features and its flows will be described by using a use-case diagram (see Fig. 2) and an activity diagram (see Fig. 3) respectively.

1. Use-case diagram

In Fig. 2, the use case diagram depicts the four e-Review actors that are (1) reviewer, (2) review leader, (3) document author, and (3) administrator. The reviewer is the person who will be directly involved in a review session by detecting errors in the requirements document and providing feedback on any requirements or documentation errors. The review leader is the person who will be assigning the reviewers with specific reading techniques to be applied during the review session.

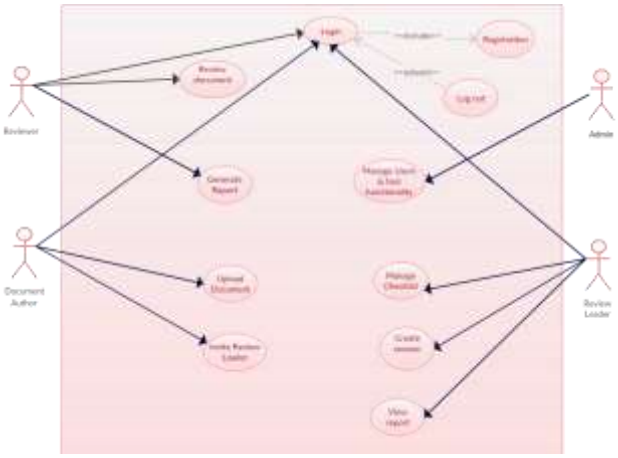


Fig. 2 Use case diagram for e-Review

The review leader is also responsible for managing the checklist to be used as a guide during the review session. In addition, the document author is the owner of the document to be reviewed, whilst the administrator has the roles of managing user access, or anything related to the system and not directly involved in the review process.

Fig. 2 depicts the interactions between reviewer and review leader, review leader and document author, relationships between the features and the system boundary. For each use case, a use case specification was created to provide the detailed scenario for each of the main features including its normal, alternative and exception flows. The completed use case specifications were included in the final report of the project documentation and will not be included in this paper.

2. Activity Diagram

Selecting the right team for requirement review process is very crucial. Ideally, a reviewer may not necessarily be from the same background discipline. The review team is recommended to involve people from different backgrounds and roles in the development such as system designers, programmers, software testers and also requirements engineers. In the activity diagram (see Fig. 3), a reviewer can sign up an account as well as login to the system. The main activities of a review are to review requirements and requirements document, and also to generate the review report.

The review process begins when the document author creates a project by uploading the SRS and invite the review

https://doi.org/10.31436/ijpcc.v9i1.375

leader to the specific project. The review leader is then added to the project, and he can invite members of the system as reviewers.

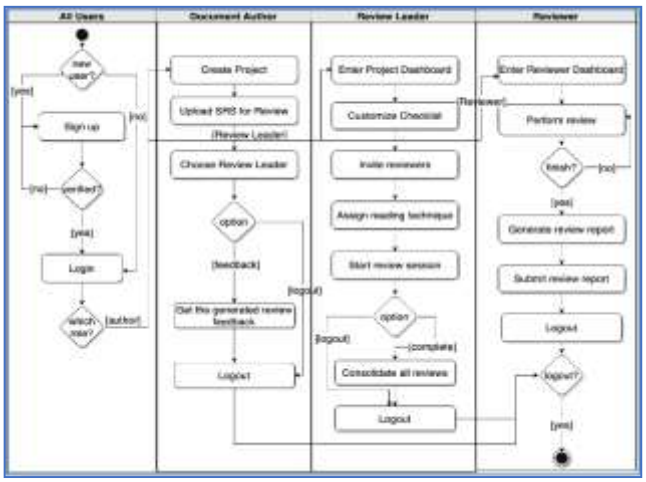


Fig. 3 Activity Diagram

The reviewers begin to review the document, generate the review report then send to the review leader. The review leader then reviews the feedback and decides whether the feedback is approved or rejected. After reviewing the feedback, the review leader also has to generate an overall report, which consolidates all the approved feedback and send the report to the document author.

B. Design Activities

This section includes the design phases for project. In this phase, we adopted the three-tier architecture model as shown in Fig. 4. The user interfaces were designed so that all the requirements for the system can be visually represented and validated by the stakeholders to ensure that there are no missing requirements in the development.

1. Architecture Diagram

In Fig. 4, the layered diagram shows the architecture of the e-review tool. The advantage of having this architecture is that the user interface, processes, and storage are separated components. Hence, they are easier to be organized during development because each layer serves a dedicated service i.e., the User Interface (UI) components serve as the interface layer, the middle layer functions as the service layer, and the bottom layer represents the data layer. In addition, by using the Multi-View-Controller(MVC) layers, the development artefacts are less complicated to be maintained for any changes during the development and also in the future.

