

## Critical Success Factors and Challenges in Agile Requirements Engineering

Mohammed Saleh<sup>1</sup>, Fauziah Baharom<sup>2</sup>, Shafinah Farvin Packer Mohamed<sup>3</sup>

<sup>1,2,3</sup>Universiti Utara Malaysia, Malaysia,

<sup>1</sup>m.rawajfeh@yahoo.com, <sup>2</sup>fauziah@uum.edu.my, <sup>3</sup>shafinah@uum.edu.my

**Article History:** Received: 10 November 2020; Revised: 12 January 2021; Accepted: 27 January 2021;

Published online: 05 April 2021

---

**Abstract:** Nowadays, the requirements of the software are changing rapidly in order to meet clients' needs, which increases the complexity of developing software. Thus, Agile requirements engineering has arisen and it focuses on how to deal with the increasing changes in software requirements by gathering requirements iteratively and collaboratively with the clients. Thus, the clients' satisfaction could be met more easily. On the other side, researchers have tried to improve agile requirements engineering from time to time, however, there are still limitations and challenges faced, which need more attention. Thus, this study is conducted by performing a systematic literature review technique to investigate the challenges, critical success factors, and the topics that need more attention in the agile requirements engineering field. In the beginning, the study obtained 178 articles related to this topic which were published from 2002 until 2019. After a thorough analysis of the articles, the study reviewed the ten (10) challenges and the proposed solutions that mentioned in the previous studies. Besides that, the study found six (6) critical success factors, and highlighted four (4) topics that need more attention from the researchers in future studies in agile requirements engineering.

---

**Key words:** agile software development, agile requirements engineering, systematic literature review

---

### 1. Introduction

At present, software development grows into a more complex process and Agile Software Development (ASD) has been broadly used to handle the increase of complexity in the software development industry [1]. 12 principles are provided by ASD in order to meet the clients' satisfaction and to ensure on-time delivery [2]. Moreover, the agile manifesto promotes speedy delivery, teamwork, and self-organization, which can help practitioners to deal with complexity of requirements during software development. In addition to that, the fulfillment of the tasks focuses on the priority of the client. [3]. Therefore, attention can be given to the clients' needs and consequently their satisfaction can be obtained. On another hand, Clancy [4] mentioned in the report of the Standish group that among the top projects, 58 were eliminated because of the defective requirement, redundant requirements, or not compatible with clients' expectations. Even though the benefits of the Agile context of organizations software development, the role of the requirements engineering (RE) in the Agile context is still vague with the software development population, and it poses several new challenges that need investigation [5]. In conventional Requirement Engineering (RE), there are a number of activities which comprise of management, negotiation, documentation, elicitation, and validation. In the same way, documentation requirements, requirement modeling, requirement prioritization, user involvement, team collaboration, interviews, and exploration are proposed to be applied in Agile RE [3]. However, in agile RE these activities are not properly separated, which makes the RE activities unclear in Agile context [6].

On the other hand, Port, Olkov, and Menzies [7] mentioned that procedures are largely dependent upon the experience of the practitioner, and these procedures are informal and vague as well. It is hard to explain Agile RE [5]. Thus, it is difficult to be distinguished and explained by academicians and software developers [5]. Moreover, Inayat et al. [5] have determined that although Agile RE provides a number of practices such as user stories, prototyping, and quick feedback, they also showed there is still a need to explore further in Agile RE and their challenges [5]. Therefore, this study conducted to investigate the Agile RE by using the systematic literature review (SLR) approach. The aims of this study are to determine challenges, and proposed solutions of the Agile RE, as well as identifying critical success factors (CSFs) related to Agile RE, and finding the topics that need further attention from researchers in the Agile context.

The remaining of the paper is organized as follows: Section 2 discusses the relevant literature on Agile RE, continued with Section 3, which defines a method of the study. Section 4 provides the findings of this study. Section 5 provides the discussion. Finally, Section 6 provides the conclusive comments on the summary and future research.

### 2. Existing Systematic Literature Review Studies On Agile RE

In the area of software engineering, a number of Systematic Literature Review (SLR) are conducted to address the Agile RE. In a study by Inayat et al. [5], a comprehensive literature review was conducted to

determine Agile RE practices and potential challenges. They presented seventeen (17) practices of Agile context while dealing with requirements, and for the challenges of Agile RE, the study found eight (8) challenges which are little documentation, schedule estimation, and client availability, unsuitable architecture, ignoring non-functional requirements (NFRs), client inability and agreement, requirements change, and contractual limitations.

In another study, Schön et al. [1] conducted a literature review to put light on deep insights of Agile RE stakeholders, such as user involvement, user perspective, shared understanding, and documentation. The study obtained data from 27 related studies. Based on the analysis of these related studies, they found many problems arise with the direct involvement of stakeholders and users. Besides, the study also recognized the major artifacts for documentation of requirements, for instance, prototypes, user stories, scenarios, story cards, and use of cases. They also suggested that there is a need to further investigate Agile RE empirically, particularly by exploring the requirement management in different project types.

In addition, a literature review was also conducted by Elghariani and Kama [3] in order to examine the challenges and practices of agile RE. The study obtained data from 22 related studies. Their findings were quite similar to the findings of Inayat et al. [5] in which six (6) challenges of Agile RE, and sixteen practices (16) were found. Among these challenges are the change of requirements, maintainability, ignoring NFRs like security, inappropriate software architecture, client availability, project constraint, and missing requirements. .

