SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

An Experimental Study on Checklist-Based and Perspective-Based Requirements Reading Techniques Using E-Review Tool

Islah Mohammad Musleh[1], Azlin Nordin[1,*], Nurul Akmar Emran[2]

[1] Department of Computer Science, Kulliyyah Information and Communication Technology, International Islamic University Malaysia 53100 Gombak, Kuala Lumpur, Malaysia
[2] Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Melaka (UTeM), Melaka, Malaysia

ARTICLE INFO

Article history:
Received 6 September 2022
Received in revised form 16 Nov. 2022
Accepted 20 November 2022
Available online 30 November 2022

Keywords:
Requirements validation; requirements error; requirements review; reading techniques; checklist-based reading; perspective-based reading

ABSTRACT

Software Requirements Review (SRR) is a formal review process in which several reviewers read all or parts of the Software Requirements Specification (SRS) to look for defects in the requirements. During requirements review sessions, reviewers may employ various reading techniques to ensure that the requirements have been completely and clearly specified. To review an SRS document, the review leader must organise a review session, the reviewers must physically meet and provide their review feedback during the session. In these situations, the review leader must schedule the review session based on the reviewers' availability, which can be laborious and time-consuming to arrange. Additionally, the review leader needs to manually consolidate all the outcomes of the review session, which could also require a considerable amount of effort and time. However, there has been an insufficient research to identify the effectiveness of reading techniques for requirement reviews by employing a dedicated tool support for requirements review. Using a web-based application, called e-Review, the aim of this study is to experiment the effectiveness of Checklist-based Reading (CBR) and Perspective-based Reading (PBR) techniques during requirements review session.

1. Introduction

The importance of Requirements Engineering (RE) in software development has been mentioned in many works of literature including in Gregory, [8], Hidellaarachchi *et al.*, [10], Liu *et al.*, [13] Wiecher *et al.*, [26]. In general, the generic RE activities are requirements elicitation, analysis and modelling, negotiation, validation, documentation, and management. The requirements validation activity is concerned with checking the requirements document for any requirements errors such as inconsistency, incompleteness, and inaccuracy in order to ensure that the requirements document adequately addresses all of the stakeholders' needs.

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

https://doi.org/10.37934/araset.28.3.351367

Requirements review is one of the most common requirement validation techniques, in which a group of reviewers read and analyze requirements, look for requirement issues, meet and discuss the issues, and agree on a set of actions to address the identified problems [16]. A variety of requirements review reading techniques such as checklist-based reading, defect-based reading, perspective-based reading, usage-based reading, and scenario-based reading have been employed for identifying requirements errors in the requirements review process [20].This study aims to experiment two of these reading techniques for requirements review that are Checklist-based Reading (CBR) and Perspective-based Reading (PBR) techniques. The selection of these two reading techniques was made possible by the fact that CBR and PBR are the most well-known reading techniques [6]. In the CBR technique, the reviewer obtains a checklist to look for certain types of requirements defects that are expressed as questions or statements, while the PBR technique focuses on analyzing requirements from several perspectives, such as end-users, designers, and testers [3].

To experiment the CBR and PBR techniques, an online collaborative requirement review tool, e-Review, was developed, which supports requirements review with two reading techniques. The purpose of this tool is to automate the requirements review process. In the e-Review tool, a review leader sends a review invitation to several reviewers, along with the reading technique(s) to be used for the review. By accepting the invitation, the reviewers can begin individually reviewing each requirement using the assigned reading technique. Once they have finished reviewing and providing feedback on all the requirements, they can submit their feedback to the review leader [12]. In contrast, 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 [12]. It is worth to note that both techniques employ the use of checklist, which is normally created and managed by the review leader.

The benefit of utilizing multiple reading techniques (as opposed to adopt a single reading technique) for requirements review is that each technique can facilitate in discovering requirements defects based on each technique's strengths. For this work, we adopted a combination of reading techniques i.e., the CBR and PBR techniques to review requirements documents in order to further improve the defect detection in requirements. In addition, once the reviewers provided their feedback, all the review feedbacks during validation need to be consolidated and this is a time-consuming process that necessitates a significant amount of effort and time. This is one of the reasons why the conventional (manual) requirements review process is a very challenging task for the review team, especially from the review leader's role. Hence, an online requirement review tool shall be able to (1) assist in validating requirements documents, (2) support in systematically managing and consolidating the requirements review feedback from multiple reviewers, and (3) provide a mechanism to allow multiple reading techniques to be adopted in the requirements review session.

The scope of this work is to compare two reading techniques for requirements review that are the (1) CBR, and (2) PBR techniques. The objective of the experiment is to investigate which reading technique is more effective in the detection of requirements defects. Hence, the research question for the experiment is: 'What is the difference between CBR and PBR in terms of defects detection rate in terms of accuracy and effectiveness?'. The primary goal of this study was to compare the CBR and PBR reading techniques to determine which one is more effective in detecting defects during requirements review activity.

The background of the research work, requirements review techniques, and research methodology are covered in the earlier sections of the report. The results and discussion will follow next, and the conclusion and recommendations for further research will be provided at the end.

2. Methodology

The literature review section will be presented based on the following sub-topics: requirements review, requirements reading methods used during the requirements validation activity, classification of requirements defects, and the final section on the CBR and PBR techniques.

2.1 Requirements Review

IEEE Standard for Software Reviews and Audits defines five different types of software review, which include: 1) technical review 2) management review 3) walkthrough; 4) review; and 5) audit (*1028-2008 - IEEE Standard for Software Reviews and Audits*, 2008). During a certain point of the software development life cycle, including planning, requirements, design, testing, and implementation, a review or a few review sessions may generally be conducted. The ISO/IEC/IEEE 24748-8:2019 standard defines technical reviews as a series of systems engineering activities conducted at a certain logical transition points in a system life cycle, by which the progress of a program is assessed relative to its technical requirements using a mutually agreed-upon set of criteria as defined in IEEE Standard for Technical Reviews and Audits on Defense Programs (IEEE Computer Society. Software & Systems Engineering Standards Committee et al., n.d.). The standard also recommends a comprehensive set of technical reviews such as Alternative System Review (ASR), System Requirement Review (SRR), System Functional Review (SFR), Preliminary Design Review (PDR), Critical Design Review (CDR), and a few more type of reviews. In addition, a technical review seeks to assess the suitability of a software product for its intended use and identify deviations from existing standards and specifications. In an SRR, several stakeholders go through the requirements documentation to look for any requirements defects.

