Received: 30 June 2023 | Revised: 20 December 2023 | Accepted: 24 May 2024

DOI: 10.1002/smr.2698

REVIEW ARTICLE

Journal of Software: Evolution and Process WILEY

Software maintenance practices using agile methods towards cloud environment: A systematic mapping

Mohammed Almashhadani [1,2] | Alok Mishra [3,4] | Ali Yazici [3]

[1]Graduate School of Natural and Applied Sciences, Atilim University, Ankara, Türkiye

[2]Council of Representatives of Iraq, Baghdad International Zone-Convention Center, Baghdad, Iraq

[3]Department of Software Engineering, Atilim University, Ankara, Türkiye

[4]Faculty of Engineering, Norwegian University of Science and Technology (NTNU), Norway

Correspondence
Alok Mishra, Faculty of Engineering, Norwegian University of Science and Technology (NTNU), Norway.
Email: alok.mishra@ntnu.no

Abstract

Agile methods have emerged to overcome the obstacles of structured methodologies, such as the waterfall, prototype, spiral, and so on. There are studies showing the usefulness of agile approaches in software development. However, studies on Agile maintenance are very limited in number. Regardless of the chosen methodology, software maintenance can be carried out in either a local (on-the-premise) or global (distributed) environment. In a local environment, the software maintenance team is co-located on the same premises, while in a global environment, the team is geographically dispersed from the customer. The main objective of this Systematic Mapping (SM) study is to identify the practices useful for software maintenance using the Agile approaches in the Cloud environment. We have conducted a comprehensive search in well-known digital databases and examined the articles that map to the pre-defined inclusion criteria. The study selected and analyzed 48 articles out of 320 published between 2000 and 2022. The findings of the mapping study reveal that Agile can resolve the major issues faced in traditional software maintenance, making the role of this approach significant in global/distributed software maintenance. Cloud computing plays a vital role in software maintenance. Most of the studies highlight the application of XP- and Scrum-based Agile maintenance models. The study found a need for more Agile maintenance solutions in the cloud, highlighting the importance of agile in software maintenance, both locally and globally. Irrespective of the environment, Cloud computing provides a centralized platform for collaboration and communication, while also offering scalability and flexibility to adapt to diverse infrastructure needs. This allows agile maintenance practices to be implemented across both local and global environments, leveraging the cloud's capabilities to overcome geographical and infrastructural challenges.

KEYWORDS

agile methods, cloud computing, global environment, local environment, software maintenance

1 | INTRODUCTION

The Agile methods have been in use since about the year 2000 with the main objective to help software practitioners to achieve their objectives by focusing on customer satisfaction as the highest priority, delivering the product on time, and accommodating changes at any time. The Agile manifesto, led by Kent Beck and his followers,[1] stands as a groundbreaking initiative in the Agile approach to software development. It supports the power of individuals and collaboration over the constraints of rigid processes and tools, emphasizing the delivery of functional software over extensive documentation, promoting customer collaboration over contract negotiations, and embracing the need for adaptability over adherence to a plan.

The methods applied in the Agile approach[2] have several characteristics that make the development process simply more Agile as the term implies. These characteristics are cooperativeness (i.e., focusing on people satisfaction rather than process), adaptiveness (response to rapid change in the business environment), iteration (it means small and frequent deliveries to the customers), and light documentation. Additionally, each development cycle requires only two to four weeks, the design is simple, can be carried out by a small team, and there is direct and face-to-face communication, collective ownership is possible related to the code and, lastly, the code quality itself is improved.

Abrahamsson et al.[S36] highlight different kinds of Agile approaches such as Extreme Programming (XP), Scrum, Crystal family of methods, Feature Driven Development, Dynamic System Development Method, and Adaptive Software Development. Each one of these approaches offers different benefits for developers.

The Agile methods have 12 principles that possess certain practices to simplify the software development processes. However, this method has faced some challenges, namely the need for face-to-face communication,[3] availability of maintainer experts,[4] and the ability to develop and maintain the software in the distributed environment.[5] One of the proposed solutions to overcome these problems is the use of Cloud Computing, which facilitates software development by reducing the total cost of development by utilizing features such as sharing data, distributed system applications, and re-prioritization of tasks.[5] In addition, the Cloud computing environment eliminates the need for installation and configuration processes and installation of software patches.[4,6] Another advantage of the Cloud is the utilization of pay-per-use approach.[7,8] Cloud computing improves software development using the Agile methods by providing increased productivity, reducing development costs, and improving software quality.[9,S2]

The application of the agile methods in the Cloud Computing environment and some of the benefits realized according to previous studies are as follows:

Cloud Computing

- Improves the quality factors
- Facilitates cooperation between team members
- Increases transparency among the teams
- Provides the infrastructure for the development process
- Facilitates traceability in the distributed environment

All these benefits motivated us to study the possibility of using the software maintenance process along with the Cloud Computing environment.[10,S35,S43,S44,S45] As it is well-known, the maintenance process is one of the longest stages of the software life cycle.[S16] Therefore, the main motivation for conducting this study is to cover other aspects of this approach by constructing several research questions that were not considered by earlier studies. Thus, this study attempts to explore the benefits and challenges of Agile methods for both local (on-premises) and global environments (distributed), to investigate the possibility of applying it in the Cloud to simplify the software maintenance process.

Briefly, in a local environment, the maintenance team is co-located in a single office and collaborates in real-time on shared databases, whereas in the global environment, the team is geographically dispersed, and collaboration is usually asynchronous using some collaboration tools. In local computing environments, agile maintenance can be implemented relatively easily. Teams can co-locate and work together closely, which facilitates communication and collaboration. Additionally, the software infrastructure is likely to be standardized and well-understood, making it easier to adapt to agile practices. However, there may be challenges in managing teams that are geographically dispersed or that have different levels of experience with agile methods. Additionally, ensuring that the software infrastructure can support agile development and maintenance may require some upfront investment. In global computing environments, the challenges of implementing agile maintenance are more significant. Teams may be located in different time zones and cultures, which can make it difficult to communicate and collaborate effectively. Also, the software infrastructure may be more complex and diverse, making it more difficult to adapt to agile practices. Cloud computing bridges these gaps by providing a centralized platform for collaboration and communication regardless of location, while also offering scalability and flexibility to adapt to diverse infrastructure needs. This allows agile software maintenance practices to be implemented across both local and global environments, leveraging the cloud's capabilities to overcome geographical and infrastructural challenges.

To investigate the resources provided by the Cloud platform in simplifying the maintenance process, the present paper analyzes the existing Agile maintenance studies by conducting an SM study to identify the trends and volume of previous attempts in this domain. Furthermore, it is

expected that this work will shed further light on different Agile maintenance practices that can be used by software practitioners in Cloud environments.

Seven research questions are framed related to Agile software maintenance in Section III of the article. To answer these questions, we searched different databases and selected (after quality assessment) 48 articles out of 320 that were published between 2000 and 2022. In this study, we have identified the articles related to Agile maintenance, specified the benefits of using Agile for software maintenance, defined different Agile (XP/Scrum) practices, identified the advantages of using those practices for maintenance, defined the challenges of using Agile for software maintenance for the local and global environment, and identified the benefits of using Cloud for the software development.

The rest of this paper is organized as follows: Section 2 introduces the background information and describes the SM method. In Section 3, the related research questions are posed. Section 4 offers the discussion and the validity of the study. Finally, in Section 5 conclusions and future work are given.

2 | RELATED WORK

2.1 | Software maintenance

The term 'maintenance' was introduced in the 1960s to indicate any modification made in a system. The Institute of Electrical and Electronic Engineers (IEEE) defines software maintenance as a *"Modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment."*[11] Software maintenance refers to continuous management to fix problems and adapt functionality/features for the system, while software development refers to the activities of creating a project from scratch. Therefore, the activities carried out in these two practices are completely different.[S13] Software maintenance has four phases, namely, the introduction phase (identify the problem), the growth phase (reveal technical problems), the maturity phase (initiate major improvements), and the decline phase (renew the system if it is not serviceable).[S13]

The maintenance phase begins after the delivery process. During maintenance, the software is modified many times, and many versions of the software are released.[S18] A classification for software maintenance which helps practitioners to understand the related activities is outlined below:[12,13,S39]

- **Adaptive maintenance**: Adapts software to the new environment.
- **Perfective maintenance**: Adds new requirements and features to the software.
- **Corrective maintenance**: is related to fixing bugs and emerging errors.
- **Preventive maintenance**: is related to preventing errors and bugs that would possibly appear in the future.
- **Emergency maintenance**: is related to fixing systems or software in the case of failure.

According to,[S16] maintenance is the longest phase in the software development life cycle (SDLC), and it consumes about 40-70% of the time and cost of the total project.

2.2 | Agile maintenance

The Agile methods accommodate very popular software development processes, and due to its characteristics, such as speed, scalability, reliability, and flexibility, is also used for maintenance processes.[14,S37] The Agile approach blurs the distinction between development and maintenance due to its emphasis on customer-centric requirements and continuous improvement throughout the software lifecycle.[15,16] Agile's emphasis on fostering close collaboration between maintainers and customers aligns effectively with the maintenance process. This heightened interaction is another key reason for adopting Agile approaches for maintenance tasks. As Figure 1 shows, Agile maintenance lies at the intersection of Agile methods and classical software maintenance.[S39] In general, Agile methods have a group of common practices, some of which work for both development and maintenance activities.[S2,S6] These practices were extracted from Agile methods such as XP, Scrum, and other related families to support Agile principles.

This paper delves into the existing literature pertaining to Agile maintenance practices, approaches, and techniques. Several digital scientific databases were explored to identify relevant studies pertaining to the aforementioned subject. In the context of Agile maintenance SM, only a single work was identified, in which 31 studies about Agile maintenance published between 2001 to 2015 were investigated.[S14] In article,[S14] emphasis was placed on showing different Agile methods, related tools, and the strengths and weaknesses of the use of Agile methods. In the next section, some Agile maintenance practices that can be used in the Cloud environment will be reviewed.