Moreover, a mapping study was also conducted by Heikkila et al. [9] on Agile RE through the examination of 28 articles. According to these researchers, there exists a weak understanding of Agile RE. A number of challenges were suggested by them, and some advantages of Agile RE. Furthermore, Islam [11] conducted a literature review from 24 articles between 2001-2016, in order to find out CSFs in ASD. As a result, five success factors and twenty-four characteristics of the success were gathered from the articles. Likewise, literature was also done by Alam et al. [12] having more than 60 articles, in order to identify the weaknesses of the different stages of the Agile context. However, the scope of the study was considered very wide [13]. Besides, the SLR did not mention the selected publication year of the studies. In a study by Soares et al. [10], the study conducted a literature review, which specified the major challenges while applying agile RE. The study found 19 articles related to its research questions. The results of the study showed a number of challenges with applying agile RE such as NFRs determination, lack of information, the definition of requirements, and communication with clients. Table 1 shows a summary of SLRs that were conducted in Agile RE. The table contains the number of studies included in SLRs, the period of studies which determines the publishing years of studies, and the research questions of SLRs.

**Table 1.**Summary of SLR on Agile RE

References	No. of studies included	Publication year of the studies	Research questions
[5]	21 studies	2002 to 2013	RQ1. “What are the adopted practices of Agile RE according to published empirical studies?” RQ2. “What are the challenges of traditional RE that may get alleviated by Agile RE?” RQ3. “What are the challenges of Agile RE?”
[1]	27 studies	2007 to 2015	RQ1: “What approaches exist, which involve stakeholders in the process of RE and are compatible with ASD?” RQ2: “Which agile methodologies, which are capable of presenting the user perspective to stakeholders, can be found?” RQ3: “What are the common ways for requirements management in ASD?”
[3]	22 studies	2000 to 2015	RQ1: “What Are the Agile Requirements Engineering Practices?” RQ2. “What Are Agile requirements engineering challenges?”
[9]	28 studies	2004 to 2014	RQ1: “What has been researched

			regarding requirements engineering in an Agile context?" RQ2: "What are the reported key benefits of Agile requirements engineering?" RQ3: "What are the reported problems and corresponding solutions related to Agile requirements engineering?"
[11]	24 studies	2001 to 2016	RQ1: "What factors define success in Agile software development?" RQ2: "How do these factors contribute to success of a project?"
[12]	60 studies	-	RQ1: "What are requirement Engineering practices used in Agile?" RQ1: "What are issues, limitations and challenges in Agile requirement engineering?"
[10]	19 studies	2001 to 2014	RQ1: "What are the difficulties of using Agile requirements on software development projects?" RQ2: "What is the perception of the participants regarding the use of Agile requirements in software projects?" RQ3: "Is there any work relating Agile requirements to technical debt?"

After a thorough analysis of the existing studies on Agile RE SLRs, most of the literature reviews focused on the practices and processes in Agile RE, and there are limited literature reviews focused on the challenges and proposed solutions in Agile RE. Indeed, the RE in the Agile context is still vague of practitioners, and it poses several new challenges, which need to investigate it. Besides, there are scarce of literature reviews that focused on CSFs in Agile RE. In fact, there is only one (1) literature review focused on identifying the CSFs in ASD. However, the scope of the study covered ASD in general, which did not focus on the context of Agile RE. Indeed, identifying the CSFs can help the practitioners to increase success in future projects [14]. Thus, this study focuses on the challenges and the proposed solutions for these challenges, besides identifying the CSFs related to requirements in Agile context. Furthermore, this study focuses on the topics that need further attention in Agile RE. Finding the topics in Agile RE that need further attention can open the doors to further research in future studies.

### 3. Research Method

This study was conducted based on guidelines by Kitchenham and Charters [15]. Thus, the main steps of this study included the research framework, principles, and the review results matched the formalism advocated by Kitchenham and Charters [15].

#### 3.1. Research Questions

The research questions conducted by the study after reviewing the related work on Agile RE are as below:  
RQ1: What are the challenges, issues among them, and proposed solutions in Agile RE?  
RQ2: What are the critical success factors in Agile context related to requirements?  
RQ3: What are the topics need more attention related to Agile RE?

#### 3.2. Search Process

This study based on the literature review procedure which essentially depended on secondary data from the electronic databases and printed proceedings such as ACM, IEEE, Springer Link, Science Direct, ISI Web of Knowledge, Wiley Inter-Science, and Taylor & Francis ISI web of knowledge. In addition, a snowball technique is employed [16]. Also, DBLP known as Digital Bibliographic Library Browser was used to search for the author's publication. Moreover, this study used two parts in the search string namely S1 and S2. S1 denoted keywords related to "agile requirement engineering" and "agile requirements". S2 contains keywords such as "agile requirements challenges", "issues in agile requirements engineering", "critical success factors for agile requirements engineering", "success factors for agile requirements engineering", "practices in agile requirements

engineering”, “topic in agile requirements engineering, “gaps in agile requirements engineering”, “weaknesses in agile requirements engineering”, ‘difficulties in requirements engineering”.

### 3.3. Exclusion and Inclusion Criteria

The study focused on the articles that were written in the English language, thus, any article was written in another language was ignored. After using search techniques in databases and digital libraries. 2585 articles were found and reviewed by the authors. The criteria for inclusion and exclusion were applied. In the first step, titles, abstracts, and conclusions were used to exclude irrelevant articles. Besides, the duplication of articles was canceled during this step. Only 199 articles were left at the end of the first step. The articles which were not explaining the scope of topics properly were excluded in the second round of exclusion. In the end, this study found that only 178 articles were useful and related to the discussion topics. Table 2 shows the number of articles in the first and second rounds including inclusion and exclusion stages.