Although there are challenges in the execution of requirements review activities in the industry, requirements review is regarded as one of the most effective software quality assurance techniques in software engineering [18]. A systematic walk-through can also be used to inform and evaluate a software solution. An independent evaluation of a software product can also be performed using a software audit (IEEE Computer Society. Software & Systems Engineering Standards Committee et al., n.d.; Nordin et al., 2018). According to Sulehri [21] requirements review is the most commonly used technique for requirements validation. To identify requirements defects, each member of the review team tries to comprehend and evaluate the requirement artefact. It is the responsibilities of each reviewer to go through the documents, discuss, and further examine the artefact to identify any requirements defects. The owner of the requirements document then needs to correct the defects found during the review. The general workflow of the requirements review process is depicted in Figure 1.

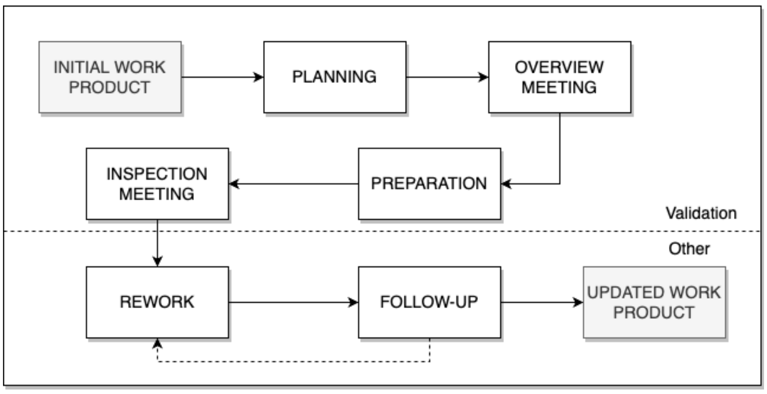

Fig. 1. Requirements Review Workflow [21]

2.2 Requirements Reading Techniques

The most prominent requirements reading techniques for requirements review are covered in this subsection. In this paper, seven types of requirements reading techniques in the literature are briefly elaborated i.e., 1) ad hoc, 2) CBR, 3) PBR, 4) usage-based, 5) scenario-based, 6) traceability-based, and 7) defect-based requirements reading techniques.

Ad hoc reading technique works without proper planning. All review team participants examine the provided requirements artefact for any requirements defects in the absence of proper guidance or instructions. This strategy may be effective with knowledgeable and experienced reviewers, but it could be ineffective with inexperienced reviewers. Sulehri [21] stated that defect detection is performed based on reviewer's experience. In contrast, inexperienced or untrained reviewers may find it difficult to detect requirements defects if no clear guidelines or instructions are provided.

The CBR technique provides a list of various question-based items to be used 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 identify requirements defects in the requirements specification [21].

PBR technique provides a set of instruction to assist reviewers in playing the roles of the main stakeholders of the project 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 the tester, and a user generates a series of use cases. Reviewers use a variety of questions to help spot defects when constructing abstractions. The items are normally on a standard category based of types of requirements defects. This is not a fixed collection of defect types, but it can be modified as required [6].

Usage-based Reading (UBR) technique employs a series of use cases to focus on during the review. The use cases will be used as a guidance for the reviewers when reviewing requirements, design, or code documentation in the same way as the test cases tell a tester how to test a system [23]. Scenario-based Reading (SBR) technique is designed to overcome the weaknesses of CBR technique, which aims to 1) guide on active review development of the document; 2) confine the reviewers' attention to a single point of concern, that is, showing what to review, which is called separation of concerns. These ideas of the scenario-based technique are applied as "reading scenarios". A reading scenario is made of two parts that are 1) an introduction that determines the focus of a scenario; 2) a collection of guidelines that provide a step-by-step overview of activities to be carried out by reviewers [6].

Defect-based Reading (DBR) technique is primarily applicable to requirements artifacts. The key concept of this technique is when an item is under review, different reviewers (normally in a group) should concentrate on a different group of defects. An example of a study which adopts the DBR technique in detecting ambiguities during interviews was conducted by Ferrari *et al*., [7]. Despite the possibility that this technique could be helpful in identifying requirements problems, it also boosts the cost of implementation because it necessitates additional personnel [6].

2.3. Defect Classification for Requirements

The main goal of the requirements review is to find defects in software requirements specifications (SRS). IEEE Standard classifies eight defect types that are (1) missing, (2) extra(superfluous), (3) ambiguous, (4) inconsistent, (5) improvement desirable, (6) not conforming to standards, (7) risk-prone, and (8) not implementable (*1233-1996 IEEE Guide for Developing System*

Requirements Specifications., n.d.). In addition, a requirements errors taxonomy was developed by Walia and Carver (Walia & Carver, 2009), where the requirement errors is classified into three categories, which are (1) people error, (2) process error, and (3) documentation error as shown in Figure 2.

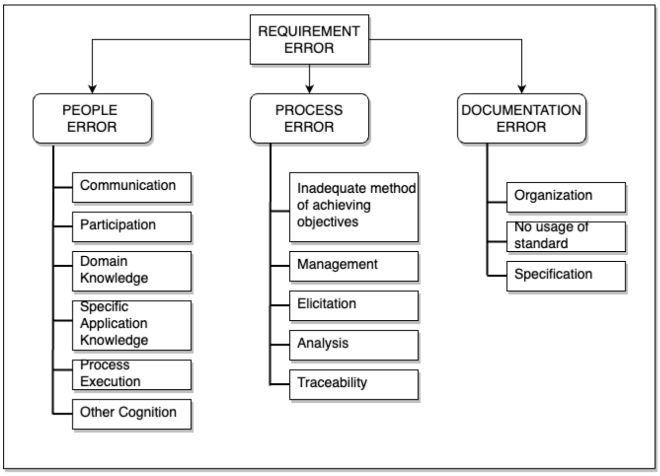

Fig. 2. Taxonomy of Requirements Errors [24]

According to the work [19], people errors are faults caused by the individuals involved in the preparation of the SRS documents; process errors are faults caused by an inadequate RE process and the selection of the incorrect tools to achieve objectives and goals; and document errors are faults caused by an incorrect specification of requirements and organisation, regardless of whether the author of the requirements documentation correctly understood the requirements.

