Journal of Theoretical and Applied Information Technology

15th October 2023. Vol.101. No 19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

EXPLORING STRATEGIES FOR OVERCOMING ISSUES OF USER INVOLVEMENT IN AGILE SOFTWARE DEVELOPMENT: A SYSTEMATIC LITERATURE REVIEW

QUDRATTULLAH OMERKHEL[1], OTHMAN MOHD YUSOP [2], SAIFUL ADLI ISMAIL[3], AZRI AZMI[4]

[1]Razak Faculty of Technology and Informatics, Universiti Teknologi Malaysia, Malaysia

[1]Computer Science Faculty, Kabul Education University, Kabul, Afghanistan

[2,3,4] Razak Faculty of Technology and Informatics, Universiti Teknologi Malaysia, Malaysia

E-mail: 1omerkhel-1988@graduate.utm.my, [2]othmanyusop@utm.my,

[3]saifuladli@utm.my, [4]azriazmi@utm.my

ABSTRACT

The present systematic literature review (SLR) explores the challenges and strategies associated with managing users during requirement elicitation within agile software development. Drawing insights from an analysis of 24 relevant studies, this study comprehensively examines the issues that arise and the effective approaches to overcome them. The findings reveal five prominent challenges of user involvement during requirement elicitation. The most dominant issues identified are the lack of user involvement, insufficient user knowledge, and a deficit in the expertise of the Product Owner. These challenges can hinder the effective integration of user perspectives and needs into the development process. To address these challenges, the study identifies seven strategies that Product Owners can adopt to facilitate effective user involvement. These strategies include Mind Maps, User Interface Mockups, Workshops, Hybridism (combining agile and non-agile techniques), Face-to-Face Meetings, Continuous Delivery, and Training and Learning initiatives. The application of these strategies empowers Product Owners and software practitioners to enhance user involvement, improve communication, and streamline the requirement elicitation process in agile software development. The outcomes of this SLR provide valuable insights for both researchers and software practitioners, exploring the complex dynamics of user involvement in agile contexts. By recognizing these challenges and deploying effective strategies, software development teams can ensure more successful requirement elicitation processes, leading to the creation of software products that better align with user needs and expectations. This review contributes to a deeper understanding of user involvement challenges and offers actionable guidance for optimizing the requirement elicitation within agile software development paradigm.

Keywords: *User Involvement, Requirement Elicitation, Agile Software Development, Systematic Literature Review.*

1. INTRODUCTION

Software engineering (SE) research aims to improve software development processes by emphasizing the involvement of users and the establishment of quality requirements, as well as the estimation of time and cost associated with the required development effort. Agile software development, have gained substantial popularity in recent years due to their flexibility, adaptability, and user-centric approach [1]. User involvement plays an essential role in ensuring that to elicit quality requirements and software solutions meet user needs and expectations [2].

The importance of user involvement in software development has been recognized for over five decades, as evidenced by extensive research in organizational management [3]. This includes "*group problem solving, interpersonal communication and personal motivation*". One of the most significant factors in determining the success of the agile software development process is Requirements Engineering (RE) and user involvement are essential to establish a

collaborative environment with a continuous feedback loop [4].

Active user involvement is a crucial element for the success of a system, especially in agile software development. Unlike traditional methods, user involvement in agile is not confined to just the initial phases of development; instead, it persists throughout the entire development process [5]. By involving users in this manner, developers can gain a deep understanding of their requirements and effectively tailor to their needs. Consequently, this approach often results in greater satisfaction among users with the final product [6]. This approach helps to reduce the risk of building a product that does not meet the user's requirements. Project risks are potential issues that can impact the project's timeline or resources, making it crucial to address them proactively to effectively manage project requirements and avoid expensive rework or potential project failure [7].

User involvement is closely related to the success of a software development project, so projects become more successful with higher user involvement, project success is defined by time (i.e. within time delivery), cost (i.e. within or under budget), and quality software [8]. User involvement in terms of user satisfaction can also be beneficial to this project and their satisfaction is supported by the first principle of the Agile manifesto [9], user satisfaction is defined in terms of fulfilling user's expectations about product performance (performance requirements, which specify a minimum acceptable performance for the system) [1].

Typically, user involvement activities associated with their involvement in requirement elicitation include tasks such as identifying, analyzing, prioritizing, reviewing, and validating the requirements, as well as testing the developed features [10]. Affected users can be included as "end users, product owners, project sponsors, subject matter experts, or business analysts" [11].

User involvement is mainly relevant during the initial phases of development, which primarily focus on problem definition and requirement identification. However, it is less likely for users to be actively involved in the design and physical coding aspects of the process. Instead, their involvement becomes more appropriate during activities such as system testing and installation, where they play a crucial role [12].

Requirements Engineering plays a vital role in the development of software products, as it ensures the essential specifications are well-defined. Unlike traditional approaches that rely on a requirements specification document, agile RE adopts a prioritized list of requirements known as the Product Backlog [13]. In agile RE, the various activities involved, such as elicitation, documentation, validation, negotiation, and management, are not strictly separated but rather repeated iteratively, with only necessary information elaborated before each new iteration commences.

In addressing the initial ambiguity of agile requirements, Inayat et al., [14] propose, the use of techniques like face-to-face communication or prototyping rather than rigidly following to documentation standards. Heck & Zaidman, [15] further describe agile RE as a collaborative process that employs just enough documentation, conducted just-in-time, and is sustainable. A key aspect of agile software development is the active involvement of users throughout the development process.
This article presents the outcomes of a comprehensive systematic literature review that concentrates on the effective management of user involvement in requirement elicitation within the context of agile software development.

The paper's structure is organized as follows: Section 2 provides a brief background outlining the significance of user involvement. Section 3 explains the research objectives and research questions, and offers a detailed explanation of the review methodology, including the search strategy, selection process, quality assessment, data extraction, and analysis. In Section 4, a comprehensive summary of the main findings is presented, offering an overview of the selected studies and addressing the research questions. The implications of the findings and the study's limitations are deliberated in Section 5. Lastly, Section 6 serves as a conclusion, presenting paths for future research in this domain.

2. BACKGROUND

User involvement plays an important role in agile software development to ensure that the developed product meets the needs and expectations of the users. However, several issues and challenges have been identified in the existing literature regarding user involvement during requirement elicitation within agile projects.

Software development companies face a variety of issues that they must address, including evolving user requirements, market dynamics, integration

complexities, and technological advancements [16]. The issues mentioned are compelling software developers to embrace agile software development methodologies as their preferred approach [17].

According to Hoda et al. [18], the agile methodology supports a high degree of collaboration between the development team and users. It highlights the delivery of software functionalities that bring tangible business value in each incremental release. User involvement is a fundamental aspect of agile practices, including activities such as "planning, prioritizing, reviewing, and providing feedback" [19].

Agile software development, despite its benefits, presents difficulties in effectively incorporating user involvement, which remains a persistent issue for software development teams [20].

Fabio et al. [21] stated that it can be challenging to assume that users are willing and able to actively involve in the agile software development process. Handy collaboration between users and the development team is crucial for the ongoing success of software development in agile environments. Therefore, it is important to re-evaluate strategies for involving users who may be hesitant or unable to involve in strict collaboration.

Empirical studies have shown that user availability and accessibility can be challenging, despite the assumption and promotion of user availability in agile methodologies. Factors such as time, cost, and workload often limit actual user involvement, leading to the use of proxy users or representatives located at the user's site [22].