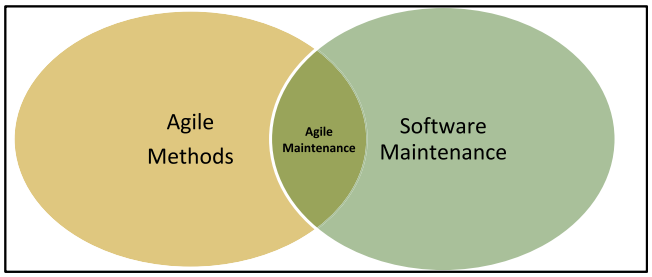

FIGURE 1 Combining agile and software maintenance (adapted from[S39]).

2.3 | Agile and cloud

There are many studies about Global Software Development (GSD), some of which show the challenges facing development in the global environment and introduce possible solutions.[17] However, there are many common challenges in a globally distributed context such as lack of communication, lack of management, low team morale, and unclear or uncertain requirements, which can be solved with the help of Cloud platforms.[10] Other studies have suggested models or frameworks for adapting Agile to the Cloud computing environment.

The authors in[10] introduced the Dynamic Systems Development model by using Google App-Engine as a Cloud platform. In addition, the Skype app is used to support collaboration among development team members. The model was evaluated by developing a warehouse management application using both environments (i.e., on-premises and on-Cloud), and the two environments were compared in terms of the development time. The results of the study indicate that the development time using Cloud Computing is faster due to the application of Cloud resources in the development process. The proposed model consisted of three phases: (1) pre-project phase, which is responsible for identifying and arranging the priorities, feasibility studies, and defining the project goal; (2) life cycle project phase, where functional model iteration is defined, the design is conducted, and the implementation of the project is done; and (3) post-project phase, which measures the functioning efficiency and effectiveness and is interested in error detection and correction.[10]

In another study,[18] the interconnectedness between Agile and Cloud platforms is studied by conducting a survey in a development organization. The survey questions mainly focus on collaboration between development team members and the Cloud services provider. The company uses Scrum as the Agile method, and Skype as a tool for collaboration among development team members. The results of this study showed that the use of Cloud Computing greatly contributed to facilitating cooperation among team members, which was positively reflected in the increased agility.

In,[19] a crowd-sourcing model is proposed, which can also be used for Agile-Cloud development. In the study, Confluence is used as a tool for communication between team members, and GitHub for code repositories. For Agile Cloud integration, Chef and Puppet Lab, are utilized.

In,[S8] the authors introduced a framework for Agile Development in Cloud Computing Environment (ADCC). In,[S43] the same authors evaluated the framework by introducing a structural environment for software development using a distributed Agile over Cloud Computing platform. The ADCC framework has four steps: (1) Agile features selection, where the desired Agile method from the Agile family is selected; (2) Cloud platform selection according to organization size (small, middle, big) and, depending on business requirements, the organization's privacy (if there is a need to secure information); and (3) code management and repository, where tools such as GitHub, FishEye, Bamboo, and BitBucket are utilized to facilitate common development tasks. The ADCC framework is a novel approach for adapting Agile development software on the Cloud. However, in the same study, in the case of Agile maintenance, some shortcomings are reported.

3 | RESEARCH METHODOLOGY

3.1 | SM process

To achieve our goal, an SM study is conducted to answer the research questions posed in Section III.C. Brereton et al.[20,21] introduced a guideline for SM in software engineering. And for the execution of the SM, he suggested three steps, namely, planning, conducting, and documenting as shown in Figure 2.

3.2 | Research scope

We have searched in different databases and identified 48 studies out of 322 studies related to the scope, which covers the following inquiries:

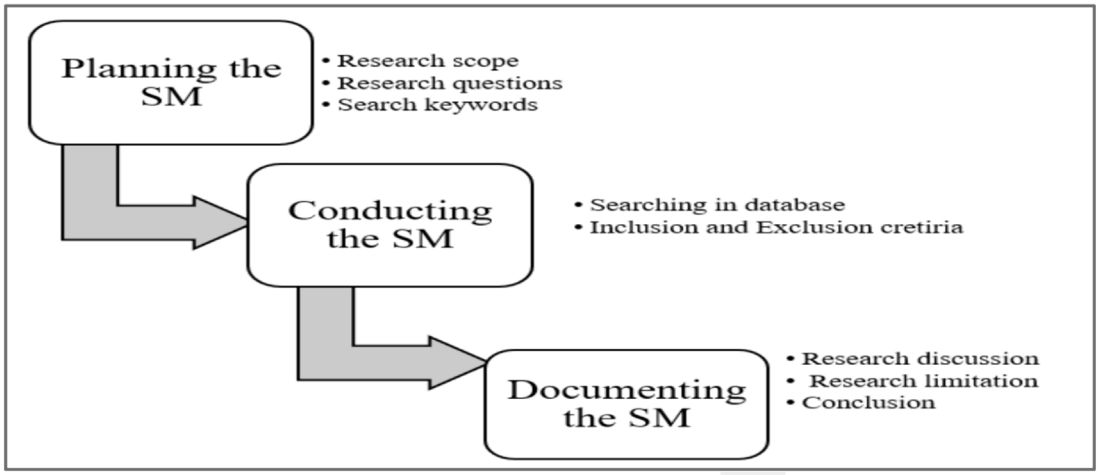

FIGURE 2 SM PROCESS.[3]

TABLE 1 Research questions.

#	Research question	Motivation
RQ1	What are the existing studies in Agile software maintenance?	Investigate the existing studies about Agile maintenance.
RQ2	What are the application and benefits of using an Agile approach for maintenance?	Identify the benefits of applying the Agile method for maintenance
RQ3	What are the Agile practices (XP/Scrum) for local and global environments?	Identify which Agile practices can be used for the local and global environment.
RQ4	What are the Agile practices (XP/Scrum) that are used for maintenance?	Identify what Agile practices can be used for maintenance.
RQ5	What are the advantages of applying Agile practices for maintenance?	Identify the benefits of using Agile practices for maintenance.
RQ6	What are the challenges in using Agile maintenance for a local and global environment?	Identify existing challenges faced by the maintenance process using the Agile method for both environments.
RQ7	What are the advantages of applying Agile methods in a Cloud Computing environment?	Identify all the benefits of using SDLC in the Cloud.

1. What are the existing studies for Agile maintenance?
2. What are the different Agile practices for maintenance?
3. What are the benefits of using Agile for maintenance in local and global environments?
4. What are the challenges of using Agile for maintenance in local and global environments?

This SM study focuses on Agile maintenance models and practices based on XP and Scrum, and it seeks the possibility of adapting the Agile maintenance model for the Cloud environment. The reason behind choosing these two approaches is the popularity of these two methods and the focus of related literature on them. It is clear that these two are the most widely used than the rest of the agile methods. Additionally, Scrum has many practices affecting project management. Similarly, XP has many practices that are suitable for project engineering such as; continuous integration, pair programming, and refactoring. In other words, these two methods complement each other in many aspects of software development.[S6] Moreover, project management and project re-engineering are the main factors in software maintenance. Additionally, there are many studies that proved the effectiveness of using XP and Scrum practices for maintenance.[S2,S4,S6,S12,S15,S19,S46,S47,S48]

3.3 | Research questions

In this SM study, the main goal is to find answers and explanations to some detailed questions related to our scope. Table 1 includes seven research questions along with the purpose of each. All research questions are focused on software maintenance and Agile methods.

3.4 | Search keywords strategy

For the searching process, a group of keywords that are related to the current SM are identified and arranged as search strings and are illustrated in Table 2.

3.5 | Search databases

Following the guidelines given in,[20,21] the databases listed in Figure 3 are searched. In the same figure, the distribution of the 322 articles is given among, IEEE Xplore, Scopus, Springer, Web of Science, Science direct, and other databases such as (ACM, Google Scholar, CiteSeer [X], ACSI, etc.).

3.6 | Inclusion and exclusion criteria

In the selection process, studies that do not conform to the objectives of the study and the RQs are excluded. The inclusion and exclusion criteria for our research are listed below:

Inclusion criteria:

- Publications between 2000 and 2022
- Publications proposing Agile maintenance methods, such as XP or Scrum
- Case studies for Agile maintenance models

TABLE 2 Search strings.

Search strings
1. Agile and software maintenance model
2. Agile and maintenance framework
3. Agile and maintenance tools
4. Agile and XP maintenance
5. Agile and Scrum maintenance
6. Agile development and Global environment
7. Agile development and Cloud

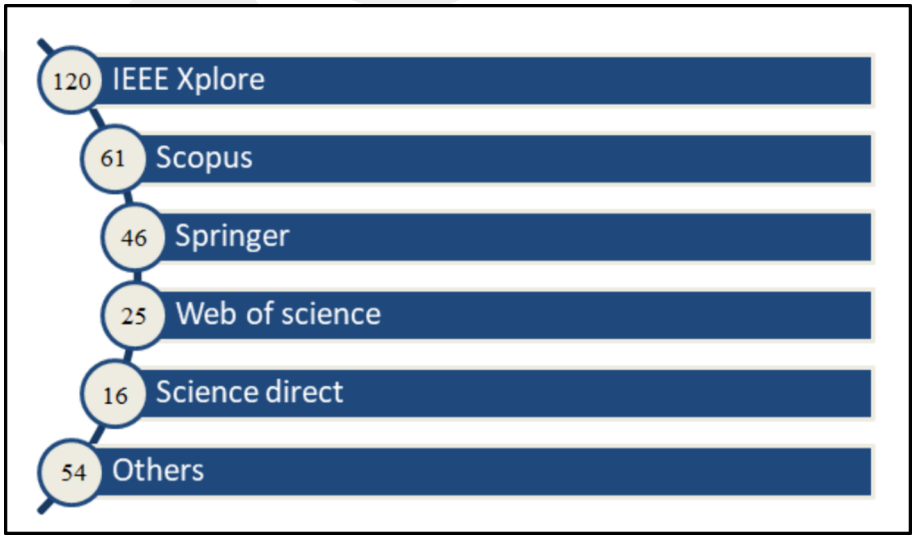

FIGURE 3 Distribution of studies OVER databases.