2.4. Checklist-based Reading (CBR) and Perspective-based Reading (PBR) Techniques

This section summarizes some of the findings on the effectiveness of CBR and PBR techniques from the literature. According to Alshazly *et al*., [3], the PBR technique is more effective in terms of the total number of faults detected than the CBR technique. The research also mentioned that the PBR technique is more effective than the CBR technique when 1) reviewers are not quite familiar with the application domain; 2) dealing with omission and ambiguous faults, and 3) dealing with inconsistent defects. The justification for such case is that the reviewers who use the PBR technique review requirements from a specific perspective, hence, the process is more systematic, and adopts a more focused reading technique. However, for dealing with incorrect defects, the CBR technique is more effective compared to the PBR technique. In addition, this work also stated that the CBR technique could assist in detecting excessive defects as compared to the PBR technique.

Some other papers stated no difference in the defect detection using the CBR or PBR techniques in terms of effectiveness and efficiency. Dieste *et al*., [6] analyzed and compared the performance of a few reading techniques including the CBR and PBR reading techniques. Both individual and group measurements showed that the CBR technique output had the same result as the PBR technique in view of performance and efficacy.

Another research paper He *et al*., [9] showed that based on the amount of meeting gains and the number of false faults removed, the team meeting is more relevant for checklist teams. The study also found that in their defect identification, the teams who used the PBR approach had less overlap than those using a checklist. In terms of the individual results, reviewers from each perspective outperformed those who used the CBR technique. A bar graph for each team's Team Defects

Detection Rate (TDDR) is displayed in Figure 3. It is evident from the figure that the PBR teams were able to detect more defects than the teams using the CBR technique. These findings show a trend that the PBR teams performed better compared to the CBR teams concerning faults detection effectiveness, although the statistical result is not as good as that of individual results [9].

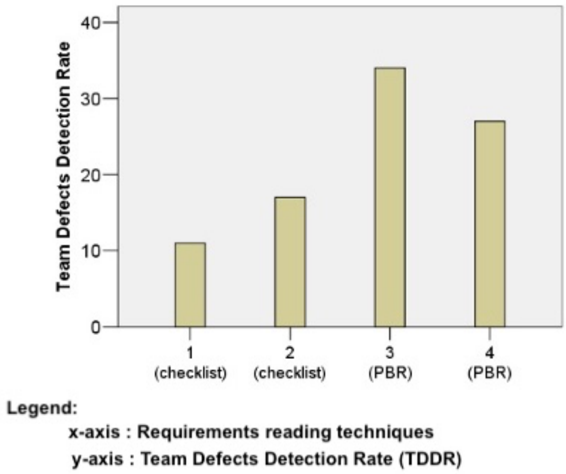

Fig. 3. Defect Detection Rate PBR vs CBR [9]

To the best of our knowledge, all the studies were carried out in physical settings using paper and pencil (manual or conventional) review; however, for this experiment, we used an online requirement review tool called e-Review. The tool will be elaborated in Section 3.2.

3. Methodology

The experimental research method is employed in this study as, according to the reviewed literature, it is the most common way to examine requirements reading strategies techniques. Nevertheless, nearly all the studies in this field that were referred used experiments to investigate the efficacy and efficiency of reading techniques in conventional requirements review, which at least has these two characteristics: 1) paper-based review of the requirements artefact; and 2) physical requirements review setting, whereby the review was carried out in a face-to-face session. During the Covid outbreak, people moved to online platforms to meet the demands of both individuals and organizations for products and services [17]. When a pandemic situation arises such as the Covid-19 pandemic, it is essential to be able to perform requirements review in an online platform with the ability of having distributed reviewers in any location, whenever and wherever. A study [15] also stated that during the pandemic, teaching and learning activities were more restricted and relied on the ICT technologies. As such, we conducted the review in online setting.

Figure 4 represents the process flow of the research methodology and the experimental design. After the literature review process, the e-Review[2] tool, which is a web-based requirements review system was deployed and prepared to be ready for the experiment. This research adopted the tool to be used as requirements review platform for the experiment with the CBR and PBR reading techniques. The e-Review tool will be discussed in Section 3.2.

[2] The e-Review tool was developed as a part of the researchers' previous work in another project.

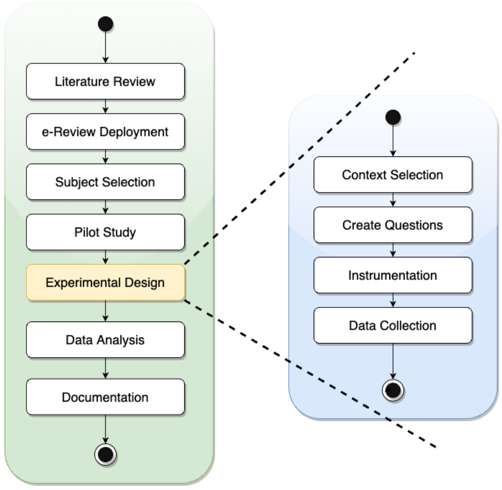

Fig. 4. Research Methodology and Experimental Design

The experiment aimed at investigating which reading technique is more effective in requirements error detection in requirements. This experiment is a preliminary validation to find effective reading techniques between CBR and PBR techniques. Due to the limitation of industry professionals to be employed as subjects, respondents of the experiments were selected from 1) computer science and software engineering undergraduate students, and 2) academics with Certified Professional for Requirements Engineering (CPRE) or Certified Testing Foundation Level (CTFL) certifications. According to the literature [4,5,19,22], using students as subjects in Software Engineering (SE) experiment is a common practice. Among the reasons are the high costs and difficulties in obtaining industry or practitioners as experimental subjects, which is why using students is thought to be the best alternative for SE experiments [19].

Furthermore, the student participants in this study were filtered based on their familiarity and understanding of the RE field, as well as other closely related SE fields such as software testing and software design. To select the most suitable participants, all SE students were sent invitations to participate in the experiment. According to the responses, not all SE students took RE or the related courses during the study duration or had previously taken the course. These students were excluded in the study.

Then, the basic criteria for participating in the experiments were specified. Subjects who were eligible to review software requirements using the CBR technique must have taken the RE course or were currently taking the course. Subjects who were assigned to review using the PBR technique from a designer perspective must at least took or currently taking the software design and architecture course. Likewise, subjects who were to review software requirements using the PBR technique from a tester perspective must have taken or currently taking the software testing course.