Agile software development emphasizes collaboration and communication between the team and users, including end-users. Without user involvement, negative consequences arise: limited feedback, misunderstood requirements, inefficient development, and decreased user satisfaction [23]. Hoda et al. [18] identified several causes that contribute to the limited involvement of users in ASD projects. These factors include *"skepticism and hype, the distance factor, lack of time commitment, dealing with large customers, fixed-bid contracts, and ineffective customer representatives."* Additionally, the consequences of insufficient user involvement in ASD projects, as mentioned by Hoda et al. [18] *"pressure to over-commit, problems in gathering and clarifying requirements, problems in prioritizing requirements, problems in securing feedback, loss of productivity, and in extreme cases, business loss".*

Agile relies on user feedback to validate and improve the product incrementally, but without it, the team lacks critical information about usage and required improvements [4].

The absence of user expertise and knowledge of the technology or the development process can obstruct effective user involvement. Users may struggle to communicate their needs or provide actionable feedback, leading to a product that does not fully meet their requirements [24].

The role of the product owner, who is responsible for enabling communication between the development team and users, is crucial. However, ineffective product owners lacking expertise in requirements engineering best practices can hinder user involvement and lead to conflicts or project failures [25]. In agile software development, an important issue is maintaining user involvement throughout the development process in regular iterations. Often, users who are impacted by the system's requirements or changes are not actively involved. E.-M. Schön et al. [26] documented experiences from agile software development practitioners, revealing instances where the product owner lacked a clear understanding of the actual user requirements, this led to situations where early User Interface (UI) prototypes were tested by the wrong users, resulting in potential conflicts and failures.

Therefore, it is worth reconsidering how software practitioners can involve users who may be reluctant and/or incapable of strict collaboration [21].

3. METHODOLOGY

The systematic literature review was performed in adherence to the established guidelines, specifically following the guidelines for Systematic Literature Reviews in Software Engineering proposed by Kitchenham and Charters [27]. The review process comprised three primary phases, and Figure 1 visually represents the key steps involved in each of these phases. To handle the large volume of studies retrieved during the review, we employed Mendeley software and Excel sheets to facilitate efficient information organization and management.

3.1 Planning the Review

The systematic literature review was precisely designed by formulating research questions that directly corresponded to our research objectives. To ensure comprehensive coverage of relevant studies, a well-defined search strategy and search string

were developed. This strategy enabled us to effectively retrieve pertinent literature while following to strict inclusion and exclusion criteria. The explicit definition of these components ensured the review's rigor and minimized any potential bias in the study selection process. Below, we offer a more comprehensive explanation of each of these aspects.

3.1.1 Objectives and Research Questions

The primary objective of this study was to assess the extant literature concerning requirement engineering and explore the role of user involvement in agile software development. To achieve this aim, two research questions were carefully formulated:

Research Question 1 (RQ1): What are the issues of the existing studies on managing users during requirement elicitation?

Research Question 2 (RQ2): How product owner can facilitate issues in managing users during requirement elicitation?

These research questions served as the foundation for guiding the systematic literature review and facilitated a thorough investigation into the specific aspects of user involvement in agile software development concerning requirement engineering. By addressing these research questions, we aimed to contribute valuable insights to the existing knowledge in this field.

3.1.2 Search Strategy

The systematic literature review was carried out following the established guidelines for SLR in Software Engineering as proposed by Kitchenham and Charters [27]. To ensure a comprehensive coverage of relevant publications, we conducted searches on several well-known and widely utilized electronic databases, including Scopus, Web of Science, IEEE Xplore, Science Direct, and Springer Link. The selection of these databases was motivated by their accessibility through the UTM Library and their reputation for hosting high-quality research papers.

The initial retrieval of papers from the electronic databases was supplemented by a reference search, also known as snowballing. This supplementary approach aimed to identify additional relevant research papers that might not have been captured in the primary search. By incorporating this technique, we aimed to include valuable studies that could have otherwise been overlooked.

Following the data retrieval phase, we applied rigorous inclusion and exclusion criteria to the retrieved research papers. These criteria were applied in two separate rounds, as described in Section 4 of the review. The purpose of applying these criteria was to ensure the selection of studies that aligned closely with our research objectives and that met the established quality standards for our systematic literature review.

3.1.3 Search Criteria

For this systematic literature review, the search criteria consisted of three parts: C1, C2, and C3. C1 included keywords related to user involvement, such as "user," "involvement," and "user involvement." C2 consisted of keywords related to agile software development methods, including "Scrum," "XP" (Extreme Programming), "agility," and "agile." Lastly, C3 encompassed keywords associated with requirement engineering, such as "requirement," "user story," "feature," and "requirement engineering."

The Boolean expression representing the search criteria was as follows:

C1 AND C2 AND C3 (1)

To initiate the search process in electronic databases, two primary search keywords were derived from the research questions: "user involvement" and "Agile requirement engineering." Building on these primary keywords, alternative keywords were formulated. These keywords were then combined to construct the final search strings for each database. The construction of each search string was performed manually, utilizing the search functionality provided by the respective database. The search process for each database was approached as a learning experience, and an experimental effort was invested to optimize the effectiveness of the search.

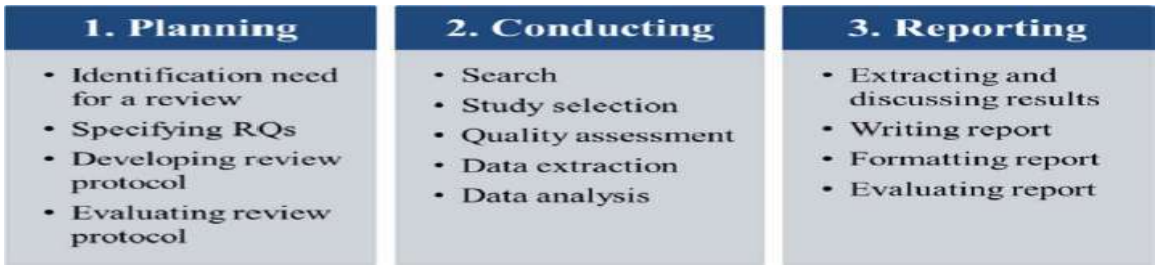

Figure 1: Phases of SLR [13]

3.1.4 Inclusion and Exclusion Criteria

In this study, the inclusion and exclusion criteria were carefully established to determine which research papers should be considered for analysis. The inclusion criteria consisted of the following requirements:
(I1) The research papers must be under peer review, ensuring a level of quality and validity in the academic community. (I2) Only research papers published in the English language were considered, ensuring uniformity in language comprehension. (I3) The research papers must be related to specific search keywords, including "user," "involvement," "user involvement," "Scrum," "XP (Extreme Programming)," "agility," "agile," "requirement," "user story," "feature," and "requirement engineering." This ensured the relevance of the papers to the topic under investigation. (I4) The papers must fall into the category of empirical research papers or conference papers, providing a strong empirical basis for the study's analysis. (I5) Only research papers published between 2012 and 2022 were included, aiming to capture the latest developments and insights in the field of agile software development.
On the other hand, certain research papers were excluded from consideration based on the following exclusion criteria:
(E1) Research papers that did not primarily focus on agile software development were excluded, as they were not directly relevant to the research objective. (E2) Papers that did not discuss the topics of user involvement and requirement engineering in agile software development were excluded to maintain the study's thematic

consistency. (E3) Research papers that lacked full-text availability were excluded, as this would hinder a comprehensive analysis of their content. (E4) Papers whose findings had already been published were excluded, as this would lead to duplication of content and redundant information. (E5) Chapters, viewpoints, keynotes, editorials, comments, tutorials, and slide presentations without associated research papers were excluded to ensure that only considerable and comprehensive research works were considered.