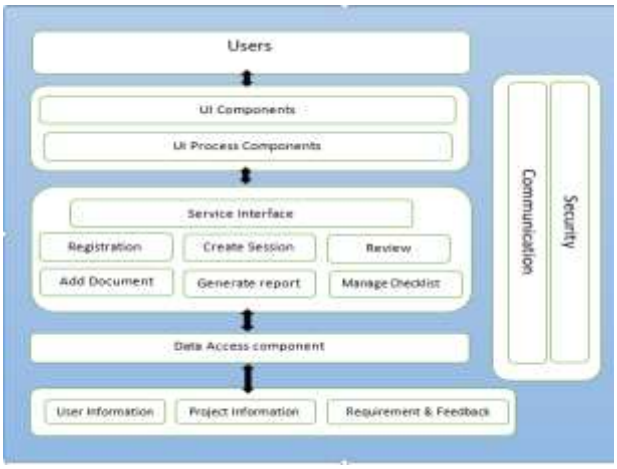


Fig. 4 Architecture diagram for e-Review

2. User Interface Design

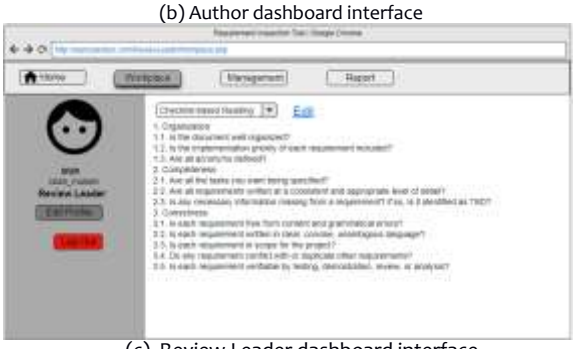
This section contains the user interface design activity. Based on the requirements that had been elicited and analyzed during requirements engineering stage, we created the user interface design of the project using the Mockups tool, which can be accessible from moqups.com. Due to space constraint, we selected only the most significant designs in this section.

Fig. 5(a) indicates the first page whenever a user logs in to the system as the role of a reviewer. They can edit their own profile, see the projects information and access to the specific project. The following design depicted in Fig. 5(b) belongs to the document author. A document author prepare the document to be reviewed i.e. in this case the SRS document. The document author can edit their profile, add new project, and see the project information such as project title and due date of a specific project. For the review leader interface (see Fig. 5(c)), the review leader can view the organization of his team, number of team member as well as he can invite more reviewers to the project. He can also edit the items in the checklist or in the PBR view that will be used in the requirements validation activity.

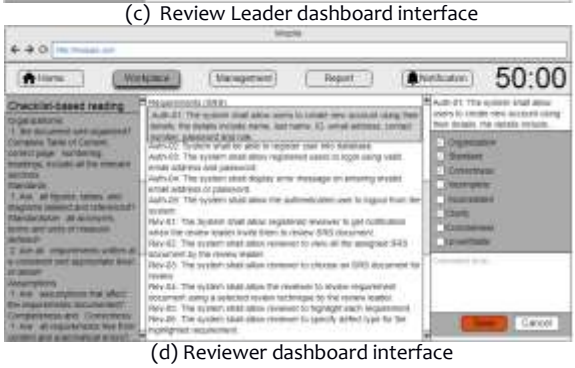
The interface design as depicted in Fig. 5(d) belongs to reviewers perspective, where the review process will be executed. In this case, the reviewer is assigned to review the SRS with checklist-based reading technique.

(a) Reviewer dashboard interface

(b) Author dashboard interface



(c) Review Leader dashboard interface



(d) Reviewer dashboard interface

Fig. 5 User Interface Design

C. Implementation

This e-Review tool has been developed in a web-based environment using PHP, JavaScript, JQuery, Ajax, HTML, CSS, mysql and Bootstrap.

Other than that, we adopted PHP and MySQL as an extension and database platform for the development implementation. We also applied the PHPMyAdmin and as an alternative tool to support the process of connecting the data layer to the system. Finally, on the security side, the administrator manages validation for the login and signup functions, which helps the system to be secured.

D. Testing

Based on the iterative model, testing was conducted to detect defects and errors in the online requirements reviewing tool. In order to conduct the testing, a set of test cases was created to be executed. The test was due to determine whether all requirements of the system have been implemented correctly and have met the main goal that is to develop online requirements reviewing tool based on reading techniques. Here, we only extracted two test cases that were used during testing process (see Table 1(a) and 1(b)). The complete test cases were documented in the project report.