Prior to the actual experiment, a pilot study was conducted. The aims of the pilot study were (1) to validate the research elements and materials for the participants including the provided tutorials, instructions for the experiment, materials provided for the e-Review tool and the reading techniques; (2) to verify the process flow of the experiment and resolve any issues with the tool and/or the instructions. There was a total of 4 participants who took part in the pilot study. After the pilot study, based on the participants' data and feedback, some amendments had been made to the e-Review tool and the instructions for the experiment. For example, some of the items of the reading techniques were not applicable to the requirements. From the feedback of the pilot study, the provided instruction was updated to ensure clarity of the instruction.

The participants were given written instructions and video tutorials on how to accomplish the activities before the experiment. During the pilot study, validation of the video lessons and the experiment instructions had already been conducted. The planning and execution of the experiment will be covered in more detail in the following subsections.

3.1. Design of the Experiment

The participants of the experiment were employed from 1) undergraduate students, who were taking or already took the RE course i.e., CSCI 4323 Requirements Engineering and 2) academicians with CPRE, CTFL certifications or had experiences in RE area. The RE course covers all the RE activities and aligned with the CPRE content. The total number of participants was 12 students and 4 academic staff. The subjects were then divided into two main groups based on the reading techniques.

Eight subjects were assigned to review a set of requirements using the CBR technique and the other eight subjects were assigned to review the same set of requirements using the PBR technique. In order to provide a centralized collection of materials for the experiment and also to avoid confusion, a Google classroom was created for each of the techniques that is one for the CBR technique's subjects and the other for the PBR technique's subjects. The relevant materials including the documented guidelines and video tutorials were provided in both the classrooms and shared with the subjects in the platform. The included materials were (1) step-by-step tutorial on using the e-Review tool for reviewing software requirements; (2) a sample of each technique with instructions. Additionally, participants had a week to evaluate the requirements at their convenience. The time each reviewer spent going over the requirements was tracked by the system. The set of requirements adopted in this experiment consists of 10 functional requirements from the FOMS. The experimental design is represented in Figure 4.

3.2. e-Review Tool

e-Review is an online web-based tool, which was designed and developed to facilitate the requirements review process. The following activity diagram shows the overall flow of the tool. See Figure 5. It is important to note that the tool is made to handle several requirements reading techniques within a single review session. This means that a review leader may choose to assign various reading techniques to different reviewers. We normally made the assumptions that the same reading technique is used throughout this experiment and limit the flow to what is necessary to accomplish the review's objectives.

In general, the system supports three different roles of users, which are (1) document (SRS) author, (2) review leader, and (3) reviewer. All different roles of users need to log in before using the system. The system is redirected to a certain page based on chosen the role. To have an account, the user must self-register at the registration page, and they can then log in after the registration is successful. However, in order to assure smooth registration and prevent any unnecessary mistakes, the authors pre-registered the participants so that they could log in immediately based on their assigned roles.

If a user registers as a document author, he will be taken to the document author's dashboard and have access to all the features designed for the role, including adding review leaders to projects, creating new projects, uploading SRS documents for review, and tracking feedback after projects are finished. The dashboard for the review leader will also be shown if the user registers as the review leader. Here, the user can access the review leader's features, including adding the assigned project, inviting reviewers, customizing the checklist, designating a specific reviewer for each review technique, compiling the reviewers' generated feedback, and sending the finalised consolidated feedback to the document author.

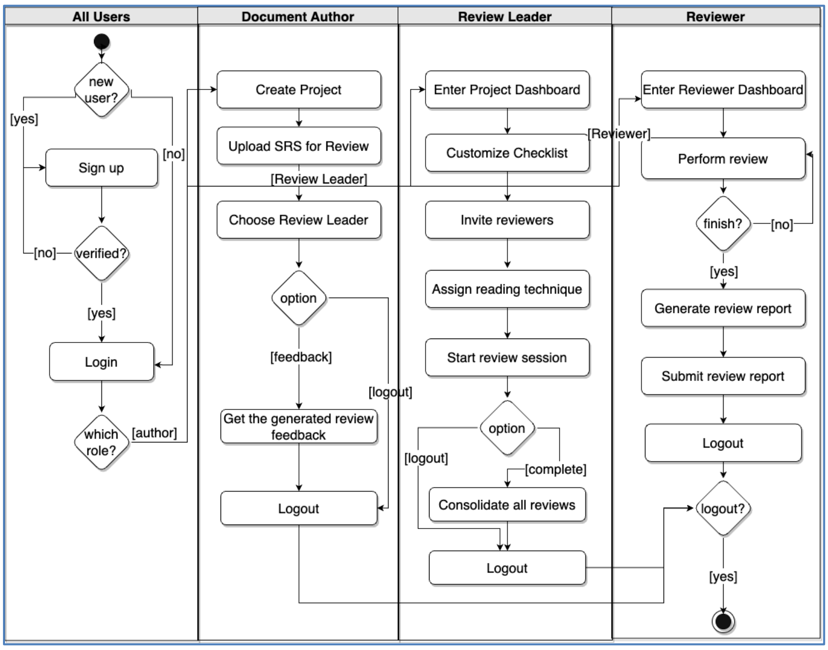

Fig. 5. e-Review Tool Workflow

If a user registers as a reviewer, the system will direct them to the reviewer's dashboard. The user can now review the assigned document, produce feedback, and submit the created insight to the review leader after accepting the invitation from the review leader. The authors of this study assumed the roles of document author and review leader to cover the flow of the complete requirements review activity as provided by the tool, and each participant was registered as a reviewer for the purposes of the experiment.

Figure 6 shows an example of a screenshot of the e-Review tool, which is focusing on the review process. The checklist items will be prompted for each requirement to guide the reviewers during the activity. The reviewer is expected to provide review feedback for each requirement by analyzing each requirement and looking for errors in the requirements. The issues are documented, and at the end, all review feedback is compiled and submitted to the review leader.