**Table 2.**Summary of inclusion and exclusion stage

Database	Initial Result	First Round		Second Round	
		Ex.	In.	Ex.	In.
IEEE Xplore	1654	155 7	97	8	89
ACM	159	126	33	2	31
Science Direct	361	334	27	5	23
Springer Link	289	268	21	2	19
ISI Web of Knowledge	72	60	12	3	9
Wiley Inter Science	18	12	6	1	5
Taylor & Francis	32	29	3	1	2
Total of Articles	2585	238 6	19 9	22	178

### 3.4. Criteria of Quality Assessment

The SLR based on the quality criteria suggested by Kitchenham and Charters [15], and also used by other SLRs in Agile RE such as Inayat et al. [5], in order to assess the quality of the chosen studies. The quality criteria are: (C1): Is the study target obviously determined?, (C2): Is the study context well handled?, (C3): Are the results obviously announced?. The response grading for C1, C2, and C3 are (Yes= 1, nominally= 0.5, No= 0). The study evaluated all articles based on these criteria of quality. The result of the first criterion (C1) was 87% of the articles. In the second criterion (C2), the result was 85% of the articles. As for the third criterion (C3), the result was 83% of the articles. At the end, the outcome of the quality criteria was positive for all questions. Table 3 shows a summary of the quality criteria and results.

**Table 3.**Summary of the quality criteria and results

QualityCriteria	Result classify	Result
(C1): Is the study target obviously determined?	(Yes= 1, nominally= 0.5, No= 0).	90%, 179 articles
(C2): Is the study context well handled?		88%, 175 articles
(C3): Are the results obviously announced?		89%, 177 articles

### 3.5. Data Extraction and Analysis

After the selection of 178 articles that are the most relevant. 57 articles were published in journals, and 36 articles in magazines, symposiums, and workshops, the remaining 85 articles were published in conferences. Figure 1 shows the percentages of articles types disseminated.

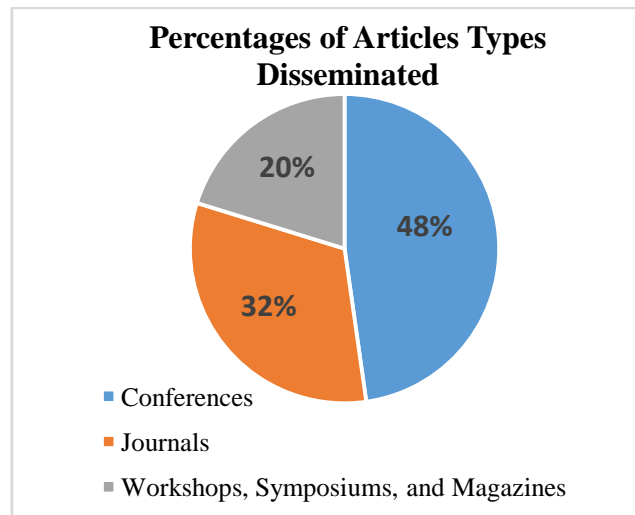


Figure 1. Distribution of articles types disseminated

Moreover, Figure 2 shows the distribution of articles based on the date published. The date starts from 2002 until 2019. The reason for selecting 2002, which is the date after the announcement of Agile manifesto in 2001. According to Figure 2, the number of articles conducted on this topic started to increase due to the increase in attention to ASD. However, the Agile RE still ambiguous for practitioners, even though there were several articles conducted on RE in agile context [17].

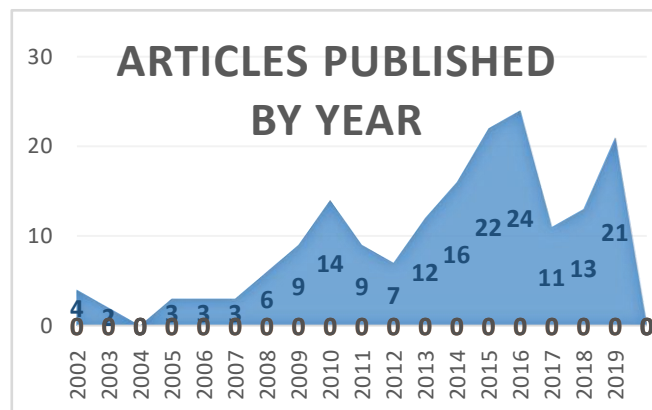


Figure 2. Distribution of the published articles

#### 4. Findings of the Review

The study presents the findings of search articles in the digital library database in Table 4.

Table 4. Summary of findings per database

Database	Mapping Date	Initial Results	Final Results
IEEE Xplore	2002 to 2019	1654	89
ACM		159	31
Science Direct		361	23
Springer Link		289	19
ISI Web of Knowledge		72	9
Wiley Inter Science		18	5
Taylor & Francis		32	2
Total of Articles		2585	178

According to Table 2, 89 articles were identified in IEEE and that around (50%) out of 178 articles. Then, Science Direct with 23 articles, Springer Link with 19 articles, ACM with 31 articles, ISI Web of Knowledge with 9 articles, Wiley Inter-Science with 5 articles, and Taylor & Francis only 2 articles. Figure 3 shows the distribution of the percentage of per database out of total included articles. Moreover, 178 articles are the total of all articles that investigated challenges and/or proposed solutions, and /or presented CSFs or/ and topics that need attention.

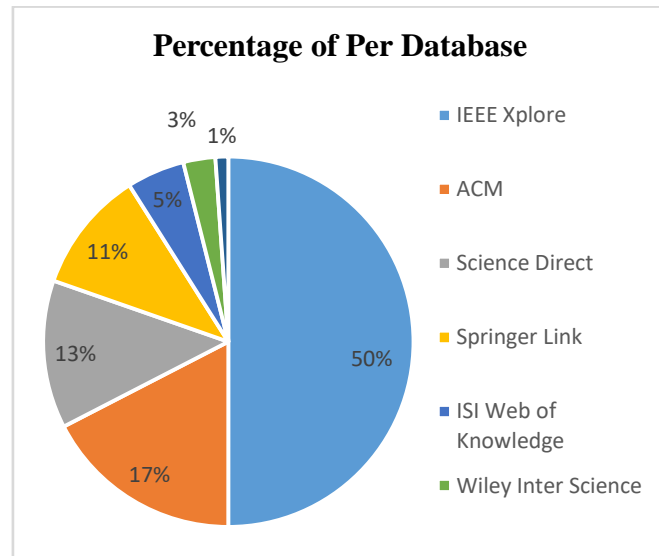


Figure 3. Distribution of percentage of per database out of total included articles

#### RQ1: What are the challenges, issues among them, and proposed solutions in AgileRE?

Requirements are the groundwork of all software products and it seeks to guarantee that client demands are rightly understood [9][18]. RE is one of the key software processes which determines how to gather, document, review, and achieve requirements [9][19]. In the classical world, when applying the waterfall method, requirements are explained perfectly and in-detail before beginning work on the design. However, in the agile context, the strategy is different. The requirements are mostly explained in a simple manner by producing user stories at the start of the Sprints, which is not enough to explain the requirements [17]. Thus, agile RE is facing several challenges [17]. Next paragraphs describe these challenges.