By precisely applying these inclusion and exclusion criteria, the study aimed to gather relevant and recent research papers on the subject of user involvement and requirement engineering in the context of agile software development. This severe approach ensured that the selected papers met high-quality standards and contributed to the overall reliability and validity of the research findings.

3.2 Conducting the Review

In this section, we present the outcomes of our systematic search and information extraction process from relevant sources and databases. The following subsections detail the steps taken to identify relevant research papers related to our study's focus on user involvement and requirement engineering in the context of agile software development. Furthermore, we explain the methodology employed for data extraction and the subsequent analysis of the collected information.

3.2.1 Study Search and Selection

Following the search strategy detailed in section 3.1.2, we conducted searches on the selected electronic databases. The initial search produced a total of 2342 research papers, as illustrated in Table 1.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1 : Search Results

S/No	Database Name	Retrieved Results
1	Scopus	35
2	Web of Science	27
3	IEEE Xplore	10
4	Science Direct	2123
5	Springer Link	147
Total Retrieved Result		2342

The research paper selection process comprises two rounds: round 1 involves title scanning, and round 2 involves abstract scanning. During the title scanning round, we searched for the pre-defined search string, consisting of specific keywords, within the titles of the research papers. We then selected those titles that included the predefined search string and excluded all other research papers from further consideration in the selection process. After completing the title scanning round, we identified and included 133 research papers for further analysis, as presented in Table 2.

Table 2 : Summary of Title Scanning Round

S/No	Database Name	Title Scanning Results
1	Scopus	17
2	Web of Science	11
3	IEEE Xplore	03
4	Science Direct	81
5	Springer Link	21
Total Title Scanned Result		133

In the second round, referred to as the "abstract scanning round," we examined the abstracts of the initial 133 research papers for the pre-defined search string. This search string comprised keywords previously explained in section 3.1.3, including "user involvement," "requirement engineering," and "agile software development."

Following a careful review of the abstracts, we identified and retained a total of 24 research papers that directly aligned with our research objectives. During this process, we removed any duplicate papers and excluded those that did not contain relevant information related to our study. The resulting 24 research papers, as presented in Table 3, constitute the final selection for in-depth analysis and investigation in our research.

Table 3: *Summary of Abstract Scanning Round*

S/No	Database Name	Abstract Scanning Results
1	Scopus	04
2	Web of Science	06
3	IEEE Xplore	02
4	Science Direct	09
5	Springer Link	03
Total Abstract Scanned Result		24

3.2.2 Quality Assessment

In our research, we employed a quality checklist derived from a previous study conducted by E.M Schon [13] to evaluate the individual studies that were included in our analysis. This checklist consisted of five distinct items, each offering three possible response options, as detailed in Table 4. The primary objective of utilizing this checklist was to assess the quality of the selected studies based on specific criteria, namely proposal validation, approach description, personal opinion, citation, and clarity of study aims.

By employing this checklist, we wanted to critically evaluate the included studies and ascertain their overall quality. This rigorous evaluation process helped us ensure the reliability, validity, and robustness of the research papers selected for our analysis. It allowed us to make informed and well-founded conclusions based on the credibility and methodological soundness of the individual studies.

3.2.3 Data Extraction and Analysis

In accordance with the guidelines proposed by Kitchenham and Charters [28], we developed a data extraction form to systematically gather relevant information from the included research papers. To simplify this process, we employed Mendeley, a reference management software, which allowed us to highlight important text passages and assign ratings. Mendeley also facilitated the extraction of data, based on predefined attributes outlined in the research protocol, ensuring a structured and organized approach for further analysis.

The extracted data encompassed crucial details about each research paper, such as the title, authors, publication date, DOI (Digital Object Identifier), and URL (Uniform Resource Locator) for proper identification and referencing. Additionally, we recorded publication-related information, including journal or conference names, publication or presentation dates (for conferences), publisher

details (if applicable), volume and issue numbers (for journals), page numbers, keywords, and abstracts describing the paper's content. Access to this comprehensive information proved essential for understanding the context and relevance of each research paper and for citation and future reference purposes.

To systematize the data collection further, the authors designed a data collection form in Excel, which facilitated the extraction of various key elements from the included studies. This form captured crucial study characteristics, such as research questions, research methods, participant details, sample sizes, data collection methods, and analysis techniques. Furthermore, it allowed us to record the key findings and conclusions of each study, along with scores from the quality checklist used for quality assessment. Additionally, the form provided space for any relevant notes or observations that might be valuable during the analysis or interpretation of the results.

The use of this structured data collection form in Excel enabled the authors to organize and manage the extracted information efficiently, facilitating the subsequent analysis and synthesis of the research findings.

During the data extraction process, we considered all identified papers. However, due to the varying reporting styles of some studies, it was not always possible to extract data in strict adherence to the predetermined format. Nonetheless, the systematic approach ensured that relevant and relevant data were gathered, contributing to the overall rigor and validity of our analysis.

4. RESULTS

In this study, we incorporated 24 relevant research papers that align with our research objectives. Initially, we offer a comprehensive description of the characteristics of these studies, presenting quantitative data such as the publication channel, research method, and overall quality assessment. This description allows for a thorough understanding of the selected papers and their key attributes.

Subsequently, we present our findings that are relevant to the research questions under investigation. These findings are derived from the systematic analysis of the included research papers, providing valuable insights and contributing to the overall understanding of the subject matter. Our presentation of the findings is organized and structured to ensure clarity and coherence in addressing the research questions at hand. Through this approach, we aim to contribute meaningfully to the existing body of knowledge and foster further discussions and advancements in the field of study.

4.1 Summary of Studies

In terms of publication channels, the research papers included in our study were either presented in conferences or published in scientific journals. Among all the studies that were included, a significant majority of 16 papers (66.67%) were found to have been published in scientific journals, while a smaller number of 8 papers (33.33%) were presented in conferences. This distribution indicates a notable preference for the distribution of research findings through scientific journals rather than conference publications, as depicted in Table 6.

Table 4: Quality Assessment

Item	Assessment Criteria	Score	Description
QA1	Is the proposal validated?	-1	No, it is not validated
		0	Partially, some aspects of the proposal have been verified in a laboratory
		1	Yes, proposal has been validated or confirmed through a research method.
QA2	Does the study present a detailed description of the approach?	-1	No, details are missing
		0	Partially, if you want to use the approach, you need to read the references
		1	Yes, the approach can be used with presented details
QA3	QA3 Does the study present a personal opinion piece or viewpoint?	-1	Yes, it does.
		0	Partially, since related work is explained and paper is set into a specific context
		1	No, the paper is based on research

QA4	Has the study been cited by other authors?	-1	No, no one cited the study
		0	Partially, between 1-5 articles cited the study
		1	Yes, more than 5 articles cited the study
QA5	Includes the paper a clear statement of the aims of the study?	-1	aims are not described.
		0	Partially, aims are described but unclearly
		1	Yes, aims are well described and clear

Table 5: Distribution According To Research Methods

Research Method	Paper total	Percentage
Model Driven	1	4.76%
Delphi study	1	4.76%
Metamodel	1	4.76%
Systematic Mapping study	1	4.76%
Grounded theory	1	4.76%
SLR	5	23.81%
Mixed method	4	19.05%
Qualitative	3	14.29%
Case study	3	14.29%
Survey	1	4.76%

Table 5 provides an insightful representation of the distribution of research methods employed in the investigated studies, along with the corresponding frequencies expressed as percentages. Notably, the most dominant research method was "SLR," utilized in approximately 23.81% of the papers. Following closely were the “Mixed methods”, "Qualitative" methods, and “Case study” implemented in 19.05%, and 14.29% of the papers, respectively.

Conversely, the remaining research methods, including "Survey," "Model Driven," "Delphi study," "Metamodel," "Systematic Mapping study," and "Grounded theory," were employed less frequently.