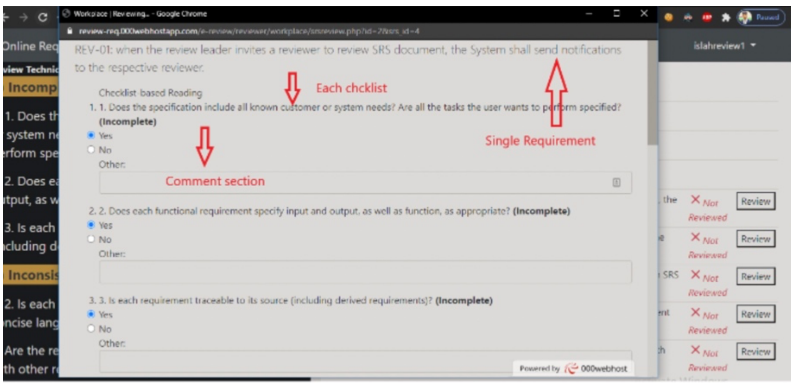

Fig. 6. A screenshot of e-Review tool

4. Result and Data Analysis

In this section, the result of the experiment is analyzed in detail to derive the research findings.

4.1. Demographic

At the beginning, all participants were asked to provide their demographic information and knowledge level of SE in a Google survey form. There were 16 participants in this study. Participants were students who enrolled in the undergraduate RE course in the 2nd Semester 2019/2020 session or academic staff who were eligible. Among the participants, 12 participants were female (75%) and 4 of them were male (25%). At the time of the experiment, each participant was either 1) enrolled in or had already finished the RE course; 2) had taken software testing or software design and architecture course(s); or 3) had CPRE or CTFL certificates, or expertise in the RE field. Ten of the students acknowledged having taken the software testing course, while eight of them mentioned having taken courses in software design and architecture. Four of the participants mentioned having taken part in a SRR in the past while five of the participants said they were aware of the requirements review but had never involved in any review activity. The experiment took over the span of a week.

4.2. Experimental Setup

The experiment was conducted based on the availability of the participants using an e-Review tool. By using the e-Review tool, each participant was expected to review the given requirements when it was convenient for them within the stated duration. The start and end times of each reviewer were recorded in the tool, and on average each participant spent approximately 30 minutes to review the given set of requirements. After the experiment, the participants were asked to give feedback on the tool. As finding for the tool's feedback part is not the scope of this paper, it is excluded from this paper, and will be reported in a subsequent work.

Firstly, the participants were required to fill in their demographic information and knowledge level in a survey form. They were then divided into two groups based on the reading techniques. In addition, the training materials for (1) both techniques and (2) step-by-step instructions for using the e-Review tool were shared via a Google classroom platform. A help section was also provided in each classroom to clarify any confusion regarding the experiment.

The experiment was conducted using a set of ten functional requirements, which were extracted from the Food Order Management System (FOMS). The requirements covered various features of the FOMS such as account management, authentication, search, and making a food order.

The CBR checklist items consist of four categories i.e., complete, consistent, standard, and verifiable. For PBR, two views were assigned: (1) the tester and (2) the designer perspectives. For each view, there were seven checklist items to consider during a review session. See a sample of CBR checklist item in Table 1. Instead of providing feedback on the entire requirements, the reviewers were expected to examine each requirement individually, and provide feedback based on the assigned techniques. Each technique-related checklist item had two potential responses: (1) yes or no, and (2) there was also an area for additional feedback for each item. A sample of the checklist implementation in e-Review tool is included in Section 3.2.

The authors developed a benchmark for evaluating review results, which was validated alongside the requirements by a RE expert. The expert has more than 7 years of experience in the field. She holds CPRE and CTFL certifications and has published articles in the RE area. The benchmark includes both the expected defects and the justification for each requirement statement.

This benchmark will be used to evaluate the data from the experiment and, later, to analyse the results. In both the pilot study and the main experiment, the researchers were available to assist the participants in clarifying any confusions. The pilot study involved four participants in all. After the pilot study, minor adjustments had been made to the tool, experiment materials, and the methodology to reduce any misunderstandings.

Table 1
CBR Checklist Item

Standard
1. Are all terms and units of measure defined?
Verifiable
1. Is each requirement verifiable by testing, demonstration, review, or analysis?
2. Are there measurable acceptance criteria for each functional and non-functional requirement?
3. Is each requirement uniquely and correctly identified?
Complete
1. Does the specification include all known customer or system needs? Are all the tasks the user wants to perform specified?
2. Does each functional requirement specify input and output, as well as function, as appropriate?
Consistent
1. Is each requirement written in consistent, clear, concise language?

4.3. Result

The number of errors discovered by participants using the two reading techniques that are the CBR and PBR is depicted in Figure 7. The experiment for the PBR technique considered both the designer and tester perspectives. Four participants reviewed the requirements using the PBR technique from the perspective of designers, while the remaining four participants reviewed the requirements from the standpoint of testers.

A defect frequency was used to analyze the results. The number of participants who detected the same error for the assigned reading technique was used to calculate the defect frequency. To clarify, the researchers set the review result to one if a specific participant's response is accurate (based on the validated benchmark) for a specific checklist item for a specific requirement. If a response is incorrect, a zero value will be assigned to a single participant's attempt. As a result, based on the assigned reading technique, this value will be accumulated and used to calculate the defect frequency for PBR and CBR.

Generally, it can be seen from Figure 7 that the number of defect frequency for PBR was higher than the CBR for almost all requirements. In terms of defect frequency rate, PBR participants discovered more defects than CBR participants. To be more specific, the highest defect frequency rate for both techniques was 4, which was the same for both techniques. Participants using PBR had a defect frequency of 4 for 2 requirements i.e., REQ2 and REQ5, whereas participants using CBR had a defect frequency of 4 for a single requirement only i.e., REQ4.

Requirements REQ3 had the lowest defect detection frequency, which was one. Furthermore, the highest frequency of defect detection using CBR for requirement REQ1 is 3 defects, and the lowest frequency is 1. While participants using PBR techniques had a maximum defect detection frequency of 2, the defect detection frequency for REQ7 was 1 for both reading technique participants. Three participants detected the same REQ8 faults using the CBR technique, with the lowest frequency of defect being 1.