- Publications addressing Agile practices for maintenance
- Publications addressing Agile practices for the global environment.
- Publications addressing Agile practices for Cloud environment.

Exclusion criteria:

- Publications written in languages other than English
- Workshops and web links
- Publications not focusing on Agile maintenance models or Agile practices

In addition, for the studies that appear in more than one database (overlapping studies), only one is included in the outcome set. The selection process is illustrated in Figure 4. In the first phase, articles according to the title related to research the scope, and the results are filtered (322 papers). In the next phase, the papers are filtered, after reading the abstract and conclusion (102 papers). After that, in the third phase, 74 papers are read in detail. In the final phase, the remaining 48 articles are investigated in detail and analyzed to extract results and findings.

3.7 | Quality assessment

We test the quality assessment for the selected studies by using the quality criteria assessment questions stated in Ref.[3] The questions are:

Q1: Are the aims of the study clearly stated?
Q2: Are the scope and experimental design of the study defined clearly?
Q3: Are the variables in the study likely to be valid and reliable?
Q4: Is the research process documented adequately?
Q5: Are all the study questions answered?
Q6: Are the negative findings presented?
Q7: Are the main findings stated clearly in terms of creditability, validity, and reliability?
Q8: Do the conclusions relate to the aim of the purpose of the study?
Answer the assessment question according to the following score:

- Yes = 2
- No = 1, and
- Somewhat = 0.

The minimum score was 0, the maximum score was 8, and the paper with a score of 4 was excluded.

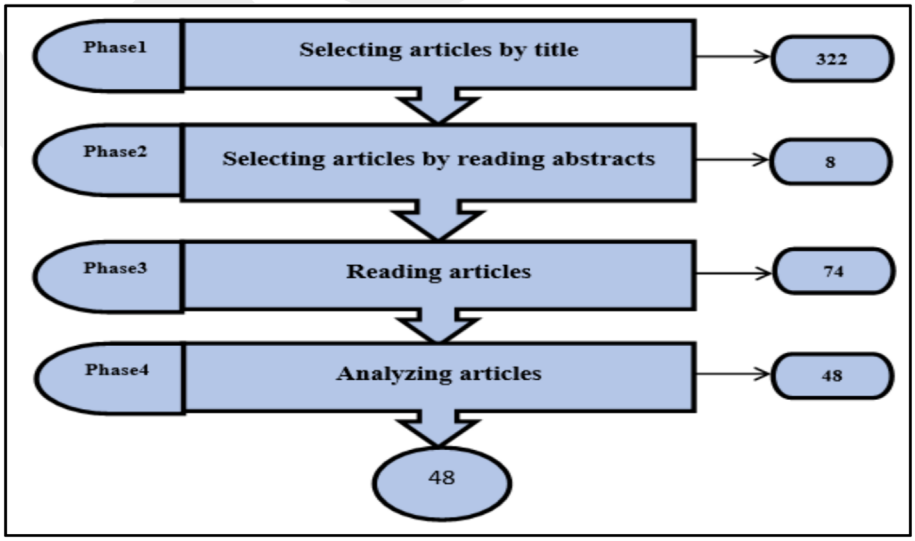

FIGURE 4 Phases of selection process.

4 | ANSWER TO RESEARCH QUESTIONS

RQ1. What are the existing studies in Agile software maintenance?

In this section, the studies that propose Agile maintenance models or studies related to Agile maintenance practices are reviewed. The main goal of this step is to identify articles in this field and help researchers study other aspects that have not been addressed so far. In many studies software maintenance for Agile methods is discussed. A list of these studies is provided in Table 3.

The evaluation of Agile maintenance studies per year, illustrated in Figure 5, shows that the highest ratio of publishing was[4] articles in 2015, then 2 articles for 2009, 2011, 2014, and 2018.

Regarding publishing, 57% out of 21 studies are journal articles, while 33% are conference articles, along with 10% others (IBM report, thesis). In Figure 6 this distribution is shown.

RQ2. What are the applications and benefits of using an Agile approach for maintenance?

There are many advantages acquired by applying Agile methods for the software maintenance process. Table 4 summarizes the main advantages the literature studies have referred to.

RQ3. What are Agile practices (XP/Scrum) for local and global environments?

The Agile approach is applied with different methods such as XP, Scrum, Crystal, Lean, Feature-Driven Development (FFD), etc.[S36] The most popular methods of Agile methodologies are XP and Scrum. While Scrum focuses on software development including project management, XP is more oriented towards engineering practices.[S6] Cloud Computing can use local and distributed environments, and there is a need to know the practices in both. Table 5 shows the different practices for XP and Scrum methods. Some of these practices can be used for both the local and global environments, while others need to adapt themselves to the global environment.

RQ4. What are Agile practices (XP/Scrum) that are used for maintenance?

In the previous question, we inquired about XP and Scrum practices. In this question, we aim to investigate Agile practices for maintenance. Agile practices that are used in the maintenance process are reviewed according to the related literature. Table 6 provides the Agile practices that have been adopted for maintenance.

RQ5. What are the advantages of applying Agile practices for maintenance?

Table 7 shows the impact of Agile practices that lead to achieving the benefits of using Agile for maintenance according to the literature studies.

RQ6. What are the challenges in using Agile maintenance for a local and global environment?

In this section, the challenges faced during the maintenance process for local environments are reviewed and possible solutions stated in the literature are identified. (see Table 8). Additionally, possible solutions provided by previous studies are listed. Based on the literature, three main challenges are reported for the global environment as mentioned in[S21,S22,S27,S29,S33]; these issues are further reviewed and possible solutions are identified accordingly (see Table 9).

RQ7. What are the advantages of applying Agile methods in a Cloud Computing environment?

In this section, the benefits of applying Agile development in a Cloud Computing environment will be reviewed. According to the literature, development and maintenance differ from each other in detail, but they have the same phases in general.[11,24–28] Therefore, knowing the benefits of using development in the Cloud will motivate and inspire us to apply the Cloud platform with software maintenance. There are several benefits of using Agile methods in the Cloud Computing environment. These are:

Quality factors: Cloud Computing helps to facilitate Agile development and maintain quality throughout the development process. The quality factors are divided into sub-factors such as scalability, maintainability,[S35] which affect and accelerate Agile development in a Cloud Computing environment.

TABLE 3 List of studies.

Study ID	Studies description	Study type
S1	A case study of an offshore Agile maintenance project that introduces practices that would address challenges facing offshore software maintenance, and present some observations about the project.	CS
S2	A case study for adapting Extreme programming XP practices at Lona Technologies. The maintenance method was demonstrated along with the case study from the company, and comments were made on the maintenance process.	CS
S4	A case study was conducted at a large software development company to verify the results of introducing XP for maintenance. The result proved that the XP practices need re-designing and adaptation to fulfill the requirements and needs.	CS
S5	A case study was conducted at an industrial company to identify the challenges faced in developing and maintaining software using the Agile methods. A literature review is provided in which tools, methods, and knowledge which are available for refactoring of code smells are investigated.	CS, LR.
S6	Researchers proposed a model for evolution and maintaining software based on the Agile methods. XP and Scrum approaches are compared and maintenance practices are investigated. Also, the proposed model was evaluated using two Canadian companies.	MOD, CS.
S9	A framework is based on Agile techniques and principles is proposed, and then applied in[13] IT services.	FRM, CS
S12	A maintenance model is proposed applying the Scrum method which deals with urgent requests during the current sprint. The authors evaluated the study by maintaining a fitness system. The maintenance team was in South Asia and customers were in the USA.	MOD, CS
S13	The authors adapted practices of Agile development for maintenance activities and produced nine heuristics. This study was conducted in collaboration with the IT department at Aalborg University and the maintenance team at Aveva organization.	CS
S14	The researchers introduced a systematic review about Agile maintenance covering[30] research articles to help practitioners and maintainers. This study answered 10 questions, leading to adopting an Agile method for maintenance.	SR
S15	This study highlights the impact of different Agile methods on maintainability as seen by fans and critics; it also shows the benefits of using Agile (XP, Scrum) approach for the maintenance process.	OTH
S16	The main purpose of this study is to enhance software maintenance governance for the Agile maintenance team and to propose a new tool called Axita developed to support the team by managing time projects effectively and arranging data in a central data warehouse. Furthermore, in this study, the authors proposed six practices for managing the maintenance process to overcome the challenges that would occur during searching information and reduce the processing time.	TOL
S17	This study was conducted on Open Stack platform to evaluate software maintenance for PF Cloud systems that have been developed using the Agile development method.	CS
S18	This is an exploratory study that investigates whether the Agile methods can help software maintainers. In addition, it introduces a matrix framework based on practitioner views, thus aiming to tackle issues that impact Agile adoption decisions.	CS, FRM
S19	The Agile process for maintenance life cycles based on the Scrum method was introduced.	OTH
S28	A list of metrics that can define a degree of maintainability, and to examining whether these metrics can apply to the Agile methods is reported.	OTH
S37	A literature review about Agile maintenance was conducted. Related works through investigating the challenges and obstacles that face Agile maintenance are analyzed.	LR
S39	22 Agile practices that can be adopted in software maintenance by conducting interviews and defining the advantages and disadvantages of these practices for the maintenance process are identified.	CS, LR
S40	An agile software maintenance method called Agile MANTEMA, focusing on small organizations, is proposed. As a case study two companies were chosen and the effectiveness of the method was measured.	MTH, CS
S46	An iterative maintenance life cycle based on XP was proposed to solve maintenance challenges such as cost, time, and effort, and increase the software maintainability.	MOD
S47	An iterative maintenance model based on XP was proposed to enhance maintenance. The model was evaluated in an academic environment by applying it in many projects. The observations showed that the use of this model can produce maintainable codes with good quality and that this model speeds up the maintenance process with less effort.	MOD, CS
S48	It is an empirical study that tests the maintainability of the system for an academic project. Each project was assigned to two groups of postgraduate students. One of them used the XP model and the other used the Waterfall model. The observation stated that the group that used the XP method produced a more maintainable code than the group that used the Waterfall model.	CS

Legend: - CS: Case Study, MOD: Models, TOL: Tool, MTH: Methods, FRM: Framework, OTH: Other.