Table 5's comprehensive depiction of the distribution of research methods offers valuable insights into the dominant approaches and techniques adopted by researchers in their respective analyses. This information is instrumental in understanding the research design and methodological choices made by the authors, thereby enhancing readers' comprehension of the studies' overall methodologies.

Table 6: Distribution Of Studies According To The Publication Channel

Database Name	Journal	Conference
Scopus	2	2
Web of Science	4	2
IEEE Xplore	-	2
Science Direct	7	2
Springer Link	3	-

In conclusion, the study highlights that research on the user involvement approach in agile software development primarily takes place within real-life contexts, closely reflecting the established work practices of software companies. However, it is crucial to recognize that findings from individual case studies may not be directly transferable to other settings, necessitating caution in the interpretation of the results.

For the evaluation of each study, we utilized the quality checklist presented in Table 4. The outcomes of the quality assessment are graphically represented in Figure 2, providing a comprehensive overview of the overall quality of the included research papers. This systematic evaluation process ensures the reliability and credibility of the findings and contributes to the robustness of our research analysis.

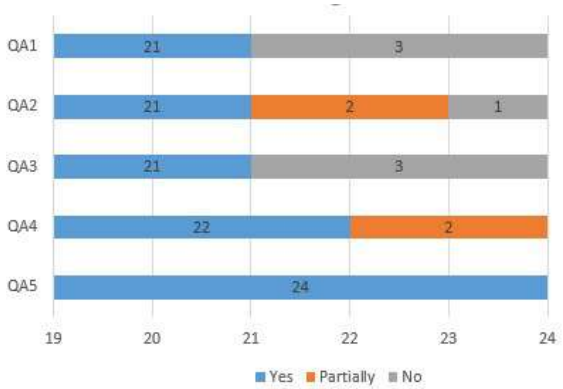

Figure 2: Quality Assessment

The quality assessment of the 24 reviewed papers was conducted based on five distinct criteria

(QA1-QA5). For QA1, which evaluated whether the research proposals were validated, 21 out of the 24 papers utilized a research method to validate their proposals, indicating a strong adherence to rigorous research practices. However, three papers lacked such validation.

Regarding QA2, which assessed the extent of approach description in the papers, 21 out of 24 studies provided comprehensive explanations of their approaches, with sufficient details to enable other researchers to replicate the methods. Nonetheless, two papers required additional references for a complete understanding, and one paper lacked important details in its approach description.

For QA3, which examined the presence of a personal opinion or viewpoint in the studies, 21 out of 24 papers were based on clearly defined research designs, while three studies did not sufficiently describe their research methods, suggesting a need for improved clarity in these cases.

QA4 relied on the number of citations from Google Scholar as of April 08, 2023, to measure the frequency of citations for each study. Among the 24 papers, 8 studies gathered more than a hundred citations, 12 received citations between 10 and 95, 2 were cited in 1-5 articles, and 2 paper received below 10 citations. It is worth noting that these citation metrics may have evolved over time since the publication of this systematic literature review.

Lastly, QA5 focused on the clarity of stated objectives in the papers. All 24 articles analyzed possessed well-defined and easily understandable research objectives, indicating a robust framing of the research questions.

In conclusion, 21 papers demonstrated adherence to all quality criteria, showcasing a high standard of research precision. However, it is crucial to acknowledge that the results might have varied at the time of the systematic literature review's publication, given the dynamic nature of citation patterns and scholarly impact over time.

4.2 (RQ1) What are the issues of the existing studies on managing users during requirement elicitation?

We have recognized five distinct issues associated with the management of users during requirement elicitation within agile software development. The subsequent comprehensive compilation of these issues summarizes the conclusions drawn from an analysis of 24 separate studies. Presented in Table 7 are precise insights into the occurrence frequency of each identified issue, along with references to the studies that addressed these concerns. The subsequent section elaborates upon the five issues relating to user involvement during the requirement elicitation phase within agile software development.

4.2.1 Lack of User Involvement

Agile methodologies propagate the assumption and authorization of user involvement, although in reality, this anticipation can prove to be impractical. Empirical investigations have proved that achieving user accessibility and availability is frequently challenging [24]. Agile software development places a strong emphasis on continuous collaboration and effective communication between the development team and stakeholders, which includes end-users. Nonetheless, the lack of user involvement within the context of agile software development can give rise to a range of adverse outcomes. These may include restricted input, misconstrued requirements, incompetent progress, and a decrease in user satisfaction [29]. While acknowledging the potential of user involvement in requirement elicitation, the actual presence of users is often constrained, impeded by variables such as temporal constraints, financial considerations, and workload burdens. Consequently, numerous agile teams opt to involve proxy users, demonstrated by product owners, who stand in for the user's role [30]. Another strategy involves the deployment of an "onsite developer," a representative stationed at the user's premises.

Bano [3] likewise documented instances where personnel absences, attributed to diverse forms of leave, led to project setbacks and complications in user involvement. According to an empirical investigation involving a health software developer from the United States [22], it was found that the most proficient users often find themselves occupied in organizational commitments or are subject to constraints imposed by superiors, impeding full-time involvement in the development team. The developer underscored that even recruiting these skillful users on a part-time basis can still present scheduling challenges. In summation, the developer struggles that recruiting highly skilled users for development initiatives can be hard due to organizational constraints and demanding agendas.

Table 7: Summary of the Issues and The Respective Studies That Have Investigated them

Issues	Freq.	Studies that reported the issue
Lack of User Involvement	3	Hugo Ferreira Martins (2019), Muneera Bano (2015), Irum Inayat (2015)
Lack of User's Knowledge	3	Vandana Gaikwad (2017), Raffaele Fabio Ciriello (2022), Muneera Bano (2015)
Lack of Product Owner's Expertise	3	Eva-Maria Schön (2017), Julian M. Bass (2014), Muneera Bano (2015)
Lack of Practical Guidelines	2	Sezin Yaman (2020), Susanna Martikainen (2014)
Reluctant Users	1	Muneera Bano (2015)

4.2.2 Lack of User's Knowledge

The absence of user proficiency within the expertise of agile software development presents challenges in understanding user needs, providing relevant input, and involving effectively in the developmental process. When users possess insufficient understanding of the technology or the developmental procedures, they may encounter difficulties in adequately expressing their requirements or offering actionable insights to the development team [21]. This can lead in a product that incompletely aligns with user demands or a development process that exhibits diminished efficiency and effectiveness [14].

To address this issue, it proves beneficial to enhance users' familiarity with agile methodologies, regulate their anticipations regarding incremental deliverables, and stimulate their active involvement in refining requirements and substantiating functionalities through product demonstrations [31]. Furthermore, the involvement of users spanning diverse levels of expertise can contribute to ensuring that the product accommodates a wide-ranging user demographic.

4.2.3 Lack of Product Owner's Expertise

Within the Scrum methodology, three roles emerge: the self-organizing team, the scrum master, and the product owner [25]. The notable import is the role of the product owner, tasked with facilitating seamless communication between user and development team throughout the software development process, as mentioned by Hoda [18], the product owner shoulders the responsibility of creating and overseeing the product backlog, which is a collection of user stories outlining the project's requirements. Nevertheless, certain product owners may find themselves deficient in proficiency concerning optimal practices in requirements engineering [32].

Incompetent product ownership can result in end-user representatives being unaware of user pain points or incorrect users being involved in testing early UI prototypes, leading to conflicts and potential project failure [13].

A proficient product owner should grasp the import of the system and its integration within the business processes, and should possess basic computer competencies. Nonetheless, practitioners operating within specific contexts have encountered impediments, with some product owners proving inadequate in providing punctual feedback and requirements, while others have struggled with representatives lacking knowledge of agile practices [33].