Test cases were created and executed to evaluate the developed system. Any failed test cases were fixed, and test cases were re-tested to ensure that the function works as expected. In addition, the system had also been tested to users. This part will be elaborated in the following section.

TABLE I
EXAMPLES OF TEST CASES FOR TESTING

Table 1(a). Test case for reading techniques customization

Test Case ID	TC002
Test Case Summary	To ensure that the review leader modify and customize both the checklist-based reading and perspective-based reading techniques for a specific project.
Related Requirements	RLD-04, RLD-05, RLD-06, RLD-07, RLD-08, RLD-09, RLD-10, RLD-11, RLD-12, RLD-13, RLD-14
Pre-requisites	The review leader should select a project
Test Procedure	The user navigate between CBR and PBR and modify by clicking the relative icons
Expected Result	Step 1: Select a reading technique Step 2: System direct review leader to the selected reading technique Step3: The review leader decide whether to add, edit or delete a defect type, or perspective and its items. Step 5: The system direct reviewer to the modification page Step 6: The review leader modify and save Step 7: The system display the modified reading technique
Actual Result	The system shall register user and direct into his respective home page
Status	Pass
Remarks	-

Table 1(b). Test case for reviewing requirements by reviewer

Test Case ID	TC003
Test Case Summary	To ensure that the reviewer can click on review button and review each requirement successfully
Related Requirement	REV-4,REV-5,REV-6,REV-7
Pre-requisites	Reviewer must be logged in, entered to certain project and click on review button for the certain requirement
Expected Result	Step 1: Reviewer chooses requirement to be reviewed. Step 2: Reviewer Clicks on Review button and will be redirected to reviewing screen with according list of checklist item and answer Step 3: Reviewer reads questions and answer it Step 4: Reviewer finishes reviewing for the certain requirement by clicking on Save button Step 5: System will respond by providing an alert says that the process is successful
Actual Result	Reviewer can review each requirement quickly and easily
Status	Pass
Remarks	-
Created By	Abdul Rio Zac
Date of Creation	12/5/2018
Executed By	Isah
Date of Execution	13/5/2018
Test Environment	OS: Windows 7 Browser: Google Chrome

IV. RESULTS AND DISCUSSION

The implementation was developed based on the design that had been prepared in the previous stage. Each type of user has their own working dashboard to include the relevant features. Fig. 6(a) displays the SRS document author dashboard where the information about the document to be reviewed is provided. The submitted

document will be filtered so that each requirement can be organized for review process in the future. The document author will appoint the review leader for the project. In Fig. 6(b), the available projects are listed from the review leader dashboard. Here, the review leader can choose which project to work on.

(a) SRS document author dashboard

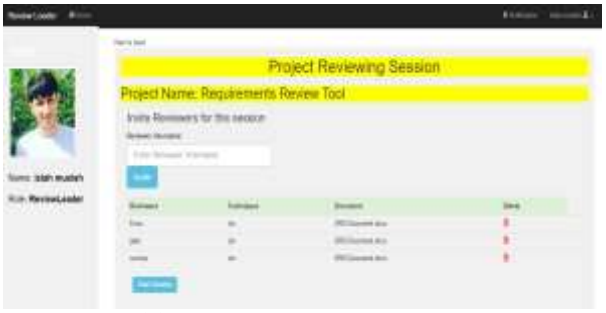
(b) Review leader dashboard

Fig. 6 Dashboard Implementation

Fig. 7(a) shows the setting for reading techniques that will be used by the assigned reviewers in the review session. The review leader is able to decide which reading technique or which combination of reading techniques to be adopted in the review session.

For example, if the review team consists of a mixture of roles in the development stages, the review leader can adopt the perspective-based reading technique. By using this option, it can focus on the benefits of the reading techniques in order to maximize defect detection during the review activity. In addition, the review leader can also adopt a mixture of reading technique including check-list based reading technique. In this feature, the review leader is able to customize the checklist items used in both the reading techniques. In Fig. 7(b), the review leader appoints the reviewers and sets the reading technique for each reviewer.