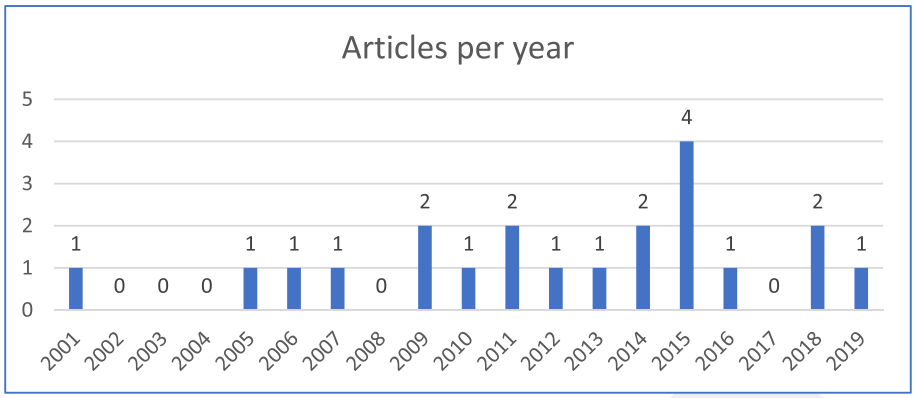

FIGURE 5 Agile maintenance articles per year.

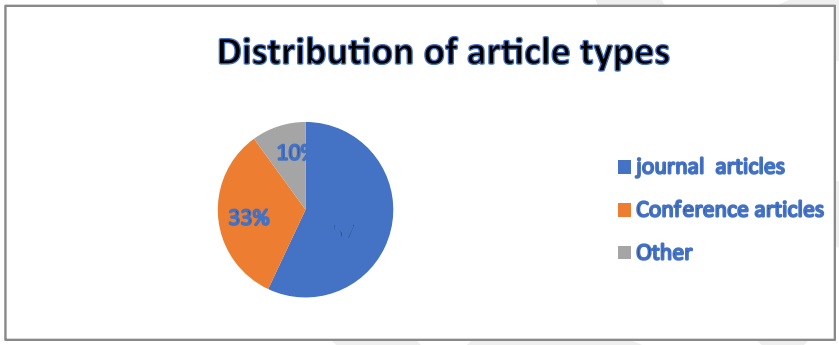

FIGURE 6 Distribution of article types.

TABLE 4 Advantages of using agile for maintenance.

Agile maintenance benefits	Studies
1. Speed up in the maintenance process.	S13,S28,S46
2. Improved communication.	S13,S20,S46
3. Increase in customer satisfaction.	S9,S12,S28
4. Increase in product quality.	S13,S15,S20,S40,S46
5. Increase in the morale among the team members.	S13
6. Increase in software maintainer's satisfaction due to reducing rework.	S12,S46
7. Simplify knowledge transfer among the teams.	S2,S4,S13
8. Improve productivity.	S4,S15,S39,S46
9. Improve project visibility.	S2,S13,S46
10. Fix bugs and merge them with code rapidly.	S15,S39
11. Improve prioritization of tasks.	S39
12. Enhance test suites.	S39
13. Increase the index of maintainability.	S39
14. Improve program comprehension.	S39,S46
15. Improve the accuracy of cost and effort forecasts.	S39
16. Improve the design at the abstraction level.	S39

The first factor is scalability, which means the ability to deal with a large team, through managing backlog, sprint, and feedback between customers and developers. Cloud Computing provides the infrastructure and resources which are required to manage a large team. Another perspective of scalability in Cloud Computing is automation, which is the ability to provide resources according to the customers' needs using scale-up or scale-down.[S8,S34,S35,S43,S45]

TABLE 5 Scrum and XP practices.

XP	Local(L) [S2,S4,S6,S36] global(G) [S21,S25,S27]	SCRUM	Local[S3,S6,S36,S42] global[S21,S24,S29]
1. Planning game	L	1. Sprint planning meeting	L/G
2. Pair programming	L	2. Distributed sprint planning meetings	G
3. Continuous Integration	L	3. Sprint Backlog	L/G
4. On-Site Customer	L	4. Product backlog	L/G
5. Testing	L/G	5. Sprint Review meeting	L
6. Refactoring	L/G	6. Daily scrum meeting	L/G
7. Small releases	L/G	7. Retrospective meeting	L/G
8. Collective ownership	L/G	8. Separate backlogs for each team	G
9. Metaphor	L/G	9. Scrum of scrum	L
10. 40-hour week	L/G	10. Distributed Scrum of Scrum	G
11. Coding standards	L/G	11. Sprint demo	L/G
12. Simple design	L/G	12. Two-week maintenance sprints	G

TABLE 6 Maintenance practices.

Agile practices	Agile method	Sources
1. Planning game	XP	S1,S2,S4,S47
2. Small releases	XP	S1,S2,S4,S47
3. Refactoring	XP	S1,S2,S4,S47
4.Pair Programming	XP	S1,S2,S4,S32
5.Collective code ownership	XP	S1,S2,S4,S6,S32
6. Continuous integration/Automated Release	XP	S1,S2,S4,S47
7. Test-Driven Development	XP	S1,S47
8. Iteration Planning	XP	S1
9. 40-hour week	XP	S1,S2,S4,S32
10. Standup meetings	XP-Scrum	S1,S4,S47
11. Coding standards	XP	S1,S2,S4
12. Onsite Client	XP	S2,S4,S32
13. Metaphor	XP	S2,S4
14. Simple Design	XP	S2,S4
15. Automated testing	XP	S1,S4
16. User Stories	XP-Scrum	S1,S6
17. Task Prioritization	XP-Scrum	S32,S37
18. Task Board	XP-Scrum	S1,S2,S4
19. Product Backlog	Scrum	S6,S37,S39
20. Retrospective	Scrum	S4,S39
21. Scrum of Scrum	Scrum	S6,S39
22. Planning meeting	Scrum	S1,S6,S39
23. Demos	Scrum	S6,S39
24. Unit test	XP	S4,S39
25. Acceptance test	XP	S6,S39
26. Small team	XP-Scrum	S1,S2,S4
27. Velocity	XP-Scrum	S32,S39
28. Release planning	XP-Scrum	S2,S39
29. Scrum master	Scrum	S6,S39

TABLE 7 Positive influences for agile practices.[S4,S9,S13,S39]

Set of agile practices	Impact on operation
1. Planning Game 2. Task Prioritization 3. Collective Code Ownership 4. Automated Testing 5. Stand-up	Improve the morale among team members.
1. Planning Game 2. Acceptance Test 3. Demo	Increased customer satisfaction.
1. Planning Game 2. Task Prioritization 3. Collective Code Ownership 4. Automated Testing 5. Stand-up meeting	Using these practices will lead to improving productivity.
1. Pair-Programming 2. Code Reviews 3. Coding Standards 4. Small Team	Simplified knowledge transfer among the teams.
1. Task Prioritization 2. User Stories 3. Product Backlog	Improved Prioritization of tasks.
1. Unit Testing 2. Continuous Integration	Fix bugs and merge them with code rapidly.
1. Coding Standards 2. Code Reviews 3. Daily Builds	Increase the product quality.
1. User Stories 2. Product Backlog 3. Velocity	Increase the accuracy of time and efforts forecasts.
1. User Stories. 2. Task Prioritization 3. Refactoring	Enhanced test suites.
1. Refactoring 2. Coding Standards 3. Iteration Planning	Increasing the index of maintainability.
1. Retrospective 2. Refactoring	Improved program comprehension.
1. Pair-Programming	Improve designing at the abstraction level.

A second factor of the quality attribute is maintainability. Cloud Computing reduces the running cost of maintaining software by eliminating the need to hire more expert engineers to manage servers. Cloud Computing provides services/resources in terms of Infrastructure, Platform, Software, and everything as a service (X as a service), PaaS provides hardware resources and tools that enable developers to develop and maintain their projects for both local and global environments.[S35] SaaS provides services for automating updating, simplifying administration, and patch management. All these services are based on Pay-As-You-Go. In addition, SaaS provides users the ability to share the codes among team members in the global environment using GitHub, CodeSpaces, SourceForge, and so on.[S35] Another service of a Cloud platform is Test as a service (TaaS), which provides tools to simplify the test process.[S35] IaaS provides infrastructure for users with full control of hardware, operating system, network, data, and servers with low cost, less effort, and effective service. Also, IaaS provides both physical and virtual servers.[S35] Due to the availability of the infrastructure, the process of maintaining, testing, and updating the programs will be easier due to the reduction of the time and effort spent during these processes.[S30]

Development infrastructure: Cloud computing provides information technology resources that can be assigned to customers wherever they are if the Internet is available. Thus, Cloud Computing can be a good environment for distributed development teams. For this reason, many organizations have employed Cloud environments to achieve their projects.[S35] Cloud computing platforms provide many hardware and software capabilities that facilitate the software development process. Also, Cloud Computing provides storage, virtualizations, and networks. All these resources can be available rapidly without the need to interact with service providers.[S26]

TABLE 8 Agile maintenance challenges in a local environment.