**1. Inappropriate architecture** that finalized by the developers in previous phases of software projects turns into unsuitable in final phases for additional requirements [20]. Furthermore, continuous change in code is an action between Agile members called refactoring. Nevertheless, ignoring refactoring during the development phase will add extra cost in the later phases [17] [14] [5].

**2. Client availability** is supposed by Agile methods, though, the real application of this assumption is questionable, as confirmed by the previous literature that client access and availability always be a challenge [21][20]. While no evidence found to contradict the argument that the requirement changes might be defined by directly the client [22]. For the process acceleration, the availability of the client is often challenging due to a number of reasons in the context of business, for instance, the client representative's workload and cost [22]. Practically, utmost the agile teams usually have substitutes or proxy clients to pretend as real clients [23]. Besides, most of the software firms are implementing the "onsite developer" strategy to make it easy for exchange between developers and clients [22].

**3. Little of documentation involves** is one of the characteristics of Agile methodologies for changing the traditional requirements documentation with direct-point for the client's goals through user stories [24]. The alteration from the documentation of traditional requirements with the direct requirement is the main challenge that methodologies of Agile demonstrate to the software developers. In some cases, whenever there is communication loss between developer and client then missing a small amount of documentation can be a major problem [17]. As mentioned by Cao and Ramesh [18] and Deneva et al. [23] as the complexity of the project increases becomes this challenge the worst.

**4. Accuracy of estimates** is a challenge that organizations are facing during the implementation of the Agile context. Although the practices of Agile methodology help to initiate the primary valuation of a project, whereas, the disadvantage of the implementation of Agile practices is that it is not able to make accurate valuations due to

unstable requirements [20], For instance, the project size usually based on available user stories, which may be not suitable in upcoming iterations [18].

**5. Requirements prioritization** is one of the significant parts for managing requirements in ASD, which performs an important role in the failure or success of any software [25]. Indeed, if the requirements are not prioritized at the appropriate time, the software product can go to fail [26]. The challenge of prioritization can occur during the continuous changing of requirements, and exclusion of unnecessary functions [25].

**6. Contractual limitations** after contract signing, volatility plays as an important role by not tolerating variations within the requirements, because these changes may highly increase the project cost and sometimes brings toward project failure. However, issues might handle by implementing strategies such as fixed payment on every release that helps to protect the investment, and also averts volatility of requirements [23]. Moreover, the elimination of incorrect and ambiguous requirements that occurs due to changes needs extra effort and financial cost. Hence, communication enhancement and client involvement may help to overcome this situation [23].

**7. Ignoring NFRs** is considered as a key challenge toward Agile RE and ASD [17] [20], and also the possible reason for the failure of the system and rework [17]. Indeed, NFRs are unnoticed during the early phases of Agile development. Furthermore, software developers spend extra effort on the FR, and NFRs are overlooked until later phases, for example, reliability, scalability, security, performance, and usability are most of the times handled later in a temporary manner between the testing stage of the system [8][17][20].

**8. Client inability and agreement** are shown in the literature as a major challenge. As Daneva et al. [23] explain that inability of the client defined as regarding the decision-making process and, involves the knowledge domain of the client. On another side, the possible solution if there is an agreement of client groups who are part of the project, this agreement among groups of the clients has a significant impact on performance, particularly in short-term cycles of development [20][23][8].

**9. Requirements changing** is considered as another important facet of Agile methodology. This explains that the Agile method's dynamic nature that enables to change, but it may cause trouble during consequences evaluation of the changes [28]. The recent development of a framework namely RE-KOMBINE has been developed to cope with para-consistent requirements specification [28], which permits the formal specification of requirements, which shows more flexibility to adjust changes. Moreover, another tool of Agile RE namely JIRA [29], which highly recommended using to address the challenging, projects [29].

**10. Missing requirements** are also considered as a challenge, especially, when using user stories to decrease the focus on requirements documentation. In this situation, the requirements can be missed mainly by a decline in the formalization of the requirements [30].

On another side, there is overlap among the challenges, for example, ignoring the NFRs are can lead to inappropriate architecture and imprecise effort estimation, because the user stories are in most cases are not enough to define NFRs in Agile RE [31]. Table 5 includes the challenges with their possible solutions proposed by the articles

**Table 5.** Agile RE challenges and proposed solutions

<b>Challenges</b>	<b>Description</b>	<b>Proposed Solutions</b>	<b>Reported Articles</b>
Inappropriate Architecture	The changing of requirements and ignoring NFRs in the early phases can become an inappropriate architecture [32].	Test-driven development (TDD) [18].	[3], [32], [5], [20]
Client Availability	The business perspective such as time, cost and workload of the clients can determine the client availability [22].	Replacement client [20], Proxy client [5]	[3], [5], [23], [33], [20]
Little documentation	The change to direct-point is a pivotal challenge in Agile RE and becomes worse whenever the increase of project complexity [20].	Prototyping [35]	[1], [3], [5], [23], [24], [20] [18], [34]