4.2.4 Lack of Practical Guidelines

The absence of a well-defined procedure for involving users in software development undertakings can lead to confusion and extension. Furthermore, bureaucratic impediments, exemplified by the requisites for obtaining permissions to involve users, possess the potential to impede the development process [34]. In agile software development, a pronounced emphasis is placed on integrating users across the development process [2]. In response to this, agile teams have the capacity to institute a user involvement protocol, clarifying the respective roles and obligations of both users and the team throughout the development process [35]. This guideline includes various facets, including the collection of user requirements, and solicitation of user input during

key project phases. Furthermore, the guideline can define the modalities through which user involvement will be facilitated, encompassing mechanisms such as user workshops, interviews, or surveys [14].

The establishment of a well-defined procedural framework and directives for user involvement plays a pivotal role in ensuring the comprehensive integration of user requirements across the development process, consequently enhancing the prospects of project success.

4.2.5 Reluctant Users

The absence of user motivation constitutes a notable challenge within the domain of agile software development. The principles of agility underscore the importance of fostering collaboration and open communication between development teams and users, thereby ensuring the alignment of software with user requirements [36]. However, the absence of user motivation poses a hindrance to attaining this level of collaborative involvement and effective communication, potentially resulting in a product that falls short of meeting user needs.

The presence of user motivation employs a favorable influence on both the degree and scope of user involvement in the developmental process. To tackle this difficulty, it becomes imperative to furnish users with feedback and acknowledgment for their contributions. This objective can be realized by recognizing their insights and feedback, providing regular updates on the progression of the software, and showcasing their contributions within the final product [37]. By cultivating user motivation, agile teams can enhance user involvement and enhance the prospects of creating a software product that appropriately satisfies user requirements.

4.3 (RQ2) How product owner can facilitate issues in managing users during requirement elicitation?

We have conducted a comprehensive analysis of 24 studies and identified 7 effective strategies for managing users during requirement elicitation within agile software development. These strategies, including mind map, user interface mockups, workshop, hybridism, face-to-face meetings, continuous delivery, and training and learning are derived from the findings of the analyzed studies. Implementing these strategies can assist software practitioners in addressing the challenges associated with user involvement during requirement elicitation within agile software development.

Table 8 provides a breakdown of the frequency at which these strategies were observed in the context of managing users during requirement elicitation in Agile software development. The table also indicates the studies that reported on each strategy. In the following paragraphs, we will describe each of these seven strategies in detail.

4.3.1 Mind Map

Fernando Wanderley [38] suggests using mind maps as a visual representation of requirements to enhance learnability and facilitate communication among users. Mind maps have gained recognition in both academia and industry as an effective technique and tool for managing requirements elicitation in agile software development [39] emphasizes the importance of visualization in the early stages of requirements elicitation to keep users engaged and involved, thereby reducing the semantic gap between domain experts and requirements engineers.

The cognitive approach of mind mapping enables individuals to work collaboratively based on their thought processes. Mind mapping is a well-established psychological method that allows for the visualization of concepts on paper or other mediums, aiding in the exploration of ideas in detail [40]. This technique generates multiple plans to arrive at the best ideas, words, tasks, or modules in software engineering. By leveraging mind maps as an additional tool in agile requirements elicitation, the process can be enhanced.

In summary, the mind map technique serves as a valuable tool for managing users during requirement elicitation within agile software development. It promotes collaboration, brainstorming, communication, organization, and visualization of key concepts.

4.3.2 User Interface Mockups

Agile methodologies recommend the use of concise requirements specifications such as User Stories to streamline the requirement elicitation process. However, for gathering presentation and interaction requirements that cannot be effectively captured through standalone stories, it has become common practice in the industry to combine User Stories with User Interface mockups [41]. These mockups serve as an intermediary language between users and developers, enabling the

descriptions of concrete requirements to users and providing technical UI descriptions to developers.

User interface mockups offer a visual representation of the requirements, making it easier for users to understand and provide feedback [42]. This helps in clarifying requirements and reduces misinterpretation. By visualizing the design and layout, UI mockups contribute to creating a better user experience that is intuitive and user-friendly. Users can provide feedback on the mockups, and the design team to make necessary adjustments to meet the user's needs [41]. Involving users early in the development process through UI mockups can significantly reduce development time. The iterative nature of incorporating user feedback on the mockups helps ensure that the final product aligns with user expectation, mitigating the risk of developing a product that falls short of meeting user needs [43].

In conclusion, UI mockups serve as a powerful tool for managing users during requirement elicitation in an agile environment. By visualizing requirements, facilitating collaboration, enhancing user experience, and reducing development time, UI mockups contribute to delivering a product that effectively meets the user's needs.

4.3.3 Workshop

In agile software development, workshops can be a valuable tool for managing user involvement during requirement elicitation process. A workshop is an interactive session that brings together stakeholders, including users, to collaboratively work on defining the requirements of a software product [44].

Table 8: Summary of Strategies and the Respective Studies That Have Investigated Them

Strategies and techniques	Freq.	Studies that reported the strategies and techniques
Mind Map	2	Fernando Wanderley (2014), Hina Saeeda (2020)
User interface mockups	2	José Matías Rivero (2013), Sezin Yaman (2020)
Workshop	2	Eva-Maria Schön (2017), Primadhika Marnada (2021)
Hybridism	4	Eva-Maria Schön (2017), Jorge Sedeño (2017), Hina Saeeda (2020), Puji Rahayua (2016)
Face to face meeting	6	Vandana Gaikwad (2017), Ulrike Abelein (2013), Irum Inayat (2015), Susanna Martikainen (2014), Yehia Ibrahim Alzoubi (2015), Muneera Bano (2015)
Continuous Delivery	2	Stephan Krusche (2014), Primadhika Marnada (2021)
Training and learning	2	Primadhika Marnada (2021), Carlos Tam (2020)

Harbers and colleagues [44] conducted research on how a Value Story workshop can be used in the user story elicitation process. The objective of this workshop is to incorporate user values into the requirements engineering process. To achieve this, it is necessary to identify both direct and indirect users. The workshop proceeds by revealing the values of each stakeholder group. It then provides specific scenario for each value, allowing the analysis of the user needs in those situation [13].

Overall, the workshop technique serves as a powerful tool for managing user during requirement elicitation in agile environment. By promoting collaboration, active involvement, rapid and iterative development, and facilitation.

Workshops can significantly contribute to ensuring that the final product meets the needs of the users. By actively involving users in the process, workshops enhance user involvement and foster a sense of ownership among users, resulting in a product that better aligns with their expectations and requirements.

4.3.4 Hybridism

A hybrid approach that combines both agile and non-agile techniques can offer an effective strategy for requirement elicitation in agile software development. By integrating non-agile techniques,

software engineers can generate better ideas and supplement the agile process with additional methods and approaches, even within the constraints of limited resources and time. The key is to ensure that the hybrid methodologies provide a simple and clear process for users to express their needs and enable maximum involvement in the project [40].

Bellucci et al. used a combination of eXtreme Programming and co-design sessions to develop a product that involved users heavily in the process, Harbers et al. [44] conducted research on the use of a Value Story workshop for the purpose of eliciting user stories in the requirements elicitation process, Lee et al. emphasized the importance of an Agile-User Centered-Design (UCD) specialist (AUS) who is responsible for facilitating communication between developers and User Experience (UX) designers to bridge the gap between the two roles, and Kautz investigated the extent of customer and user involvement in agile software development using Participatory Design methods.