Agile maintenance challenges	Possible solutions
1. **Iterative development**: Applying tasks within an iteration (sprint) is common in Agile methods. In maintenance, this leads to lost synergy and maintenances objectives. This is due to the emergency tasks in maintenance that causes interrupting the current sprint.[S2,S13,S16,S37]	According to a study,[S13] which was applied at Aveva company, to improve the performance of sprint, they encouraged the maintenance personnel to compare the sprints between team members.
2. **Focusing on work objectives**: One of the maintenance challenges is to focus on the current objective, due to the interruption caused to emergency requests, which loses and weakens focus on objectives.[S13,S16,S37]	One possible solution stated in[S13] is to allocate some time for unpredictable tasks in sprint planning. In [S12] proposed a model that treats with an urgent request, by pausing the current sprint after checking this request if it is corrective maintenance and storing the current sprint in the version control system to resume this sprint after handling the urgent request.
3. **Team working closely**: One of the maintenance challenges is when software maintainers work on individual tasks according to their project and acquire experience in one perspective instead of acquiring experience in multi-domain.[19,S13,S37]	In[S13], Aveva maintainers have developed an informal method, which provides expert assistance to the team instead of using pair programming.
4. **Close customer involvement**: In maintenance, it is a good idea to involve software maintainers with in-site customers, but it is difficult to apply this action. Thus, it is difficult to achieve closed customer's involvement.[19,S13,S37]	Giving customers who have a good relationship with the team more responsibilities in dealing with difficult problems and evaluation. This you will be able to engage the customer in a more effective manner.[S13]
5. **Face-to-face communication**: In maintenance, the tasks are diverse, for that reason face-to-face communication is considered as a needless action.[19,S13,S37]	In,[S13] they solved this challenge through intensive preparation by the Scrum master before the meeting. Thus, the team will regain confidence in the engineers' estimates.
6. **Light documentation**: In maintenance, documentation is considered as an irreplaceable part that helps maintainers to understand the system. The Agile method emphasizes on reducing documentation (light documentation) as possible as.[S9,S13,S15,S17,S37]	Simplify document and organize it in structure manner to include useful information and avoid heavy documentation that conflicted with Agile principles.[S13,S37]
7. **Frequent testing:** Usually, comprehensive system testing is considered impractical, because this kind of testing is limited and valid only for the current fix. The Agile method focuses on automated testing, but it is difficult to apply that for maintenance, so maintainers need to find proper solutions that affect positively the quality of the product.[S13,S37]	One of the effective solutions to moving to collective ownership, which helps engineers to test others' fixes, doing that will encourage co-education between team members.[23,S13]
8. **Motivation through collective ownership**: collective ownership is difficult to apply since maintainers work on individual tasks, i.e., there are no common tasks between each other, thus getting motivation is not easy.[S13,S37]	Devoting knowledge sharing among team members and making it a regular practice process.[S13,S37]
9. **Knowledge transfer through openness**: it might be no benefits of Openness and information sharing if there is no interconnection of tasks. Therefore, there is no transfer of knowledge in this case[S13,S37]	It is necessary to find a possible way to transfer knowledge between the maintainer's team, one of these ways according to[S13] is to apply the traditional Scrum board with some modification.

Collaboration: It is one of the significant principles of the Agile approach. Specifically, in a distributed environment, it is considered an essential activity among the team to reduce the time and effort needed for development. There are several forms of cooperation represented by sharing the documents among team members, such as test reports, SRS documents, prototypes, sprint planning, backlog planning, stand-up meetings, and Scrum of Scrum. Clouds provide tools that can simplify development, for instance, Cloud IDE, GitHub, Eclipse, and others.[S8, S11,S26,S35] Additionally, communication among teams is considered a way of collaboration used for sharing and discussing information through the available tools, such as Wikis, Skype, emails, etc.[S34,S35,S43]

Regard to transparency: In the Agile method, many activities that indicate transparency, such as burn-down charts, user stories, and product backlog, these activities have to be visible to team members in the Scrum method to predicate the next sprint and future tasks. Therefore, any defects discovered by the team need to be addressed according to the priorities listed in the backlog. In a Cloud Computing environment, all these activities can be achieved with the support of project management tools, such as JIRA, Mingle, Rally, ScrumWorks, Trac, VersionOne, Xplanner, AgileFant, Stats, jmxtrans, Metrics, Esper, Ganglia, Graphite, Cube, CloudSpoke on Topcoder, Trustie, and REDMINE, not to mention code management tools including Code Spaces, GitHub, GoogleCode, SourceForge, Unfuddle, GitLab, BitBucket, Git, and Mercurial.[S8,S34,S35] More advantages of the combination of Agile and Cloud Computing are listed in Table 10.

TABLE 9 Agile maintenance challenges for the global environment.

Agile maintenance challenges	Possible solutions
1. **Problems regarding Communication**: several problems existed such as misunderstanding problems, difficulties with a face-to-face meeting, and the high cost of the communications.[S1,S7,S10,S21,S22,S29,S33]	• Summarize details about tasks, thus will increase the communication speed. • Organize meetings between product owners and Scrum master. • Simplify communication between team members using Wiki pages or any other tools. • Keep up-to-date information between team members. • Determined end meeting time. • Specify local Scrum master as an alternative product owner[S31,S33]
2. **Problems regarding control**: most common problems related to control are difficulties related to quality control and process, rearrangement of priorities, and difficulties related to cooperation.[S7,S10,S29,S33,S38]	• For multiple backlogs, we can use Scrum of Scrums for high-level coordination. • Organize meetings according to time zones for each team. • Specify backlog for each region. • Synchronize each local backlog at the end of the day. • Use tracking bugs and tasks system. • Training the team on Agile principles and practices. • Discussion about current sprint progress and backlog three times weekly.[S31,S33]
3. **Problems regarding trust**: There are two main factors; trust among team members and team morale.[S7,S22,S27,S33,S38]	• Adapt the Scrum process, regarding place value for people. • Use Instant Messaging (IM) to simplify communication between the team. • Enhance interconnectedness and collaboration through face-to-face interaction.[S23,S31,S33]

TABLE 10 The benefits of using agile in the cloud.

Factors	Advantages
1. Supporting software testing.	Software developers do not care about managing, maintaining, and plan hardware resources, such as the number of servers necessary for the testing process.[S8,S11,S34]
2. Virtualization	Virtualization helps to support parallel development required for small iterations in Agile development. In addition, users can scale up the number of processors and storage, which are considered essential factors for the parallel environment.[S8, S34]
3. Traceability	Traceability is one of the most important features in Cloud Computing as it enables developers to observe the changing code globally by all team members.[S8,S34]
4. Prototypes and Demo	In the Cloud, deploying prototypes and sharing them with customers is easier than in traditional environments.[S8]
5. Performance	Due to the nature of the Cloud that supports developing software in a parallel manner, establishing the principle of decentralization in decision-making among team members, it naturally leads to improving overall performance.[S8,S34,S41, S45]
6. Reduce cost	Cloud computing works on the "pay-as-you-go" principle, which helps to reduce Agile development method expenditures.[S8,S11,S34,S41,S45]

5 | DISCUSSION

The SM study conducted in this paper, revealed many benefits in applying Agile methods to maintenance, as well as challenges to overcome. In this section, a discussion of RQs based on the SM study is given.

RQ1: What are the existing studies in Agile software maintenance?

Forty-eight studies related to our RQs out of 322 between 2000 and 2022 were searched using different databases are analyzed. There are 14 case studies, 4 literature reviews, 4 models, 2 frameworks, 1 tool, and 4 other studies within this period. Based on the existing studies, there is a need to conduct research on tools, methods, and models to cover all aspects of this domain.

RQ2: What are the applications and benefits of using an Agile approach for maintenance?

There are many benefits for software maintainers in using Agile, such as reducing cost, effort, time, and others.[S2,S4,S9,S12,S13,S15,S20,S28,S39,S46] We have listed many advantages in RQ2. In our opinion, though, there is a need to verify the existing benefits listed above by using computational methods to support the existing case studies. Regarding the Cloud Computing environment, there is a need to verify the benefits of using Agile-maintenance based on the Cloud environment.

Rehman et al[S12] proposes that the Scrum maintenance model is helpful in dealing with emergency and urgent sprints during the normal sprint work (i.e., proper planning, version control, user involvement, and testing). Scrum models give value to the client in the prioritization of sprint execution. Choudhari and Suman[S46] note that XP presents various ways of practice in software maintenance. They suggest that using XP speeds up the maintenance process with less effort and produces a more maintainable code for future maintenance and evolution.

RQ3: What are the Agile practices (XP/Scrum) for local and global environments?

Agile methods have many practices devoted to its 12 principles some of which can be used locally while others need to be adopted for the global environment. We have identified 12 practices for Scrum and XP.[S2,S3,S4,S6,S21,S24,S25,S27,S29,S42,S56] Due to the differences between the on-premises environment and the Cloud, Agile practices need to adapt to Cloud Computing environments accordingly.