Accuracy of Estimates	Poor user story and ignoring NFRs can drive imprecise estimates. Besides, most agile estimation techniques based on expert estimation, the lack of experiences can produce also the imprecise estimates [31]	Expert Judgment, Story Points, COSMIC FP	[11], [3], [5], [31].
Requirements Prioritization	The continuous changing the requirements and the lack of documentation can cause prioritization problems [25].	Prioritization Techniques (AHP, Planning Game)	[25], [36], [20], [18].
Contractual Limitations	The contractual changes can increase in costs, time and sometimes failure of projects, therefore, legal measures should be taken to avoid such a situation [23].	Payment per release, increasing communication and involving clients [14],[23].	[20], [23]
Ignoring the NFRs	The user stories in most cases not enough to capture the NFRs, besides, Agile methodologies not have a special method or practices to elicit the NFRs [37].	Visual models NORMAP, NORMATIC [27], NERV [38].	[11], [3], [5], [37], [39],[20], [18]. [40]
Client Inability and Agreement	Incompetence of client in terms of decision-making, complete domain knowledge, and consensus of more than one client group involved in a project can produce the client inability and agreement [23].	Repeated communication [5] Iterative RE [20]	[9] [20], [18].
Requirements Changing	The flexible nature of Agile methodologies welcomes changes, but it can create trouble when evaluating the consequences of these changes [5]	RE-KOMBINE [28]	[11], [3], [5], [41], [29].
Missing Requirements	The little of documentation, continuous changing and the lack of client involvement can produce the lose requirement [30].	Test-driven development (TDD)[18].	[30], [42], [10], [43].

**RQ2: What are the critical success factors in an agile context related to requirements?**

CSFs define as the characteristics and elements that should take into account from the practitioners in the context of Agile RE in order to ensure success. After a thorough analysis of the related articles, there are six (6) factors namely the environment and culture of the organization, client participation, training, connection between stockholder, the grade of details, and team background (experience). Table 6 shows the CSFs identified by articles.

**Table 6.**CSFs in Agile RE

CSFs	Description	Reported Articles
Environment and culture of the organisation	This factor depends on how the environment and culture of the organisation create processes to solve the problems and every change produce problems [47].	[13], [48], [49], [47], [50], [51], [52].



Client participation	The team members should make sure the participation of clients included in all activities of Agile RE to avoid missing any requirements and reduce the fail of projects [13].	[13], [48], [49], [51], [53], [54].
Training	The high training for the teams to gathering team members with a mature knowledge can increase the probability of success to address the requirements challenges [49].	[49], [47], [55], [53], [56], [54].
Connection between stockholder	The poor connection between the stockholders may produce unsatisfied requirements by the clients and may even fail to deliver in time [56].	[47], [56], [54], [46], [57].
The grade of details	The grade of details for requirements has a significant impact on estimation, when the requirements have low level of details that may reduce the accuracy of the estimate and cost additional effort and time [13].	[13], [4], [31], [58].
Team background (experience)	The experiences of team members have a considerable influence while dealing with requirements, in this time, the team with expertise has a high potential to be a success in agile context [47].	[47], [53], [46], [52].

**RQ3: What are the topics need more attention related to agile RE?**

After an extensive literature review of the related articles, the study attempt to bridge the gap to discover four main topics related to Agile RE in the field of software development.

**1. Limited empirical evaluation studies:**In a study conducted by Wohlin et al. [59] adopting the method of controlled empirical study. They also were of the view that classification such as questionnaires and case study could be considered as empirical analysis. Whereas, other approaches are mentioned in other articles such as a focus group, comparative articles, and simulations were considered as non-empirical studies. During the detailed examination, this study was able to find that among the articles which were retrieved 107 were non-empirically evaluated whereas 71 were empirically evaluated. Thus, it is been revealed that non-empirically validation of the articles was 60%.

**2. Insufficient studies for management of change:**Change management is one of the basic phases of Agile RE, and also closely linked with requirement management as well. In requirement management, only six (6) relevant articles were found. For instance, Soundararajan and Arthur [61], Anitha et al. [62], and Sillitti et al. [63] discussed the distinctions and similarities among traditional methods (V-model) and Agile methods to handle the volatility in gathering the requirements, in order to minimize the expensive accommodation cost towards the changes in the requirements. In a framework known as RE-KOMBINE was introduced by Erns et al. [28]. The objective of this framework was to analyze the factors, which could support the requirements process of the lightweight Agile. However, in Agile RE change management is considered as the challenge and still requires more empirical investigation [17] [64].

**3. Ignoring NFRs:**Even though there is a wide range of literature available on NFRs, is most of the related articles focused on non-agile context. However, there are studies such as a study by Fard and Mitropoulos [27], that tries to propose a method namely NORMAP, in order to help the practitioners to avoid ignoring NFRs. Additionally, a NORMATIC is a java based simulation instrument that also helps in the modeling of NFRs for the processes of partial-automated [27]. However, there is still a need for further researches on ignoring NFRs in Agile RE [17].

**4. Insufficient studies for requirements estimation:** The difference between estimation effort and actual effort is challenging to fulfill the requirements of the projects in Agile context [31]. Indeed, there are reasons that were discussed by articles such as requirement changes, missing requirements, and ignoring the NFRs [31][51]. In addition, a number of problems may be faced due to a lack of experience for practitioners in the estimation [64]. However, this topic needs more attention especially the estimation of effort is one of the major reasons for the success of the projects [65].

## 5. Discussion

This study conducted a SLR, to explore the challenges in Agile RE and proposed solutions pertaining to it, besides, this study aims to identify the CSFs and the topics need more attention in the literature to highlight for future studies. In related to RQ1, ten (10) challenges in Agile RE have identified after review the related articles. Furthermore, it is apparent that Agile RE presents several challenges as mentioned in table 5, eight (8) of these challenges were mentioned in SLR by Inayat [5], and the two (2) challenges that not mentioned before which are requirements prioritization and missing requirements, these challenges are considered as pivotal challenges and need to take into account from the practitioners during the development of software products [30][25]. On another side, there are issues that participate to produce these challenges, and some studies proposed solutions. For example, the NFRs are often an afterthought towards the end of the development period in Agile RE, and the failure of the system is often due to the ignorance of NFRs [17][32]. The user stories in most cases not enough to capture the NFRs. In addition, agile methodologies do not have a special method to elicit the NFRs [37]. Besides, the main problem as reported in the area of elicitation is the lack of guidelines for agile NFRs elicitation [66].