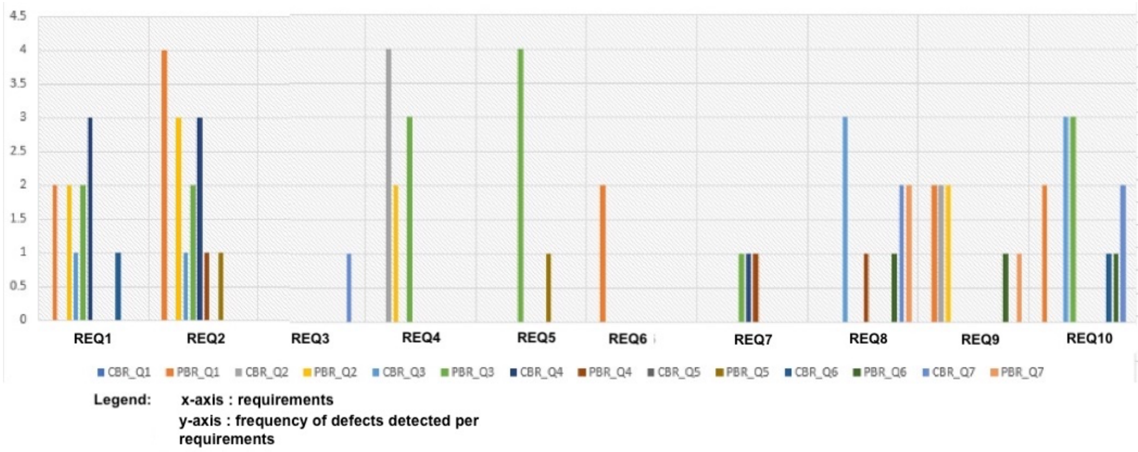

Fig. 7. CBR and PBR Defect Frequency

Participants of both techniques found a total of 5 defects for REQ9 with the highest frequency of 2 defects for both CBR and PBR techniques. Requirements REQ10 was related to the search functionality, which had the highest defect detection frequency of 3 for both CBR and PBR techniques.

4.4. Data Analysis

During the experiment, participants' feedback was entered into the e-Review database. Participants were given options of yes or no for each item of CBR and PBR reading techniques that were recorded. In addition, participants also had the option to comment on their opinion or error for each item. The comment box for each item aimed to make each item open-ended, which allowed the participant to express their opinions at length. Feedback was then exported into Excel sheets and interpreted into numbers for analysis.

4.4.1 Effectiveness of CBR versus PBR based on defect detection

The effectiveness of each technique was evaluated based on the findings by comparing the number of defects discovered using each technique. The frequency of problems discovered was compiled in order to better understand each criterion. As shown in Figure 8, participants using the PBR technique were able to detect requirements defects in almost all the assigned requirements. In comparison to PBR participants, who were successful in finding defects in nine out of ten requirements, CBR participants were unable to discover any defects in three of the assigned requirements. As a result, the PBR method was found to be more effective than CBR in terms of defect detection.

4.4.2 Effectiveness of CBR versus PBR based on numbers of defects detected

This section focuses on a more in-depth analytical comparison of CBR and PBR for all requirements related to defect detection. For REQ1, participants who used CBR encountered requirements issues with completeness and verification defects, whereas participants who used PBR discovered consistency and completeness issues. Both participant groups found the same number of defects, although they were of slightly different defect categories.

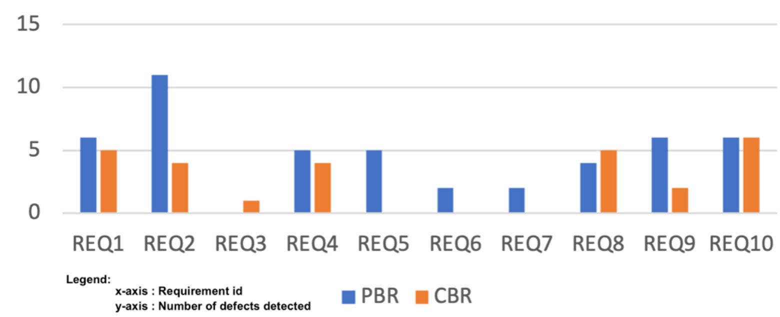

Fig. 8. CBR vs PBR

The highest number of defects, which were associated with interface, incompleteness, and inconsistency, were observed in REQ2, according to PBR users as represented in Figure 9. For a few of the requirements, the number of defects also stands in contrast to what was expected. For example, participants using the PBR technique did not manage to discover any defects for REQ3, whereas participants using the CBR technique failed to detect any defects in REQ4–7. However, a defect related to the consistency of the requirement was discovered by CBR users for REQ3. The standard, verification, and consistency were the expected defects to be discovered for this requirement. The same scenario was true for requirement REQ6, where only PBR participants were able to find a single defect. Requirement REQ6 is related to the order feature and the expected defect was related to ambiguity category.

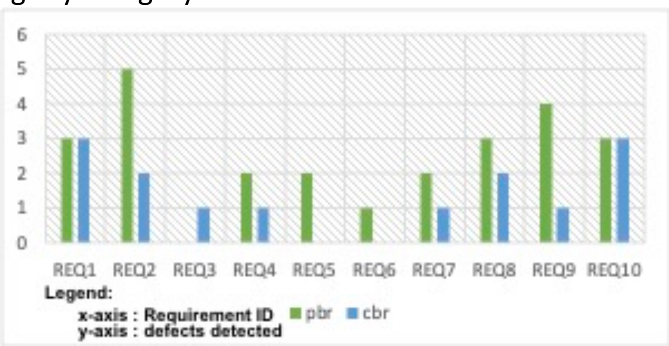

Fig. 9. Defects rate for PBR and CBR (based on requirements)

To further elaborate on the question of which reading technique is more effective between CBR and PBR, Table 2 lists the summary of the finding based on the accumulated data concerning defects found (union of defects found by the individual participants) and total defect frequency (defect found by multiple participants). From this table, the PBR group scored 75% in contrast to only 41% for CBR group for accumulated defects found by individual participants.

Table 2
CBR vs PBR Finding Summary

(Review Item) Requirements	**Food Ordering Management System (FOMS)**	
Reading Techniques	CBR	PBR
Defects Found/Total Defects	13/32 (41%)	24/32 (75%)
Total Frequency of Defects Detected	27	44
Effectiveness	1.27	2.27

5. Discussion

The threat to the accuracy and interpretation of the findings will be briefly discussed in this section of the paper.

5.1. Threats to Validity

A potential external threat to validity is the lack of industrial practitioners in the experiment. However, this risk is minimized by employing fourth-year students who are about to complete their studies and beginning to work in the industry. In addition, the four academic staff who used the PBR technique for reviewing the requirements can be a potential threat to validity since they were more experienced than the students. However, for the CBR technique, all participants were undergraduate students. As explained earlier, participants were assigned to the respective techniques based on their prior RE knowledge and experience in the area.