In distributed and global software development, there are numerous challenges, such as communication among team members,[29–33] lack of physical proximity,[34,35] team cohesion,[30] shared context, and knowledge.[31,35,S45] Knowledge management issues becomes exponentially important in the context of distributed and global software development efforts as knowledge is spread across locations and coordinating and integrating this knowledge is challenging[36] and unavailability of team members.[37] For establishing collaborative relationships among the team members, frequent visits are made.[30,37] There are two types of visits, seeding visits and maintaining visits. Seeding visits occur during the early stages of the project. Their aim is to build connections.[38,S25] and are held during the early stages of development cycles.[31] Whereas, maintaining visits are shorter and aim to build collaboration connections.[37]

Ramesh et al.[31] suggest that collaboration can be established by visits of members from onshore and offshore sites. Developers, customers, and managers should meet one another to build good collaboration relationships. The challenge is the traveling of members for meetings in distributed locations. To cope, Paasivaara et al[S29] report that Agile practices using Scrum use agility-supporting practices for distributed projects. These practices include frequent visits, unofficial distributed meetings, and regular gatherings. Some other agile methodologies, such as Scrum[38] and XP,[39] have successfully customized distributed projects. Scrum uses frequent and open communication; for this, they use multiple communication tools, such as videoconferencing, Internet telephony, desktop sharing, and chat for formal and informal meetings.[S24]

Paasivaara et al[S24] state that frequent visits are needed, especially at the beginning of a project and in critical project phases such as testing or planning. For offshore environments, Agile development methodologies use Web-conferencing for real-time and interactive collaboration. Furthermore, white-boarding and code-sharing, are used for kick-off meetings, daily Scrum meetings, and pair-programming activities during Agile development.[S25]

RQ4: What are the Agile practices (XP/Scrum) that are used for maintenance?

We identified different Agile practices used for the maintenance process in these studies[S1,S2,S4,S6,S32,S37,S39,S47] The current studies have adapted Agile practices to the local and global environment. There is a need for studies to adapt Agile maintenance practices in the Cloud Computing environment. Mira Kajko et al[S6] have observed that agile maintenance has different degrees of agility in various phases which are lower at the pre-implementation phases and higher at the implementation phase. This observation concludes that Agile practices help to some extent in the evaluation and maintenance of software.[S6] Extreme programming in terms of software maintenance, helps to understand the code by using different patterns. What is more, it reduces the code size by minimizing code complexity, which can be carried out by stripping the unused code. In this way, the required staff is reduced by 25%, ultimately increasing productivity three-fold.[S46]

RQ5: What are the advantages of applying Agile practices for maintenance?

According to the studies,[S4,S9,S13,S39] Scrum and XP practices mentioned in RQ4, are directed towards the maintenance process to improve customer satisfaction, increase interaction between team members, simplify testing,20 facilitate code review, increase code quality, and other advantages stated in RQ5. Conducting studies on the adaptation of agile practices in the Cloud environment, along with their negative and positive effects, will facilitate the software maintenance process.

RQ6: What are the challenges in using Agile maintenance for a local and global environment?

Many advantages of applying Agile practices for maintenance are stated[S4,S9,S13,S39] and explained in RQ5, some of which are proved through a case study in a real industrial environment. In addition, in RQ6, we address some challenges of using Agile in local and global environments and state the possible solutions according to the literature studies.[17,40,S2,S9,S13,S15–S17,S37] We can benefit from these solutions when adopting Agile maintenance in the Cloud environment.

Agile development is iteration-based, and common changes and task lists are absent in maintenance.[S2] To elaborate, maintenance is often done by interruption of the client in requirement change, and sprints are subjected to interruption by client's requirements. Furthermore, there are a number of common delivery points or integrated releases used in agile maintenance.[S16] It is also noted that maintenance is done with customer intervention in different systems. For this, maintenance engineers use face-to-face communication and often work side-by-side with customers. In Agile, usually, documentation is frequently ignored and incomplete.[S16]

Software maintenance involves high-cost activities as compared to software development.[41–44] This is due to the fact that efforts spent in the maintenance process are greater compared to those in the development phase. Due to the high cost, the decisions made during the maintenance process are more critical for the system.

RQ7: What are the advantages of applying Agile methods in a Cloud Computing environment?

The studies in the literature mention many advantages that software professionals obtain by integrating Agile methods with Cloud Computing. The main benefits are related to scalability, maintainability, availability, collaboration, and transparency.[45,S8,S11,S21,S22,S26,S27,S29,S30,S33–S35,S43] Accordingly, the Agile methods have already been used for development processes in the Cloud Computing environment and have provided many promising results.[S35] For this reason, adopting the Agile maintenance process is possible, but it needs further investigation by researchers. The SM revealed that more attention is needed from the researchers in practical terms for Agile software development and Cloud Computing.

Cloud Computing provides different benefits in managing servers by reducing capital investment and running costs.[46] Younas et al[S35] note that Cloud Computing supports Agile software development by providing hardware infrastructure, virtualization of resources, automated testing, automatic deployment, and communication platforms. Furthermore, Cloud Computing reduces the time-to-market of software while increasing transparency through the smooth execution of user story, backlog management, and traceability.[S35]

6 | THREATS TO VALIDITY AND LIMITATIONS

There are several limitations and threats that may affect the outcome of the study. To avoid these threats, in this study we took certain actions and measures as follows:

Well-known electronic databases (IEEE Xplore, Scopus, Springer, Web of Science, Science direct, and other databases such as (ACM, Google Scholar, CiteSeer (X), ACSI, etc. have been investigated to explore answers to research questions in this study. The papers searched in this study are limited to these databases. Thus, there is a possibility of missing relevant papers.

Bias over publications: Researchers held several meetings to minimize researcher bias. However, there might be some papers in some electronic databases that we have missed in this research. Further, new papers are also published very frequently and therefore, we might have missed some new papers published recently.

Searching and selecting studies: Our strategy was designed to find as many articles as possible. We built a wide range of search strings over digital databases. Although the search outcomes included many irrelevant articles, we decided to keep the search string fixed, as such, we do not miss any relevant papers. We considered regulating the search string to diverse our search results. We applied the inclusion and exclusion principle to choose relevant papers. All researchers in this study examined the principle of selection based on the quality evaluation criteria mentioned above in the quality assessment. Most biases were expected in quality evaluation because research questions may be difficult to justify. Also, the score of papers may vary from one author to another.

- Regarding the research questions, in order to fully cover all aspects of the Agile maintenance issues, the authors use the brainstorming method to determine what questions the research addresses.
- Concerning the related studies, for the purpose of obtaining all relevant papers, many databases were searched using various terms and synonyms related to the research questions.
- Regarding the inclusion and exclusion criteria for the selected studies, to overcome the problem of individual bias, the studies were selected and excluded by unanimous agreement among the authors.

6.1 | Construct validity

After the final paper pool was constructed, the data was systematically mapped along research questions by using the Goal Question Metric (GQM) approach. The GQM approach facilitates minimizing risks of construct validity by providing traceability between goals and questions.

6.2 | External validity

The study was constrained by the limitation of the selected digital library databases, which only included papers in English. While the titles and abstracts of some non-English papers were available, the full text of these papers was not accessible. Consequently, even though these papers were relevant to the topic, they could not be considered for analysis.

6.3 | Conclusion validity

Conclusion validity in a systematic mapping (SM) assesses the accuracy and trustworthiness of drawn conclusions about a research study. It focuses on completeness, accuracy, and representativeness. This SM on Agile software maintenance rigorously explores various dimensions like methodologies and challenges, ensuring a thorough analysis. To enhance accuracy, strict inclusion criteria and transparent methods were used, minimizing bias. The study's precision strengthens the validity of results, offering a reliable foundation for further research and decision-making. Additionally, a diverse range of studies was included, making the findings representative and applicable to stakeholders in Agile software maintenance.

6.4 | Quality assessment

In the customary approach, quality assessment involves two researchers, one conducting the assessment and the other providing validation. In our review, however, the second researcher carried out both data extraction and quality assessment. This deviation from the standard practice introduces the possibility of biases, particularly in cases where research questions are challenging to justify. Consequently, the assessment scores may vary depending on the researcher involved.

7 | IMPLICATIONS FOR RESEARCHERS AND PROFESSIONALS

As part of this SM study, we identified the following implications for researchers and professionals:

This SM study has yielded important insights for both researchers and practitioners in the field of Agile software maintenance. Researchers can explore various Agile maintenance practices in the Cloud environment, which offer several benefits over conventional approaches. However, challenges exist in terms of manageability, scalability, communication, collaboration, and transparency. Furthermore, there is a scarcity of empirical studies on Agile maintenance practices in local and global environments. Therefore, we encourage future research in this area. Additionally, studies on the application of Agile maintenance practices in various software organizations are recommended to facilitate knowledge sharing among researchers and practitioners. Interestingly, our study found that XP and Scrum-based Agile maintenance models are prevalent in the literature. Future studies should focus on exploring other Agile methods. Comparing the adoption of Agile maintenance practices in small, medium, and large software development organizations could provide valuable insights. Investigating how different software quality attributes are affected by Agile maintenance practices in the Cloud environment can benefit both researchers and software professionals. Finally, industrial studies and case studies are needed to gain insights into practitioners' perspectives on Agile methods in software maintenance in the Cloud environment.

8 | CONCLUSIONS AND FUTURE WORK

This study employed a Systematic Mapping approach to explore the landscape of Agile software maintenance, addressing seven key research questions. We analyzed 48 relevant articles published between 2000 and 2022, focusing on the utilization of Agile methodologies in maintenance processes, associated benefits, specific practices for local and global environments, potential challenges, and proposed solutions.

Key findings revealed the following:

- Limited Research Focus: Despite the widespread adoption of Agile development, research on Agile maintenance remains relatively scarce, with only 48 articles identified within the examined timeframe.
- Environmental Considerations: Local environments facilitate collaboration and infrastructure compatibility, while global settings present challenges due to geographical dispersion and cultural differences. Cloud computing emerged as a viable solution, offering a centralized platform for seamless collaboration and addressing diverse infrastructure needs, enabling successful Agile maintenance practices across both contexts.
- Challenges and Solutions: Communication difficulties, lack of physical proximity, team cohesion, shared context and knowledge, and unavailability of team members were identified as major challenges in global environments. Agile practices, such as Scrum and Extreme Programming, helped overcome these hurdles by fostering communication, collaboration, and knowledge sharing.
- Differentiating Agile Maintenance: Agile software maintenance differs from traditional approaches in terms of requirements, customer involvement, and software design.
- Performance Enhancement: Cloud computing significantly improved the performance of Agile maintenance models, streamlining the software maintenance process. Notably, the Scrum model was found to enhance client satisfaction, while Extreme Programming effectively addressed various maintenance-related issues.
- Varied Agility Levels: Agile maintenance models demonstrated varying degrees of agility across different phases of the process. The implementation phase exhibited higher agility compared to the pre-implementation phase.