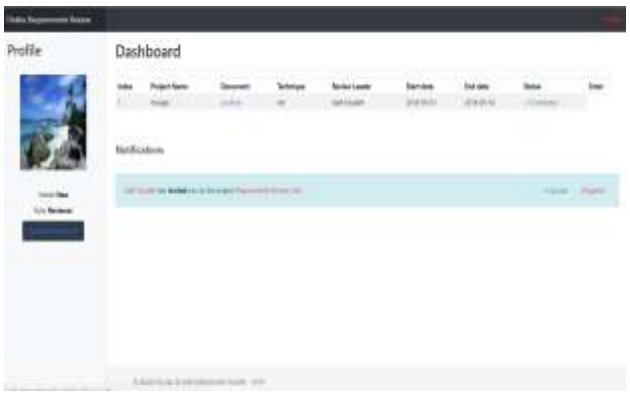
(a) Review leader's reading techniques customization page



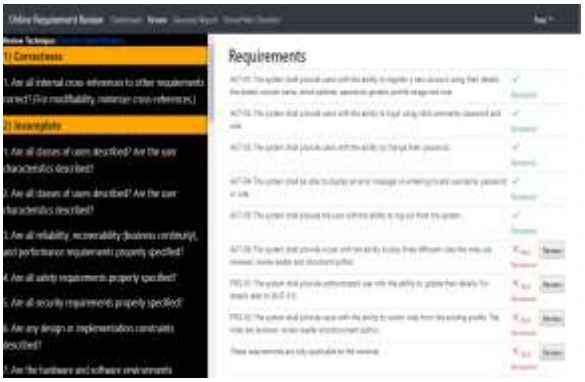
(b) Project reviewing session page

Fig. 7 Review Setting

For reviewers, when they log in, they will be landed on the reviewer dashboard as shown in Fig. 8(a). Here, the reviewers shall have access to the projects that has been assigned to them by the review leader(s). Initially, reviewers are required to accept the notification sent by the review leader. By doing this, reviewers will have direct access to the relevant documents to be reviewed. Fig. 8(b) denotes the reviewer's workspace when the reviewer performs requirements review activity by using the assigned reading technique.



(a) Reviewer dashboard



(b) Reviewer requirements reviewing page

Fig. 8 Reviewer dashboard

Fig. 9(a) depicts the feedback report after the reviewers had completed the review process. The result from the reviews can be generated to be submitted to the review leader. The feedback from all the assigned reviewers can be consolidated by the review leader. Here, the review leader can decide to accept or reject the feedback from the reviewers. By doing this, redundant feedback can be filtered to focus on the correction stage later (See Fig. 9 (b)).

(a) Reviewers feedback report

(b) Consolidation of reviewers' feedback by review leader

Fig. 9 Review Report

During the testing, a group of users performed a review activity by using the e-Review system. Following the session, they were asked to complete a post-experiment survey. The following Figures 10 and 11 depict the examples of the survey result. From out of 11 respondents that were from undergraduate Computer Science Students (Software Engineering specialization) and academicians, 54.5% used CBR, 27.3% and 18.2% used PBR as designer or tester perspective based on their backgrounds. The details of the experiment can be referred at [15].

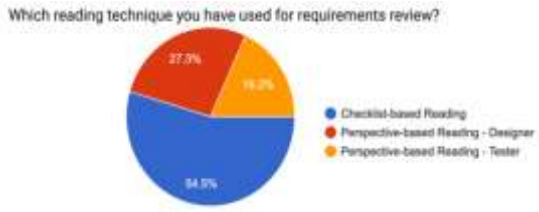


Fig. 10 Reading techniques

The respondents were asked if they have any difficulties in using the tool during the review. Most of the respondents responded positively and a few gave recommendations for improvements as can be seen in Fig. 11.

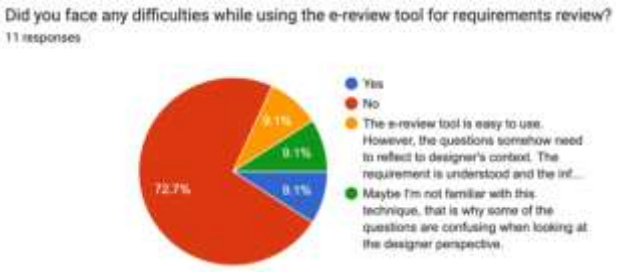


Fig. 11 Feedback of the e-Review tool

They were also asked to provide an-open ended feedback in the survey. Most of the feedback was positive.

V. Conclusion and Future Work

A requirement engineer typically focuses on reviewing requirements during requirements validation stage before the requirements are brought to another stage of requirements engineering process. Requirements review is a formal review process, in which several reviewers read all or part of SRS and search for requirements defects.

This study aimed to investigate the use of checklist-based and perspective-based reading techniques in the e-Review tool. The e-Review tool could support a requirements review team as well as the review leader in terms of saving resources such as cost, time, and effort. The tool could assist reviewers to perform requirements review by using multiple combination of reading techniques, which can be more effective, and also could help to detect more defects based on the benefits of each technique. In future work, we plan to further experiment the e-Review tool in terms in its effectiveness and usability.