For clarification, in order to review using the PBR technique, participants must play the roles of a designer, or a tester. The students who had not yet taken any of the elective courses related to software design and (or) software testing at the time of the experiment, they were assigned to the CBR technique instead of the PBR technique. The utilization of the e-Review tool rather than manual reviewing could also pose a threat to validity. Although both video and written tutorial were provided, participants could find it more challenging to pose questions during the review than they would during a manual conventional review session.

Internal validity may also be compromised by the participants' knowledge that their performances in this experiment will not affect their grades in the course. There is a possibility that the participants will be discouraged. For instance, they might perceive that their participation is a waste of time or that they lack motivation to comprehend the experiment procedures. Nonetheless, the instructor had discussed potential benefits of the requirements review exposure. It was made clear that genuine participation was required for the experiment.

5.2 Interpretation of Result

The primary goal of this study was to compare the CBR and PBR reading techniques to determine which reading technique is more effective in defects detection. The findings in this research were evaluated to support the findings among some earlier experiments, including work in (He & Carver, 2006). Additionally, the analysis's findings support the research question that is to find the answer to 'What is the difference between CBR and PBR in terms of defects detection rate in terms of accuracy and effectiveness?'. The results of individual performance analysis (based on accuracy of the defect detection depending on the expert benchmark) and frequency of defect detection revealed that participants who used the PBR technique found more defects than those who used the CBR technique.

The primary cause of this finding might be related to the various reviewers' perspectives in the PBR, including that of the designer and tester. Participants who reviewed the requirements from the perspective of a designer might analyze the requirements to his understanding of design knowledge and determine whether anything was missing, unclear or confusing. Furthermore, participants who reviewed from a tester's perspective could relate the requirements to testing activity, for instance, if any information is missing or ambiguous in the stated requirements, which may lead to issues during testing. While with the CBR technique having the same goal to detect defects, the reviewers might not be focused as they need to consider all possibilities. However, due to the small sample size and use of a single experiment, we acknowledged that the findings might not hold to other RE circumstances to be generalized.

6. Conclusions and Future Work

This section briefly provides the conclusion of the study and explains the theoretical and practical contribution of this paper to the requirements validation area.

Software Requirements Review (SRR) is a formal requirements review procedure activity, which involves several reviewers reading all or portions of the software requirements specification (SRS) and looking for requirements errors. This study used an e-Review tool as a platform to investigate the effectiveness of checklist-based (CBR) and perspective-based reading (PBR) techniques. This study, which involved 16 participants, sought to find out whether the number of defects found differed while using different reading techniques. The experiments included undergraduate students and academicians with relevant knowledge as participants. To review the provided requirements for defects, the participants were divided into two groups: CBR reviewers and PBR reviewers. Each member of the two groups reviewed the provided requirements on their own.

An experimental methodology was used to perform the study on the provided requirements, which were extracted from FOMS. Furthermore, both strategies utilized the same quantity of resources. According to the results of the experimental investigation, participants who used the PBR technique for evaluating requirements outperformed those who used the CBR technique in terms of defect detection. This discovery corroborates the findings of some other studies in this field, including those by Maldonado *et al*., [14] and He *et al*., [9]. The various viewpoints are taken into account for the cause. As a result, it may be concluded from our findings that the PBR technique is preferable for reviewing requirements.

As a result of our findings, we can conclude that the PBR technique is a better option for reviewing requirements. We believe that more research with more requirements and a larger industrial sample size will yield a better outcome.

6.1 Theoretical Contribution

This research is intended to compare the effectiveness of different requirements reading techniques. In order to uncover the comparison, an experiment was carried out to evaluate those reading techniques. The research used a web-based e-Review tool to facilitate the review session during the experiment. This tool was originally designed to support several requirements reading techniques to be used in requirements review session to maximize the benefits of each reading technique. Other studies including in Alshazly *et al*., [3] also recommended this strategy. In conventional requirements review setting, handling multiple reading techniques at once is tedious and may necessitate a significant amount of time, effort, and resources. Nonetheless, it is anticipated that this problem can be reduced by employing a specialized tool to assist the review leader in managing the complex flow.

This study aids in comparing the efficacy of various requirements reading methods. An experiment was conducted to evaluate those reading techniques to find the comparison. To facilitate the review session during the experiment, the research adopted a web-based e-Review tool. To optimize the benefits of each reading technique, this tool was created to support several reading techniques during requirements review session. Other work such as in Alshazly *et al*., [3] also recommended this strategy. In conventional requirements review setting, it is tedious to handle more than one reading technique at a time. In addition to the complexity of the review process, the consolidation of the review feedback from all the reviewers including removing duplications and providing the necessary reports are among the challenges that also need to be handled. Nonetheless, these issues can be reduced by using a dedicated tool that can facilitate the review leader to manage the flow and focus on the other important tasks. However, for the experiment, we did not focus on the use of multiple reading techniques at the same time.

6.2 Practical Contribution

The literature indicates that most of the studies to the best of our knowledge such as Nordin *et al*., [16] and Shull *et al*., [20] that had been conducted on identifying the effective requirements reading strategies were based on conventional review activity used within physical setting. This study, however, is different from the other studies in that it employed an online e-Review tool for the experiment.

6.3 Future Work

As previously stated, the designed and developed e-Review tool was created to accommodate multiple reading techniques during a requirement review session. However, the researchers did not concentrate on using various reading techniques simultaneously for this experiment. We plan to be able to investigate the tool's effectiveness in a requirement review context in the near future. The researchers also plan to keep assessing the e-Review tool's efficacy and usability in further work. In addition, there is also a possibility to address any cultural differences that might give an impact to how reviewers provide their feedback during requirements review sessions, which was inspired by a work, which explores the cultural differences in another domain such as in Wang *et al*., [25].

Acknowledgment
This research was funded by a grant awarded by the Malaysia Ministry of Higher Education (MOHE) i.e., FRGS/1/2019/ICT01/UIAM/03/1. The authors also would like to express our appreciation to the International Islamic University Malaysia (IIUM), Kulliyyah of Information and Communication Technology (KICT), and the Computer Science Department for providing us with the opportunity to work on this project.