This study provides valuable insights into the current state of Agile software maintenance, highlighting its benefits and challenges in diverse environments. Cloud computing's potential to address critical challenges in global settings and enhance performance underlines its importance for the future of Agile maintenance practices. Further research is needed to explore the long-term impacts and effectiveness of Agile approaches in various contexts, contributing to the continuous improvement and optimization of software maintenance processes. Furthermore, an interesting area for future research lies in the examination of how Artificial Intelligence and automation will influence the efficacy of agile maintenance processes within the Cloud environment. A recent comprehensive systematic literature review on the utilization of Large Language Models (LLMs) in the domain of software engineering[47] revealed promising outcomes, particularly in tasks such as code review, program repair, and debugging. The horizon of future research beckons us to expand the scope of LLMs to encompass all facets of agile software maintenance processes within the Cloud. Delving further into the research landscape, we find the need to explore the synergies between DevOps and Agile maintenance, fostering collaborative bridges between development and operations teams, a paramount objective during software maintenance, be it in local or global contexts. Lastly, a promising frontier in research involves the development of novel metrics and performance measurement techniques to evaluate the effectiveness of agile maintenance processes within real-world development environments.

In conclusion, the following insights are also noted:

- Agile can resolve major issues faced in traditional software maintenance.
- Cloud Computing services contribute a vital part in facilitating software maintenance.
- A majority of the studies highlight XP- and Scrum-based Agile maintenance models.
- The application of the Agile methods for software maintenance can contribute to simplifying the maintenance process due to the characteristics and advantages of this methodology.
- Several advantages/benefits are gained by adopting the Agile methods in the Cloud computing environment.

CONFLICT OF INTEREST STATEMENT

Authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

ORCID

Alok Mishra https://orcid.org/0000-0003-1275-2050
Ali Yazici https://orcid.org/0000-0001-5405-802X

REFERENCES

1. Manifesto for Agile Software Development, *agilemanifesto.org*. http://agilemanifesto.org. Accessed Oct. 04, 2022.
2. PRISMA-P Group, Moher D, Shamseer L, et al. Preferred reporting items for systematic review and meta-analysis protocols (PRISMA-P) 2015 statement. *Syst Rev*. 2015;4(1):1. doi:10.1186/2046-4053-4-1

3. Rizwan M, Qureshi J, Sayid I. Scheme of Global Scrum Management Software The Proposed L-Scrumban Methodology to Improve the Efficiency of Agile Software Development View project Integeration and Documentation of Agile Software Methodologies View project Scheme of Global Scrum Management Software. *Int J Inf Eng Electron Business*. 2015;2:1-7. doi:10.5815/ijieeb.2015.02.01
4. Nazir A, Raana A, Fahad Khan M. Cloud Computing ensembles Agile Development Methodologies for Successful Project Development. *Int J Modern Educ Comput Sci*. 2013;5(11):28-35. doi:10.5815/ijmecs.2013.11.04
5. Wang W, R.A.S.D.i. t, "Digital.ai Agility an Agile Planning Solution," *Collab.net*., 2020. https://www.open.collab.net. Accessed Oct. 04, 2022.
6. Portelli B. "The Beauty of Agile in the Cloud," *AgileConnection*, 2010. https://www.agileconnection.com/article/beauty-agile-Cloud. Accessed Oct. 04, 2022.
7. Mell P, Grance T. "The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology Special Publication 800-145," 2011.
8. Franken S, Kolvenbach S, Prinz W, Alvertis I, Koussouris S. -NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/). Peer-review under responsibility of Institute of Communication and Computer Systems. ScienceDirect HOLACONF -Cloud Forward: From Distributed to Complete Computing CloudTeams: Bridging the Gap between Developers and Customers during Software Development Processes-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/). Peer-review under responsibility of Institute of Communication and Computer Systems. *Proc Comput Sci*. 2015;68:188-195. doi:10.1016/j.procs.2015.09.234
9. Tuli A, Hasteer N, Sharma M, Bansal A. Empirical investigation of agile software development. *ACM SIGSOFT Softw Eng Notes*. 2014;39(4):1-6. doi:10.1145/2632434.2632447
10. Kalem S, Donko D, Boskovic D. "Agile methods for cloud computing Agile Methods for Cloud Computing," 2013: 1355-1359.
11. Sommerville I. *Software engineering 8*. Pearson; 2016.
12. Bennett KH, Rajlich VT, Software maintenance and evolution. *Proceedings of the conference on the future of Software engineering - ICSE'00*, 2000. doi:10.1145/336512.336534
13. Dybå T, Dingsøyr T. Empirical studies of agile software development: A systematic review. *Inf Softw Technol*. 2008;50(9–10):833-859. doi:10.1016/j.infsof.2008.01.006
14. Malhotra R, Chug A. "Comparative analysis of agile methods and iterative enhancement model in assessment of software maintenance," *IEEE Xplore*, Mar. 01, 2016. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7724470. Accessed Oct. 04, 2022.
15. Janicic M. Maintenance and maintainability within agile software development. Science of Maintenance, May 2021.
16. Almashhadani M, Mishra A, Yazici A, Younas M. Challenges in Agile Software Maintenance for Local and Global Development: An Empirical Assessment. *Inform*. 2023;14(5):261. doi:10.3390/info14050261
17. Mishra D, Mishra A. Research trends in management issues of global software development: evaluating the past to envision the future. *J Global Inf Technol Manag*. 2011;14(4):48-69.
18. Inayat I, Salim S, Kasirun Z. Agile-based Software Product Development Using Cloud Computing Services: Findings form a Case Study 2013.
19. Tsai W-T, Wu W, Huhns MN. Cloud-Based Software Crowdsourcing. *IEEE Int Comput*. 2014;18(3):78-83. doi:10.1109/mic.2014.46
20. Brereton P, Kitchenham BA, Budgen D, Turner M, Khalil M. Lessons from applying the systematic literature review process within the software engineering domain. *J Syst Softw*. 2007;80(4):571-583. doi:10.1016/j.jss.2006.07.009
21. Staffs K. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. EBSE; 2007.
22. Highsmith J, Cockburn A. Agile software development: the business of innovation. *Computer*. 2001;34(9):120-127. doi:10.1109/2.947100
23. Sukumaran S, Sreenivas A. "Identifying test conditions for software maintenance," 2005.
24. Almudarra F, Qureshi B. Issues in Adopting Agile Development Principles for Mobile Cloud Computing Applications. *Proc Comput Sci*. 2015;52:1133-1140. doi:10.1016/j.procs.2015.05.131
25. Grubb P, Takang AA. *Software Maintenance: Concepts and Practice*. 2nd ed. World Scientific; 2003. Accessed: Oct. 04, 2022. [Online]. Available: https://books.google.com.tr/books?id=ZBzJCgAAQBAJ&printsec=copyright&redir_esc=y#v=onepage&q&f=false
26. Pressman RS, Maxim B. *Software engineering: a practitioner's approach*. Mcgraw-Hill Higher Education; 2014.
27. Zagal JP, Ahues RS, Voehl MN. Maintenance-oriented design and development: a case study. *IEEE Softw*. Jul. 2002;19(4):100-106. doi:10.1109/ms.2002.1020296
28. Silcox PH. Software Maintenance Management. *IEE Proc E Comput Dig Techniq*. 1980;127(6):277. doi:10.1049/ip-e.1980.0056
29. Berczuk S. "Back to Basics: The Role of Agile Principles in Success with an Distributed Scrum Team," 2007.
30. Simons M. *Internationally Agile: The Challenges of Offshore Development*. InformIT; 2002.
31. Ramesh B, Cao L, Mohan K, Xu P. Can distributed software development be agile? *Commun ACM*. 2006;49(10):41-46. doi:10.1145/1164394.1164418
32. Farmer M. "DecisionSpace infrastructure: agile development in a large, distributed team," *IEEE Xplore*, Jun. 01, 2004. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1359801 (accessed Oct. 04, 2022).
33. Therrien E. "Overcoming the Challenges of Building a Distributed Agile Organization," *IEEE Xplore*, Aug. 01, 2008. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4599507 (accessed Oct. 04, 2022).
34. Holmstrom H, Conchuir E, Agerfalk P, Fitzgerald B. "Global Software Development Challenges: A Case Study on Temporal, Geographical and Socio-Cultural Distance," *2006 IEEE International Conference on Global Software Engineering (ICGSE'06)*, Oct. 2006, doi: 10.1109/icgse.2006.261210.
35. Yap M. "Follow the Sun: Distributed Extreme Programming Development" Proceedings of Agile Conference, 24th-29th July 2005, pp. 218-224., 2005.
36. Mishra D, Aydin S, Mishra A, Ostrovska S. Knowledge management in requirement elicitation: Situational methods view. *Comput Stand Interf*. 2018;56:49-61. doi:10.1016/j.csi.2017.09.004
37. Braithwaite K, Joyce T. XP Expanded: Distributed Extreme Programming. *Extr Progr Agile Process Softw Eng*. 2005;180-188. doi:10.1007/11499053_21
38. Schwaber K, Beedle MA. *Agile Software Development with SCRUM*. 1st ed. Prentice Hall; 2001.
39. Beck K. *Extreme programming explained: embrace change*. Addison-Wesley; 2004.
40. Ralyté J, Lamielle X, Arni-Bloch N, Léonard M. A framework for supporting management in distributed information systems development. In: *2008 Second International Conference on Research Challenges in Information Science*. IEEE; 2008.