One of the practical contributions of this work is to provide a proof-of-concept of using the existing reading techniques in a requirements review activity. In addition, the work adopts the current technologies to identify defects in requirements review based on the selected reading techniques. This idea can be embedded in requirements management tools specifically for requirements validation feature.

ACKNOWLEDGMENT

This research was funded by a grant awarded by the Malaysia Ministry of Higher Education (MOHE) FRGS/1/2019/ICT01/UIAM/03/1. We also would like to extend our appreciation to the Kulliyyah of Information and Communication Technology (KICT), International Islamic University Malaysia (IIUM) as well as the Computer Science Department for the opportunity to work for this project.

CONFLICT OF INTEREST

The author(s) declare that there is no conflict of Interest

REFERENCES

[1] S. Gregory, "Requirements for the New Normal: Requirements Engineering in a Pandemic," *IEEE Softw*, vol. 38, no. 2, pp. 15–18, 2021.

[2] Z. Liu, B. Li, J. Wang, and R. Yang, "Requirements engineering for crossover services: Issues, challenges and research directions," *IET Software*, vol. 15, no. 1, pp. 107–125, 2021, doi: 10.1049/sfw2.12014.

[3] C. Wiecher, J. Greenyer, C. Wolff, H. Anacker, and R. Dumitrescu, "Iterative and Scenario-based Requirements Specification in a System of Systems Context," *preprint - arXiv:2102.05400*, pp. 1–16, 2021.

[4] D. Hidellaarachchi, J. Grundy, R. Hoda, and K. Madampe, "The Effects of Human Aspects on the Requirements Engineering Process: A Systematic Literature Review," *IEEE Transactions on Software Engineering*, vol. 5589, no. August 2020, 2021, doi: 10.1109/TSE.2021.3051898.

[5] D. M. Fernández *et al.*, "Naming the pain in requirements engineering: Contemporary problems, causes, and effects in practice," *Empir Softw Eng*, vol. 22, no. 5, pp. 2298–2338, 2017, doi: 10.1007/s10664-016-9451-7.

[6] G. Kotonya and I. Sommerville, *Requirements Engineering: Processes and Techniques*, 1 edition. Chichester ; New York: Wiley, 1998.

[7] J. Lahtinen, *Application of the perspective- based reading technique in the nuclear I & C context CORSICA work report 2011*. 2011.

[8] L. H. Sulehri, "Comparative Selection of Requirements Validation Techniques Based on Industrial Survey," Blekinge Institute of Technology, Ronneby, 2009.

[9] A. Nordin, N. Z. Abidin, and S. H. M. Zaini, "Collaborative requirements review," *International Journal of Engineering & Technology*, vol. 7, no. 2.14, pp. 66–69, Apr. 2018, doi: 10.14419/ijet.v7i2.14.11158.

[10] F. Shull, I. Rus, and V. Basili, "How perspective-based reading can improve requirements inspections," *Computer (Long Beach Calif)*, vol. 33, no. 7, pp. 73–79, Jul. 2000, doi: 10.1109/2.869376.

[11] O. Dieste, E. Fernández, P. Pesado, and R. García-Martínez, "Analysis of inspection technique performance," 2009, Accessed: Aug. 14, 2022. [Online]. Available: http://sedici.unlp.edu.ar/handle/10915/21090

[12] F. Fabbrini, M. Fusani, S. Gnesi, and G. Lami, "The linguistic approach to the natural language requirements quality: Benefit of the use of an automatic tool," *26th Annual NASA Goddard Software Engineering Workshop, IEEE/NASA SEW 2001*, pp. 97–105, 2001, doi: 10.1109/SEW.2001.992662.

[13] "ARM Requirements Tool - Automated Quality Analysis of Natural Language Requirement Specifications | SE Goldmine." https://segoldmine.ppi-int.com/node/44711 (accessed Jan. 08, 2023).

[14] M. Kamalrudin, L. L. Ow, and S. Sidek, "Requirements Defects Techniques in Requirements Analysis: A Review," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 10, no. 1–7, pp. 47–51, Feb. 2018, Accessed: Jan. 03, 2023. [Online]. Available: https://jtec.utem.edu.my/jtec/article/view/3594

[15] I. M. Musleh, A. Nordin, and N. A. Emran, "An Experimental Study on Checklist-Based and Perspective-Based Requirements Reading Techniques Using E-Review Tool," *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 28, no. 3, pp. 351–367, Nov. 2022, doi: 10.37934/ARASET.28.3.351367.