In addition, the accuracy of estimates based on the knowledge of the practitioners and with experiences to estimate the development effort [14][1]. While, new members of a team or junior members, besides the little documentation may lead to the imprecise estimations in the context of Agile [31]. Besides, this study explored six (6) CSFs of agile RE, for example, the grade of details for requirements has a significant impact on estimation, when the requirements have low levels of details in Agile context that may reduce the accuracy of the estimate and cost additional effort and time [13].

Indeed, there is a relation between the challenges and CSFs, the client participation is considered as CSFs, while client inability and agreement are considered as challenges in Agile RE, and that emphasize the good cooperation from the client have a significant impact in Agile RE. This study managed to discover four main topics related to Agile RE in response to RQ3, insufficient empirical assessment studies, limited studies for change management, ignoring NFRs, limited studies for requirements estimation, these topics need more attention from academicians in future researches.

## 6. Conclusion

This study is a systematic literature review on challenges, CSFs, and explored the topics that need more attention pertaining to Agile RE. The study of Kitchenham and Charters [15] have been used as a guideline to conduct the SLR. An electronic database was used to find articles that related to Agile RE. 178 articles related to Agile RE was found from 2002 until 2019. After a thorough analysis of articles, the study presented ten (10) challenges and six (6) CSFs. In addition, there is still a need for further studies in the future towards in some topics such as change management, ignoring NFRs, and requirements estimation, besides that, there is a need for more studies based on empirical evaluation in Agile RE. On another side, the limitations of any SLR are the potential inaccuracy in the extraction of data, and the prejudice in selecting studies. In order to remove these limitations, the study conducted the extraction of data by the keywords of research questions through the manual search and using auxiliary tools such as EPPI-Reviewer Web and SysRev. However, alternative keywords for requirements such as tasks, user stories, backlog, and cards, did not take into account during the research, which may lead to finding other studies also.

## References

1. 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.
2. J. F. Andry, H. Tannady, and F. E. Gunawan, Purchase Order Information System using Feature Driven Development Methodology, *International Journal of Advanced Trends in Computer Science and Engineering*, pp. 1–6, 2020.
3. K. Elghariani and N. Kama, Review on Agile requirements engineering challenges, *2016 3rd Int. Conf. Comput. Inf. Sci. ICCOINS 2016 - Proc.*, no. August 2016, pp. 507–512, 2016, doi: 10.1109/ICCOINS.2016.7783267.
4. Clancy, T. (2014). The Standish Group CHAOS Report. *Project Smart*. 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.

5. A. De Lucia and A. Qusef, Requirements engineering in agile software development, *J. Emerg. Technol. Web Intell.*, vol. 2, no. 3, pp. 212–220, 2010, doi: 10.4304/jetwi.2.3.212-220.
6. Port, A. Olkov, and T. Menzies, Using simulation to investigate requirements prioritization strategies, *ASE 2008 - 23rd IEEE/ACM Int. Conf. Autom. Softw. Eng. Proc.*, pp. 268–277, 2008, doi: 10.1109/ASE.2008.37.
7. D. M. Fernández *et al.*, Naming the pain in requirements engineering: Contemporary problems, causes, and effects in practice, *Empir. Softw. Eng.*, vol. 22, no. 5, pp. 2298–2338, 2017, doi: 10.1007/s10664-016-9451-7.
8. V. T. Heikkila, D. Damian, C. Lassenius, and M. Paasivaara, A Mapping Study on Requirements Engineering in Agile Software Development, *Proc. - 41st Euromicro Conf. Softw. Eng. Adv. Appl. SEAA 2015*, pp. 199–207, 2015, doi: 10.1109/SEAA.2015.70.
9. F. Soares, N. S. R. Alves, T. S. Mendes, M. Mendonca, and R. O. Spinola, Investigating the Link between User Stories and Documentation Debt on Software Projects, *Proc. - 12th Int. Conf. Inf. Technol. New Gener. ITNG 2015*, pp. 385–390, 2015, doi: 10.1109/ITNG.2015.68..
10. M. Islam, A Systematic Literature Review on the Critical Factors that Contribute to Success of Agile Development Projects, no. June, p. 40, 2016.
11. S. Alam, S. N. Bhatti, and S. Asim, Impact and Challenges of Requirement Engineering in Agile Methodologies : A Systematic Review, no. January, 2017, doi: 10.14569/IJACSA.2017.080455.
12. D. Liu, An empirical study of Agile planning critical success factors, no. June, pp. 1–66, 2017.
13. R. Telesko, Road to agile requirements engineering: Lessons learned from a web app project, *Stud. Syst. Decis. Control*, vol. 141, pp. 65–78, 2018, doi: 10.1007/978-3-319-74322-6\_5.
14. B.A. Kitchenham, Guidelines for performing Systematic Literature Reviews in Software Engineering, Version 2.3, Keele University and University of Durham, EBSE Technical Report, 2007.
15. S. Jalali and C. Wohlin, Systematic literature studies, p. 29, 2012, doi: 10.1145/2372251.2372257.
16. K. Curcio, T. Navarro, A. Malucelli, and S. Reinehr, The Journal of Systems and Software Requirements engineering : A systematic mapping study in agile software development, *J. Syst. Softw.*, vol. 139, pp. 32–50, 2018, doi: 10.1016/j.jss.2018.01.036..
17. K. Elghariani and N. Kama, Review on Agile requirements engineering challenges, *2016 3rd Int. Conf. Comput. Inf. Sci. ICCOINS 2016 - Proc.*, no. August 2016, pp. 507–512, 2016, doi: 10.1109/ICCOINS.2016.7783267..
18. Wolfgang, E. (2011). Working with user stories. In Agile requirements engineering workshop, July 2011.
19. A. 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, 2010, doi: 10.1111/j.1365-2575.2007.00259.x.
20. M. Pichler, H. Rumetshofer, and W. Wahler, Agile requirements engineering for a social insurance for occupational risks organization: A case study, *Proc. IEEE Int. Conf. Requir. Eng.*, pp. 246–251, 2006, doi: 10.1109/RE.2006.8.
21. Z. Racheva, M. Daneva, K. Sikkil, R. Wieringa, and A. Herrmann, Do we know enough about requirements prioritization in agile projects: Insights from a case study, BT - 2010 18th IEEE International Requirements Engineering Conference, RE2010, September 27, 2010 - October 1, 2010, "18th IEEE Int. Requir. Eng. Conf.", pp. 147–156, 2010, doi: 10.1109/RE.2010.27.
22. M. Daneva *et al.*, Agile requirements prioritization in large-scale outsourced system projects: An empirical study, *J. Syst. Softw.*, vol. 86, no. 5, pp. 1333–1353, 2013, doi: 10.1016/j.jss.2012.12.046.
23. L. Software, A. Systems, E. Conference, H. Regency, M. Bay, and S. Diego, Practical Agile Requirements Engineering, 2010.
24. R. Asghar, The Impact of Analytical Assessment of Requirements Prioritization Models : An Empirical Study, no. February, 2017, doi: 10.14569/IJACSA.2017.080240.
25. M. Yaseen, A. Mustapha, S. Qureshi, A. Khan, and A. U. Rahman, A Graph Based Approach to Prioritization of Software Functional Requirements, *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 3, p. 4285, 2020.
26. M. Farid, The Normap methodology: Lightweight engineering of non-functional requirements for agile processes, *Proc. - Asia-Pacific Softw. Eng. Conf. APSEC*, vol. 1, pp. 322–325, 2012, doi: 10.1109/APSEC.2012.23.
27. N. A. Ernst, A. Borgida, I. J. Jureta, and J. Mylopoulos, Agile requirements engineering via paraconsistent reasoning, *Inf. Syst.*, vol. 43, no. July, pp. 100–116, 2014, doi: 10.1016/j.is.2013.05.008.
28. H. M. Sarkan, T. P. S. Ahmad, and A. A. Bakar, Using JIRA and redmine in requirement development for Agile methodology, *2011 5th Malaysian Conf. Softw. Eng. MySEC 2011*, pp. 408–413, 2011, doi: 10.1109/MySEC.2011.6140707.