The study by Hina Saeeda [40] demonstrates that hybrid methodologies can enhance the requirements elicitation process in agile software engineering. By combining two domain techniques, Joint Requirement Development (JRD) (non-agile) and mind maps (non-software), the study addressed challenges related to requirements elicitation in agile methodologies.

In conclusion, integrating non-agile techniques with agile methodologies can effectively manage user involvement during requirement elicitation within agile software development. By incorporating non-agile approaches such as co-design sessions, value story workshop, UCD, JRD, and mind map into agile software development, teams can ensure that the product development remains user-centric and meets the user needs while still adhering to the agile principles of flexibility and adaptability.

4.3.5 Face-to-Face Meeting

Face-to-face communication with users during requirement elicitation is highly emphasized in agile methodology. Agile values, direct interactions and recognizes the significance of effective collaboration and understanding between the development team and users. This approach prioritizes user stories over extensive specification documents, promoting minimal documentation. Frequent face-to-face communication enables users to provide feedback and steer the project in alignment with their understanding of the requirements [24].

Regular meetings foster informal communication among users, facilitating the evolution of requirements in agile development. However, the frequency of communication depends on the availability and willingness of team members to participate. Some users may be more accustomed to traditional development methods and may initially struggle to or trust agile methods [45].

While face-to-face interactions are preferred by many for discussing requirements with users. However, for remote users, video calls via WebEx are the preferred mode of communication [31]. If direct interaction is not possible, remote studies and surveys are used for eliciting requirements. The process of eliciting preliminary requirements usually involves multiple rounds of discussions with users and is not completed in the initial meeting. When multiple stakeholders are involved on the user side, prioritization and consensus building over the requirements are achieved by analyzing inputs obtained from one-to-one interactions with users. The final decision on requirements is made by the product owner [46].

Studies, such as the one by Ramesh et al. [47], indicates that increased face-to-face communication can reduce ambiguities in project requirements and minimize the need for maintaining extensive documentation. This suggests that face to face communication and collaboration between users and developers can lead to better understanding and clarity in project requirements, thereby reducing the reliance on detailed documents.

4.3.6 Continuous Delivery

Jez Humble and David Farley [48] defined Continuous Delivery (CD) as a set of practices and principles aimed at accelerating and increasing the frequency of software releases. Their definition emphasizes the goal of releasing software quickly and often, and highlights the importance of implementing specific practices and principles to achieve this objective.

Continuous Delivery builds upon the workflows and techniques used in build and test automation, known as Continuous Integration. While continuous integration focuses on automating the build process on a central server, CD takes it a step further by automating all workflows necessary for testing and deploying a new build. This includes the automating processes such as testing, packaging, and deploying new software releases, with the aim of streamlining the entire software delivery pipeline

from development to production. Continuous delivery's focus is to deliver software frequently and efficiently while maintaining high quality and minimizing risk [49].

In Agile methodology, development teams can deliver software not only at the end of a sprint but also during a sprint, whenever feedback is needed. This event-based approach enhances communication between developers and users by shortening feedback cycles. It also enables continuous user involvement, especially when there are enough test users available, such as in a beta test. This means the development team can receive feedback and make necessary changes to the software more quickly and efficiently, resulting in a better end product that meets user needs [49].

4.3.7 Training and Learning

According to Misra et al. [50], training and learning are crucial for the success of agile software development practices, especially in large-scale agile transformations. Proper training is identified as a critical factor for success. Teams without adequate training may struggle to effectively do agile practices. On the other hand, agile practices do not necessarily require formal training for knowledge sharing. Instead, they emphasize the importance of mentoring and professionally guided discussions, which are believed to yield better results. Agile practices promote a learning process through continuous experimentation, as they do not have a rigid "how-to" [4].

There is consensus regarding the importance of training and learning in agile software development. It is suggested that a well-trained team is better equipped to manage user involvement, prevent misunderstandings, and ultimately achieve superior results [4]. They enable teams to understand user needs, improve communication, and employ effective user involvement during requirement elicitation within ASD.

5. DISCUSSION

Within this section, we offer an explanation of the outcomes derived from our SLR, which originate from a comprehensive examination of 24 relevant studies. Our discussion commences with an exploration of the import and relevance of the findings relating to our specified research questions. Subsequently, we undertake an explanation of the limitations and boundaries inherent in our study's scope and methodology.

5.1 Meaning of Findings

5.1.1 General Findings

The outcomes of the SLR illuminate the challenges and strategies associated with user involvement during requirement elicitation within agile software development. It is obvious this is an important research area with a wide range of aspects studied in literature, highlighting the complexity and cross-functional nature of this field. Furthermore, the close relationship between this research field and real-life work practices in software companies is notable, as many studies analyzed aspects in practical contexts. Overall, the paper provides valuable insights into the challenges and best strategies for managing users during requirement elicitation. We emphasize the necessity of adopting a systematic approach to ensure the success of the requirement elicitation process. This paper serves as a useful guide for product owners and other stakeholders involved in the requirement elicitation process, offering actionable insights to enhance the effectiveness of the process.

5.1.2 Findings Related to RQ1

In response to the first research question, our investigation unveiled that the dominant body of literature concerning the management of users during requirement elicitation lacks a distinct concentration and is widely scattered. Predominantly, the extant studies emphasize the obstacles and hindrances encountered throughout this procedural phase. Based on our SLR, we identified 5 prominent challenges of managing users during requirement elicitation within agile software development. The most frequently reported challenges include lack of user involvement, lack of user's knowledge, and lack of Product Owner's Expertise. These challenges can impede software development practitioners from effectively managing users during requirement elicitation within agile context. However, there is a lack of systematic approaches and frameworks to address these challenges.

5.1.3 Findings Related to RQ2

To address the second research question, we suggest that the product owners can facilitate the management of users during requirement elicitation by employing various strategies identified in the investigated studies. These strategies include Mind Map, User interface Mockups, Workshops,

Hybridism, Face to face Meetings, Continuous Delivery, and Training and Learning. Based on our findings Face to face meetings and hybridism is the most used strategies to involve users in the process of requirements elicitation within agile software development. Additionally, we recommend that product owners possess strong leadership skills to manage user expectations and resolve conflicts that may arise during the process.

5.2 Limitations of the Review

Although the systematic literature review in this study aimed for comprehensiveness, there's a possibility that not all studies relevant to user involvement during requirement elicitation were included. The extensive amount of published literature and the specific search terms or databases used might have led to the exclusion of some relevant studies. Nevertheless, we followed a strict search strategy and followed to a predefined plan to ensure the study's thoroughness. The selection of research papers was primarily carried out by the paper's first author, who is a PhD student. This approach could introduce subjectivity into the selection process. Yet, whenever challenging decisions arose, the first author consulted with other authors to minimize subjectivity.

Another potential limitation of our review is the potential bias towards certain types of studies or sources, which could impact the validity of the findings. For example, if the search terms were confined to a specific field or language, relevant studies from other areas or languages might not have been considered. As a result, the findings of the review may not be applicable to all situations, as different organizations and industries could have unique requirements and challenges related to user involvement.

We acknowledge that some aspects of the reviewed studies, such as techniques and artifacts, might not have been adequately documented, potentially influencing the obtained results. To mitigate this concern, we conducted a comprehensive quality assessment of the studies included. However, it's plausible that if the studies had been reported more accurately, our findings could have differed.

6. CONCLUSIONS AND FUTURE WORK

The results obtained from the systematic literature review provide valuable insights into the challenges and strategies associated with managing users during requirement elicitation within agile software development. The study aimed to address two primary research objectives: understanding the issues of existing studies and identifying how product owners can facilitate effective management of users during this process.