References

[1] *1028-2008 - IEEE Standard for Software Reviews and Audits*. (2008). 1–53.
[2] 1233-1996 IEEE Guide for Developing System Requirements Specifications. (n.d.).
[3] Alshazly, Amira A., Ahmed M. Elfatatry, and Mohamed S. Abougabal. "Detecting defects in software requirements specification." *Alexandria Engineering Journal* 53, no. 3 (2014): 513-527. https://doi.org/10.1016/j.aej.2014.06.001
[4] Ameta, Deepti, and Saurabh Tiwari. "Why does the experiment fail with students? An Experience." In *14th Innovations in Software Engineering Conference (formerly known as India Software Engineering Conference)*, pp. 1-5. 2021. https://doi.org/10.1145/3452383.3452398
[5] Berander, Patrik. "Using students as subjects in requirements prioritization." In *Proceedings. 2004 International Symposium on Empirical Software Engineering, 2004. ISESE'04.*, pp. 167-176. IEEE, 2004. https://doi.org/10.1109/ISESE.2004.1334904
[6] Dieste, Oscar, and Enrique Fernández. "Analysis of inspection technique performance." In *XV Congreso Argentino de Ciencias de la Computación*. 2009.
[7] Ferrari, Alessio, Paola Spoletini, Beatrice Donati, Didar Zowghi, and Stefania Gnesi. "Interview review: detecting latent ambiguities to improve the requirements elicitation process." In *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pp. 400-405. IEEE, 2017. https://doi.org/10.1109/RE.2017.15

[8] Gregory, Sarah. "Requirements for the new normal: Requirements engineering in a pandemic." *IEEE Software* 38, no. 2 (2021): 15-18. https://doi.org/10.1109/MS.2020.3044403
[9] He, Lulu, and Jeffrey Carver. "PBR vs. checklist: a replication in the n-fold inspection context." In *Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering*, pp. 95-104. 2006. https://doi.org/10.1145/1159733.1159750
[10] Hidellaarachchi, Dulaji, John Grundy, Rashina Hoda, and Kashumi Madampe. "The effects of human aspects on the requirements engineering process: A systematic literature review." *IEEE Transactions on Software Engineering* (2021). https://doi.org/10.1109/TSE.2021.3051898
[11] IEEE Computer Society. Software & Systems Engineering Standards Committee, Institute of Electrical and Electronics Engineers., & IEEE-SA Standards Board. (n.d.). *IEEE standard for technical reviews and audits on defense programs*. https://doi.org/10.1109/IEEESTD.2015.7105321
[12] Lahtinen, Jussi. "Application of the perspective-based reading technique in the nuclear I&C context: CORSICA work report 2011." (2012).
[13] Liu, Zhengli, Bing Li, Jian Wang, and Rong Yang. "Requirements engineering for crossover services: Issues, challenges and research directions." *IET Software* 15, no. 1 (2021): 107-125. https://doi.org/10.1049/sfw2.12014
[14] Maldonado, Jose C., Jeffrey Carver, Forrest Shull, Sandra Fabbri, Emerson Doria, Luciana Martimiano, Manoel Mendonça, and Victor Basili. "Perspective-based reading: a replicated experiment focused on individual reviewer effectiveness." *Empirical Software Engineering* 11, no. 1 (2006): 119-142. https://doi.org/10.1007/s10664-006-5967-6
[15] Masrom, Maslin, Mohd Nazry Ali, Wahyunah Ghani, and Amirul Haiman Abdul Rahman. "The ICT implementation in the TVET teaching and learning environment during the COVID-19 pandemic." *International Journal of Advanced Research in Future Ready Learning and Education* 28, no. 1 (2022): 43-49.
[16] Nordin, A., Abidin, N. Z., & Zaini, S. H. M. (2018). Collaborative requirements review. *International Journal of Engineering & Technology*, *7*(2.14), 66–69. https://doi.org/10.14419/ijet.v7i2.14.11158
[17] Roslan, Nur Mira Alisa, and Nur Syazwani Mohd Nawi. "FOOD DELIVERY SERVICE APPLICATION DURING PANDEMIC COVID-19." *Journal of Technology and Operations Management* 17, no. 1 (2022): 71-79. https://doi.org/10.32890/jtom2022.17.1.7
[18] Salger, Frank. "Requirements reviews revisited: Residual challenges and open research questions." In *2013 21st IEEE International Requirements Engineering Conference (RE)*, pp. 250-255. IEEE, 2013. https://doi.org/10.1109/RE.2013.6636725
[19] Salman, Iflaah, Ayse Tosun Misirli, and Natalia Juristo. "Are students representatives of professionals in software engineering experiments?." In *2015 IEEE/ACM 37th IEEE international conference on software engineering*, vol. 1, pp. 666-676. IEEE, 2015. https://doi.org/10.1109/ICSE.2015.82
[20] Shull, Forrest, Ioana Rus, and Victor Basili. "How perspective-based reading can improve requirements inspections." *Computer* 33, no. 7 (2000): 73-79. https://doi.org/10.1109/2.869376
[21] Sulehri, Latif. "Comparative selection of requirements validation techniques based on industrial survey." (2010).
[22] Svahnberg, Mikael, Aybüke Aurum, and Claes Wohlin. "Using students as subjects-an empirical evaluation." In *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement*, pp. 288-290. 2008. https://doi.org/10.1145/1414004.1414055
[23] Thelin, Thomas, Per Runeson, and Claes Wohlin. "An experimental comparison of usage-based and checklist-based reading." *IEEE Transactions on Software Engineering* 29, no. 8 (2003): 687-704. https://doi.org/10.1109/TSE.2003.1223644
[24] Walia, Gursimran Singh, and Jeffrey C. Carver. "A systematic literature review to identify and classify software requirement errors." *Information and Software Technology* 51, no. 7 (2009): 1087-1109. https://doi.org/10.1016/j.infsof.2009.01.004
[25] Wang, Dan, Terh Jing Khoo, and Zhangfei Kan. "Exploring the Application of Digital Data Management Approach for Facility Management in Shanghai's High-rise Buildings." *Progress in Energy and Environment* 13 (2020): 1-15.
[26] Wiecher, Carsten, Joel Greenyer, Carsten Wolff, Harald Anacker, and Roman Dumitrescu. "Iterative and scenario-based requirements specification in a system of systems context." In International Working Conference on Requirements Engineering: Foundation for Software Quality, pp. 165-181. Springer, Cham, 2021. https://doi.org/10.1007/978-3-030-73128-1_12