29. J. Vilela, E. Gonçalves, A. Holanda, B. Figueiredo, and J. Castro, Retrospective, relevance, and trends of SAC requirements engineering track, *Proc. ACM Symp. Appl. Comput.*, vol. 04-08-April-2016, pp. 1264–1269, 2016, doi: 10.1145/2851613.2851757.
30. M. Usman, E. Mendes, and J. Börstler, Effort estimation in Agile software development: A survey on the state of the practice, *ACM Int. Conf. Proceeding Ser.*, vol. 27-29-April-2015, 2015, doi: 10.1145/2745802.2745813.
31. J. Vilela, E. Gonçalves, A. Holanda, B. Figueiredo, and J. Castro, Retrospective, relevance, and trends of SAC requirements engineering track, *Proc. ACM Symp. Appl. Comput.*, vol. 04-08-April-2016, pp. 1264–1269, 2016, doi: 10.1145/2851613.2851757.
32. E. Hochmüller, The requirements engineer as a liaison officer in agile software development, *Proc. 1st Agil. Requir. Eng. Work. AREW'11 - Conjunction with ECOOP'11*, pp. 0–3, 2011, doi: 10.1145/2068783.2068785.
33. R. Goetz, How Agile Processes Can Help in Time-Constrained Requirements Engineering, *Int. Work. Time Constrained Requir. Eng.*, 2002, [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.19.280&rep=rep1&type=pdf>.
34. M. Käpyaho and M. Kauppinen, Agile requirements engineering with prototyping: A case study, *2015 IEEE 23rd Int. Requir. Eng. Conf. RE 2015 - Proc.*, pp. 334–343, 2015, doi: 10.1109/RE.2015.7320450.
35. R. Asghar, Role of Requirements Elicitation & Prioritization to Optimize Quality in Scrum Agile Development, vol. 7, no. 12, pp. 300–306, 2016.
36. R. R. Maiti and F. J. Mitropoulos, Capturing, eliciting, predicting and prioritizing (CEPP) non-functional requirements metadata during the early stages of agile software development, *Conf. Proc. - IEEE SOUTHEASTCON*, vol. 2015-June, no. June, 2015, doi: 10.1109/SECON.2015.7133007.
37. R. R. Maiti and F. J. Mitropoulos, Capturing, eliciting, predicting and prioritizing (CEPP) non-functional requirements metadata during the early stages of agile software development, *Conf. Proc. - IEEE SOUTHEASTCON*, vol. 2015-June, no. June, 2015, doi: 10.1109/SECON.2015.7133007.
38. M. Cardinal, Addressing Non-Functional Requirements with Agile Practices Who Am I?. Agile Specification, Addison-Wesley, Spring 2012.
39. W. Behutiye and P. Karhapää, Non-functional Requirements Documentation in Agile Software Development : Challenges and Solution Proposal, pp. 1–8.
40. N. A. Ernst, A. Borgida, I. J. Jureta, and J. Mylopoulos, Agile requirements engineering via paraconsistent reasoning, *Inf. Syst.*, vol. 43, no. July, pp. 100–116, 2014, doi: 10.1016/j.is.2013.05.008.
41. E. Bjarnason, M. Unterkalmsteiner, and M. Borg, PT US CR, *Inf. Softw. Technol.*, 2016, doi: 10.1016/j.infsof.2016.03.008.
42. J. Larsson and M. Borg, Revisiting the challenges in aligning RE and V&V: Experiences from the public sector, *2014 IEEE 1st Int. Work. Requir. Eng. Testing, RET 2014 - Proc.*, pp. 4–11, 2014, doi: 10.1109/RET.2014.6908671.
43. 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.
44. M. H. N. Nasir and S. Sahibuddin, Critical success factors for software projects: A comparative study, *Sci. Res. Essays*, vol. 6, no. 10, pp. 2174–2186, 2011, doi: 10.5897/sre10.1171.
45. E. Kouzari, V. C. Gerogiannis, I. Stamelos, and G. Kakarontzas, Critical success factors and barriers for lightweight software process improvement in agile development: A literature review, *ICSOFT-EA 2015 - 10th Int. Conf. Softw. Eng. Appl. Proceedings; Part 10th Int. Jt. Conf. Softw. Technol. ICSOFT 2015*, pp. 151–159, 2015, doi: 10.5220/0005555401510159.
46. 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.
47. T. Chow and D. B. Cao, A survey study of critical success factors in agile software projects, *J. Syst. Softw.*, vol. 81, no. 6, pp. 961–971, 2008, doi: 10.1016/j.jss.2007.08.020.
48. S. C. Misra, V. Kumar, and U. Kumar, Identifying some important success factors in adopting agile software development practices, *J. Syst. Softw.*, vol. 82, no. 11, pp. 1869–1890, 2009, doi: 10.1016/j.jss.2009.05.052.
49. A. Port, A. Olkov, and T. Menzies, Using simulation to investigate requirements prioritization strategies, *ASE 2008 - 23rd IEEE/ACM Int. Conf. Autom. Softw. Eng. Proc.*, pp. 268–277, 2008, doi: 10.1109/ASE.2008.37.
50. A. Stankovic, V. Nikolic, M. Djordjevic, and D. B. Cao, A survey study of critical success factors in agile software projects in former Yugoslavia IT companies, *J. Syst. Softw.*, vol. 86, no. 6, pp. 1663–1678, 2013, doi: 10.1016/j.jss.2013.02.027.