In terms of the first research objective (RQ1), the review highlighted five key challenges related to user management during requirement elicitation within agile software development:

1. **Lack of User Involvement:** Despite agile methodologies advocating for user involvement, the actual presence of users can be constrained by various factors, leading to adverse outcomes such as misconstrued requirements and decreased user satisfaction.
2. **Lack of User's Knowledge:** When users lack proficiency in agile software development, they may struggle to express their requirements effectively, leading to misaligned products and diminished efficiency.
3. **Lack of Product Owner's Expertise:** The role of the product owner is important for effective communication between users and development teams. However, inadequate product owner expertise in requirements engineering can lead to misunderstandings and conflicts.
4. **Lack of Practical Guidelines:** The absence of well-defined procedures for user involvement can result in confusion and delays in the development process. Establishing clear guidelines for user involvement can alleviate these challenges.
5. **Reluctant Users:** User motivation plays a significant role in the success of agile software development. Unmotivated users can hinder collaboration and effective communication, potentially leading to products that do not meet user needs.

The second research objective (RQ2) focused on strategies that product owners can employ to facilitate effective user involvement during requirement elicitation:

1. **Mind Map:** Visual representations of requirements using mind maps can enhance collaboration, communication, and understanding among users and development teams.
2. **User Interface Mockups:** Creating mockups of the user interface helps users and developers better visualize

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

requirements, reducing misinterpretation and improving user experience.

3. **Workshops:** Interactive workshops involving stakeholders, including users, promote collaboration, active involvement, and consensus building, resulting in better-defined requirements.
4. **Hybridism:** Combining agile and non-agile techniques can provide a comprehensive approach to requirement elicitation, leveraging the strengths of both methodologies.
5. **Face-to-Face Meetings:** Direct communication between users and developers through face-to-face meetings fosters clearer understanding, rapid feedback, and iterative development.
6. **Continuous Delivery:** Embracing continuous delivery practices allows for frequent user feedback and involvement, leading to quicker adjustments and a better-aligned final product.
7. **Training and Learning:** Proper training and learning opportunities for development teams can enhance their ability to manage user involvement effectively and ensure a user-centric approach.

The systematic literature review sheds light on the challenges and strategies involved in managing users during requirement elicitation within agile software development. The identified challenges underscore the importance of addressing user involvement comprehensively, while the suggested strategies offer practical ways for product owners and development teams to enhance their processes. By acknowledging these findings and implementing the recommended strategies, software development practitioners can navigate the complexities of user involvement and improve the quality of their products.

In conclusion, this review underscores the need for more empirical studies focusing on managing users during requirement elicitation within agile software development, considering different project settings, such as different agile methodologies, scaling, or geographical distance among team members. It is also important to investigate the impact of cultural and organizational factors on user involvement during requirement elicitation as well as explore the roles of other stakeholders, including developers and project managers, in facilitating effective user involvement. Addressing these research gaps will further enhance our understanding of how to effectively manage users during requirement elicitation and contribute to the success of agile software development projects.

REFERENCES:

[1] C. Tam, E. Jóia, T. Oliveira, and J. Varajão, "International Journal of Project Management The factors influencing the success of on-going agile software development projects," *Int. J. Proj. Manag.*, vol. 38, no. 3, pp. 165–176, 2020, doi: 10.1016/j.ijproman.2020.02.001.

[2] A. Hinderks, F. J. Domínguez Mayo, J. Thomaschewski, and M. J. Escalona, "Approaches to manage the user experience process in Agile software development: A systematic literature review," *Inf. Softw. Technol.*, vol. 150, no. September 2021, p. 106957, 2022, doi: 10.1016/j.infsof.2022.106957.

[3] M. Bano and D. Zowghi, "A systematic review on the relationship between user involvement and system success," *Inf. Softw. Technol.*, vol. 58, pp. 148–169, 2015, doi: 10.1016/j.infsof.2014.06.011.

[4] C. Tam, E. J. da C. Moura, T. Oliveira, and J. Varajão, "The factors influencing the success of on-going agile software development projects," *Int. J. Proj. Manag.*, vol. 38, no. 3, pp. 165–176, 2020, doi: 10.1016/j.ijproman.2020.02.001.

[5] J. He and W. R. King, "The role of user participation in information systems development: Implications from a meta-analysis," *J. Manag. Inf. Syst.*, vol. 25, no. 1, pp. 301–331, Jun. 2008, doi: 10.2753/MIS0742-1222250111.

[6] M. Bano, D. Zowghi, and F. Rimini, "User Satisfaction and System Success : An Empirical Exploration of User Involvement in Software Development User Satisfaction and System Success : An Empirical Exploration of User Involvement in Software Development Keywords :," no. September, 2016, doi: 10.1007/s10664.

[7] B. Atkin and M. Skitmore, "Editorial: Stakeholder management in construction," *Construction Management and Economics*, vol. 26, no. 6. pp. 549–552, Jun. 2008, doi: 10.1080/01446190802142405.

[8] A. Ahimbisibwe, R. Y. Cavana, and U. Daellenbach, "A contingency fit model of critical success factors for software development projects: A comparison of agile and traditional plan-based methodologies," *J. Enterp. Inf. Manag.*, vol. 28, no. 1, pp. 7–33, 2015, doi: 10.1108/JEIM-08-2013-0060.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[9] A. Manifesto, "Agile manifesto," 2001.

[10] N. Iivari, "'Constructing the users' in open source software development: An interpretive case study of user participation," *Inf. Technol. People*, vol. 22, no. 2, pp. 132–156, 2009, doi: 10.1108/09593840910962203.

[11] B. Ives, M. H. Olson, S. M. Science, and N. May, "User Involvement and MIS Success : A Review of Research USER INVOLVEMENT AND MIS SUCCESS : A REVIEW OF RESEARCH *," vol. 30, no. 5, pp. 586–603, 2010.

[12] K. Werder, "TEAM AGILITY AND TEAM PERFORMANCE – THE MODERATING EFFECT OF USER INVOLVEMENT," 2016.

[13] E. M. Schön, J. Thomaschewski, and M. J. Escalona, "Agile Requirements Engineering: A systematic literature review," *Comput. Stand. Interfaces*, vol. 49, pp. 79–91, 2017, doi: 10.1016/j.csi.2016.08.011.

[14] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, and S. Shamshirband, "A systematic literature review on agile requirements engineering practices and challenges," *Comput. Human Behav.*, vol. 51, no. January, pp. 915–929, 2015, doi: 10.1016/j.chb.2014.10.046.

[15] P. Heck and A. Zaidman, *A systematic literature review on quality criteria for agile requirements specifications*, vol. 26, no. 1. Springer US, 2018.

[16] A. Börjesson and L. Mathiassen, "Improving software organizations: Agility challenges and implications," *Inf. Technol. People*, vol. 18, no. 4, pp. 359–382, 2005, doi: 10.1108/09593840510633329.

[17] D. R. F. Ciriello, J. A. Glud, and K. H. Hansen-Schwartz, "Becoming agile together: Customer influence on agile adoption within commissioned software teams," *Inf. Manag.*, vol. 59, no. 4, p. 103645, 2022, doi: 10.1016/j.im.2022.103645.

[18] R. Hoda, J. Noble, and S. Marshall, "The impact of inadequate customer collaboration on self-organizing Agile teams," *Inf. Softw. Technol.*, vol. 53, no. 5, pp. 521–534, 2011, doi: 10.1016/j.infsof.2010.10.009.

[19] R. Kasauli, E. Knauss, J. Horkoff, G. Liebel, and F. G. de Oliveira Neto, "Requirements engineering challenges and practices in large-scale agile system development," *J. Syst. Softw.*, vol. 172, p. 110851, 2021, doi: 10.1016/j.jss.2020.110851.