51. J. Sheffield and J. Lemétayer, Factors associated with the software development agility of successful projects, *Int. J. Proj. Manag.*, vol. 31, no. 3, pp. 459–472, 2013, doi: 10.1016/j.ijproman.2012.09.011.
52. A. Shahane, P. Jamsandekar, and D. Shahane, Factors influencing the agile methods in practice - Literature survey & review, *2014 Int. Conf. Comput. Sustain. Glob. Dev. INDIACom 2014*, no. March 2014, pp. 556–560, 2014, doi: 10.1109/IndiaCom.2014.6828020.
53. M. H. N. Nasir and S. Sahibuddin, Critical success factors for software projects: A comparative study, *Sci. Res. Essays*, vol. 6, no. 10, pp. 2174–2186, 2011, doi: 10.5897/sre10.1171.
54. J. A. Livermore, Factors that significantly impact the implementation of an agile software development methodology, *J. Softw.*, vol. 3, no. 4, pp. 31–36, 2008, doi: 10.4304/jsw.3.4.31-36.
55. K. Dikert, M. Paasivaara, and C. Lassenius, Challenges and success factors for large-scale agile transformations: A systematic literature review, *J. Syst. Softw.*, vol. 119, pp. 87–108, 2016, doi: 10.1016/j.jss.2016.06.013.
56. M. H. N. Nasir and S. Sahibuddin, Critical success factors for software projects: A comparative study, *Sci. Res. Essays*, vol. 6, no. 10, pp. 2174–2186, 2011, doi: 10.5897/sre10.1171.
57. A. Destefanis, Requirements Effort Estimation : State of The Practice Requirements Effort Estimation : State of The Practice, no. October 2015, 2016.
58. R. Conradi and A. I. Wang, *Empirical Methods and Studies in Software Engineering*. 2003.
59. W. Fatimah and W. Ahmad, Handling Requirements Using FlexREQ Model, pp. 661–664..
60. S. Soundararajan and J. D. Arthur, A soft-structured agile framework for larger scale systems development, *Proc. Int. Symp. Work. Eng. Comput. Based Syst.*, pp. 187–195, 2009, doi: 10.1109/ECBS.2009.21.
61. P. C. Anitha and D. Savio, Managing Requirements Volatility while ‘ Scrumming ’ within the V-Model, pp. 17–23, 2013.
62. Sillitti, M. Ceschi, B. Russo, and G. Succi, Managing uncertainty in requirements: A survey in documentation-driven and Agile companies, *Proc. - Int. Softw. Metrics Symp.*, vol. 2005, pp. 145–154, 2005, doi: 10.1109/METRICS.2005.29.
63. S. Dragicevic, S. Celar, and M. Turic, Bayesian network model for task effort estimation in agile software development, *J. Syst. Softw.*, vol. 127, pp. 109–119, 2017, doi: 10.1016/j.jss.2017.01.027.
64. M. Ochodek, Approximation of COSMIC functional size of scenario-based requirements in agile based on syntactic linguistic features - A replication study, *Proc. - 26th Int. Work. Softw. Meas. IWSM 2016 11th Int. Conf. Softw. Process Prod. Meas. Mensura 2016*, pp. 201–211, 2017, doi: 10.1109/IWSM-Mensura.2016.039.
65. M. Younas, D. N. A. Jawawi, I. Ghani, and R. Kazmi, Non-Functional Requirements Elicitation Guideline for Agile Methods, *J. Telecommun. Electron. Comput. Eng.*, vol. 9, no. 3-4 Special Issue, pp. 137–142, 2017.