[20] M. Larusdottir and J. Gulliksen, "The Journal of Systems and Software A license to kill – Improving UCSD in Agile development," vol. 123, pp. 214–222, 2017, doi: 10.1016/j.jss.2016.01.024.

[21] R. Fabio, J. Aagaard, and K. H. Hansen-schwartz, "Information & Management Becoming agile together : Customer influence on agile adoption within commissioned software teams," *Inf. Manag.*, vol. 59, no. 4, p. 103645, 2022, doi: 10.1016/j.im.2022.103645.

[22] H. F. Martins, A. Carvalho, D. O. Junior, and E. D. Canedo, "Design Thinking : Challenges for Software Requirements Elicitation," pp. 1–27, 2019, doi: 10.3390/info10120371.

[23] M. Bano, D. Zowghi, and F. da Rimini, "User satisfaction and system success: an empirical exploration of user involvement in software development," *Empir. Softw. Eng.*, vol. 22, no. 5, pp. 2339–2372, 2017, doi: 10.1007/s10664-016-9465-1.

[24] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, and S. Shamshirband, "A systematic literature review on agile requirements engineering practices and challenges," *Comput. Human Behav.*, vol. 51, pp. 915–929, 2015, doi: 10.1016/j.chb.2014.10.046.

[25] Scrum Guides, *Ken Schwaber*. 2020.

[26] E. M. Schön, J. Thomaschewski, and M. J. Escalona, "Agile Requirements Engineering: A systematic literature review," *Comput. Stand. Interfaces*, vol. 49, pp. 79–91, 2017, doi: 10.1016/j.csi.2016.08.011.

[27] B. Kitchenham and S. M. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering Guidelines for performing Systematic Literature Reviews in Software Engineering EBSE Technical Report EBSE-2007-01 Software Engineering Group School of Computer Science and Ma," no. October 2021, 2007.

[28] Kitchenham, "Guidelines for performing systematic literature reviews in software engineering," *Tech. Rep.*, vol. 4, pp. 5356–5373, 2007, doi: 10.1109/ACCESS.2016.2603219.

[29] M. Bano, *Alignment of Requirements and Services with User Feedback University of Technology , Sydney*. 2015.

[30] A. L. Lorca, R. Burrows, and L. Sterling, "Teaching motivational models in agile requirements engineering," *Proc. - 2018 8th Int. Work. Requir. Eng. Educ. Training, REET 2018*, pp. 30–39, 2018, doi: 10.1109/REET.2018.00010.

[31] V. Gaikwad and P. Joeg, "A case study in requirements engineering in context of agile,"

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Int. J. Appl. Eng. Res., vol. 12, no. 8, pp. 1697–1702, 2017.

[32] J. M. Bass, "How product owner teams scale agile methods to large distributed enterprises," *Empir. Softw. Eng.*, vol. 20, no. 6, pp. 1525–1557, 2015, doi: 10.1007/s10664-014-9322-z.

[33] M. Bano and D. Zowghi, "A systematic review on the relationship between user involvement and system success," *Inf. Softw. Technol.*, vol. 58, pp. 148–169, 2015, doi: 10.1016/j.infsof.2014.06.011.

[34] S. Yaman, F. Fagerholm, M. Munezero, T. Männistö, and T. Mikkonen, "Patterns of user involvement in experiment-driven software development," *Inf. Softw. Technol.*, vol. 120, no. December 2019, p. 106244, 2020, doi: 10.1016/j.infsof.2019.106244.

[35] S. Martikainen, M. Korpela, and T. Tiihonen, "User participation in healthcare IT development: A developers' viewpoint in Finland," *Int. J. Med. Inform.*, vol. 83, no. 3, pp. 189–200, 2014, doi: 10.1016/j.ijmedinf.2013.12.003.

[36] M. Bano and D. Zowghi, "A Systematic Review on the Relationship between User Involvement and System Success," *Inf. Softw. Technol.*, 2014, doi: 10.1016/j.infsof.2014.06.011.

[37] D. Zowghi and F. Rimini, "Problems and Challenges of User involvement in Software Development : an Problems and Challenges of User involvement in Software Development : an Empirical Study," no. April, 2015, doi: 10.1145/2745802.2745810.

[38] F. Wanderley, A. Silva, J. Araujo, and D. S. Silveira, "SnapMind: A framework to support consistency and validation of model-based requirements in agile development," *2014 IEEE 4th Int. Model. Requir. Eng. Work. MoDRE 2014 - Proc.*, pp. 47–56, 2014, doi: 10.1109/MoDRE.2014.6890825.

[39] A. Chen and J. Beatty, *Visual models for software requirements*. Microsoft Press., 2012.

[40] H. Saeeda, J. Dong, Y. Wang, and M. A. Abid, "A proposed framework for improved software requirements elicitation process in SCRUM: Implementation by a real-life Norway-based IT project," *J. Softw. Evol. Process*, vol. 32, no. 7, pp. 1–24, 2020, doi: 10.1002/smr.2247.

[41] J. M. Rivero, E. R. Luna, J. Grigera, and G. Rossi, "Improving user involvement through a model-driven requirements approach," *2013 3rd Int. Work. Model. Requir. Eng. MoDRE 2013 - Proc.*, pp. 20–29, 2013, doi: 10.1109/MoDRE.2013.6597260.

[42] S. Yaman, F. Fagerholm, M. Munezero, T. Männistö, and T. Mikkonen, "Patterns of user involvement in experiment-driven software development," *Inf. Softw. Technol.*, vol. 120, no. November 2019, p. 106244, 2020, doi: 10.1016/j.infsof.2019.106244.

[43] E. M. Schön, J. Thomaschewski, and M. J. Escalona, "Identifying agile requirements engineering patterns in industry?," *ACM Int. Conf. Proceeding Ser.*, vol. Part F1320, pp. 1–10, 2017, doi: 10.1145/3147704.3147733.

[44] P. Marnada, T. Raharjo, B. Hardian, and A. Prasetyo, "Agile project management challenge in handling scope and change: A systematic literature review," *Procedia Comput. Sci.*, vol. 197, no. 2021, pp. 290–300, 2021, doi: 10.1016/j.procs.2021.12.143.

[45] U. Abelein, H. Sharp, and B. Paech, "Does Involving Users in Software Development Really Influence System Success ?," pp. 17–23, 2013.

[46] Y. I. Alzoubi, A. Q. Gill, and A. Al-Ani, "Empirical studies of geographically distributed agile development communication challenges: A systematic review," *Inf. Manag.*, vol. 53, no. 1, pp. 22–37, 2016, doi: 10.1016/j.im.2015.08.003.

[47] B. Ramesh, L. Cao, and R. Baskerville, "Agile requirements engineering practices and challenges: an empirical study," *Inf. Syst. J.*, vol. 20, no. 5, pp. 449–480, Sep. 2010, doi: 10.1111/J.1365-2575.2007.00259.X.

[48] J. Humble and D. Farley, *Continuous delivery: reliable software releases through build, test, and deployment automation*. 2010.

[49] S. Krusche and L. Alperowitz, "Introduction of continuous delivery in multi-customer project courses," *36th Int. Conf. Softw. Eng. ICSE Companion 2014 - Proc.*, pp. 335–343, 2014, doi: 10.1145/2591062.2591163.

[50] S. C. Misra, V. Kumar, and U. Kumar, "Identifying some critical changes required in adopting agile practices in traditional software development projects," *Int. J. Qual. Reliab. Manag.*, vol. 27, no. 4, pp. 451–474, 2010, doi: 10.1108/02656711011035147/FULL/HTML.