41. Reyes W, Smith R, Fraunholz B. *Agile approaches to software maintenance: an exploratory study of practitioner views*. presented at the IGI Publishing, Vancouver; 2007.
42. Capilla R, Dueñas JC, Ferenc R. A retrospective view of software maintenance and reengineering research. *J Softw Evol Process*. 2011;25(6):569-574. doi:10.1002/smr.548
43. Thomazinho HCS, L'Erário A, Fabri JA. A Case Study on the Strategy of Maintaining Commercial Software with a Large Number of Users. *J Inf Syst Eng Manag*. 2017;2(4):1-9. doi:10.20897/jisem.201723
44. Stojanov Z, Stojanov J. Exploring software maintenance process characteristics by using inductive thematic analysis. Procc ICAIIT2016. 2016:9-17. doi:10.20544/aiit2016.02
45. Hashmi SI, Clerc V, Razavian M, et al. "Using the Cloud to Facilitate Global Software Development Challenges," *2011 IEEE Sixth International Conference on Global Software Engineering Workshop*, Aug. 2011, doi: 10.1109/icgse-w.2011.19.
46. Misra SC, Mondal A. Identification of a company's suitability for the adoption of cloud computing and modelling its corresponding Return on Investment. *Math Comput Model*. 2011;53(3–4):504-521. doi:10.1016/j.mcm.2010.03.037
47. Hou X, Zhao Y, Liu Y, et al. Large Language Models for Software Engineering: A Systematic Literature Review. ArXiv, abs/2308.10620; 2023.

APPENDIX: SELECTED STUDIES (48)

S1. Jain N. "Offshore agile maintenance," *IEEE Xplore*, Jul. 01, 2006. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1667596 (accessed Oct. 04, 2022).
S2. Poole C, Huisman JW. Using extreme programming in a maintenance environment. *IEEE Softw*. 2001;18(6):42-50. doi:10.1109/52.965801
S3. Dagnino A, "An evolutionary lifecycle model with Agile practices for software development at ABB," *IEEE Xplore*, Dec. 01, 2002. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1181514. Accessed Oct. 04, 2022.
S4. Svensson H, Host M. "Introducing an agile process in a software maintenance and evolution organization," *IEEE Xplore*, Mar 01, 2005. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1402140. Accessed Oct. 04, 2022.
S5. Hanssen GK, Yamashita AF, Conradi R, Moonen L. "Maintenance and agile development: Challenges, opportunities and future directions," *IEEE Xplore*, Sep. 01, 2009. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5306278. Accessed Oct. 04, 2022.
S6. Kajko-Mattsson M, Nyfjord J, "A Model of Agile Evolution and Maintenance Process," *IEEE Xplore*, Jan. 01, 2009. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4755767. Accessed Oct. 04, 2022.
S7. Jalali S, Wohlin C. "Agile Practices in Global Software Engineering – A Systematic Map," *2010 5th IEEE International Conference on Global Software Engineering*, Aug. 2010, doi: 10.1109/icgse.2010.14.
S8. Younas M, Ghani I, Norhayati D, Jawawi A, Khan M, Jawawi D. A Framework for Agile Development in Cloud Computing Environment Textual similarity analysis of software artifacts in regression testing View project LEARNING ANALYTICS FRAMEWORK TO SUPPORT SELF-REGULATED LEARNING STRATEGIES IN MASSIVE OPEN ONLINE COURSES (MOOC) IN REDUCING DROPOUT View project A Framework for Agile Development in Cloud Computing Environment . *J Int Comput Serv*. 2016;(5):67-74. doi:10.7472/jksii.2016.17.5.67
S9. Prochazka J. Agile Support and Maintenance of IT Services. *Inf Syst Dev*. 2011;597-609. doi:10.1007/978-1-4419-9790-6_48
S10. Talluri M, Haddad HM. Best managerial practices in agile development. Proceedings of the 2014 ACM Southeast Regional Conference, Mar. 2014, doi: 10.1145/2638404.2638456.
S11. Singh S, Chana I. Introducing Agility in Cloud Based Software Development through ASD. *Int J u- e- Serv Sci Technol*. 2013;6(5):191-202. doi:10.14257/ijunesst.2013.6.5.17
S12. Ur Rehman F, Maqbool B, Qasim Riaz M, Qamar U, Abbas M. "Scrum Software Maintenance Model: Efficient Software Maintenance in Agile Methodology," 2018 21st Saudi Comput. Soc. Natl. Comput. Conf., pp. 1–5., 2018.
S13. Heeager LT, Rose J. Optimising agile development practices for the maintenance operation: nine heuristics. *Emp Softw Eng*. 2015;20(6):1762-1784. doi:10.1007/s10664-014-9,335-7
S14. Tarwani S, Chug A. Agile Methodologies in Software Maintenance: A Systematic Review. *Inform*. 2016;40:415.
S15. Kumar B. The Sway of Agile Processes over Software Maintainability. *Int J Comput Appl*. 2015;109(1):25-29. doi:10.5120/19152-0581
S16. Abdullah S, Subramaniam M, Anuar S. Improving the governance of software maintenance process for agile software development team. *Int J Eng Technol*. 2018;7(4):113-117.
S17. Yamato Y, Katsuragi S, Nagao S, Miura N. Software maintenance evaluation of agile software development method based on OpenStack. *IEICE Trans Inf Syst*. 2015;E98.D(7):1377-1380. doi:10.1587/transinf.2015edl8049
S18. Reyes W, Fraunholz B. Agile Approaches to Software Maintenance: An Exploratory Study of Practitioner Views. Management Association International Conference, Vancouver, British Columbia, Canada May 19-23, 2007.
S19. Cardoso M, Filho M. Agile Processes for the Maintenance Cycle: A Smarter Work Cycle for a Smarter Planet. 2012.
S20. Rudzki J, Hammouda I, Mikkola T. Agile experiences in a software service company. 2009 35th Euromicro Conference on Software Engineering and Advanced Applications, 2009, doi: 10.1109/seaa.2009.31.
S21. Beecham S, Noll J, Richardson I. Using agile practices to solve global software development problems—a case study. *2014 IEEE International Conference on Global Software Engineering Workshops*, Aug. 2014, doi: 10.1109/icgsew.2014.7.
S22. McHugh O, Conboy K, Lang M. Agile practices: the impact on trust in software project teams. *IEEE Softw*. May 2012;29(3):71-76. doi:10.1109/ms.2011.118
S23. Batra D. Modified agile practices for outsourced software projects. *Commun ACM*. Sep. 2009;52(9):143-148. doi:10.1145/1562164.1562200
S24. Paasivaara M, Durasiewicz S, Lassenius C. Using scrum in distributed agile development: a multiple case study. *Proc Int Conf Global Softw Eng*. 2009;17. doi:10.1109/ICGSE.2009.27
S25. Danait A. "Agile Offshore Techniques -A Case Study," Agile Development Conference (ADC'05), 2005.
S26. Butt SA, Tariq MI, Jamal T, Ali A, Diaz Martinez JL, De-La-Hoz-Franco E. Predictive Variables for Agile Development Merging Cloud Computing Services. *IEEE Access*. 2019;7:99273-99282. doi:10.1109/access.2019.2929169
S27. Jain P, Levine D, Kircher M, Corsaro A. Distributed eXtreme Programming, Extreme Programming and Flexible Processes in Software Engineering, 2001, pp. 66-71.

S28. Agarwal M, Majumdar R. Software maintainability and usability in agile environment. *Int J Comput Appl*. 2013;68(4):30-36. doi:10.5120/11569-6873
S29. Paasivaara M, Durasiewicz, S, Lassenius C. Distributed agile development: using scrum in a large project. 2008 IEEE International Conference on Global Software Engineering, Aug. 2008, doi: 10.1109/icgse.2008.38
S30. Cocco L, Mannaro K, Concas G. A model for global software development with cloud platforms. 2012 38th Euromicro Conference on Software Engineering and Advanced Applications, Sep. 2012, doi: 10.1109/seaa.2012.67.
S31. Akbar MA, Mahmood S, Alsalman H, Razzaq A, Gumaei A, Riaz M. Identification and prioritization of cloud based global software development best practices. *IEEE Access*. 2020;8:191242-191262. doi:10.1109/ACCESS.2020
S32. Shaw S. Using Agile Practices in a Maintenance Environment. 2007.
S33. Lee S, Yong HS. Distributed agile: project management in a global environment. *Emp Softw Eng*. 2009;15(2):204-217. doi:10.1007/s10664-009-9,119-7
S34. Butt SA. Study of agile methodology with the cloud. *Pac Sci Rev B: Human Soc Sci*. 2016;2(1):22-28. doi:10.1016/j.psrb.2016.09.007
S35. Younas M, Jawawi DN, Ghani I, Fries T, Kazmi R. Agile development in the cloud computing environment: A systematic review. *Inf Softw Technol*. 2018;103:142-158. doi:10.1016/j.infsof.2018.06.014
S36. Abrahamsson P, Ronkainen J. Agile Software Development Methods: Review and Analysis. Bolzano Raspberry Pi Experiment View project FLEXI project View project," 2002.
S37. Ibrahim KS, Yahaya J, Mansor Z, Deraman A. "The Emergence of Agile Maintenance: A Preliminary Study," International Conference on Electrical Engineering and Informatics (ICEEI), 2019.
S38. Awar K, Shujah M, Sameem I, Hafeez Y. "A Model for Applying Agile Practices in Distributed Environment: A Case of Local Software Industry,"" in Proceedings of 2017 International Conference on Communication, Computing and Digital Systems C-CODE, 2017.
S39. Devulapally GK. *Agile in the context of Software Maintenance A Case Study*. Blekinge Institute of Technology; 2015.
S40. Pino FJ, Ruiz F, García F, Piattini M. A software maintenance methodology for small organizations: Agile_MANTEMA. *J Softw Evol Proc*. 2011;24(8):851-876. doi:10.1002/smr.541
S41. Jyothi VE, Kaitepalli N, Rao KN. Effective implementation of agile practices—in collaboration with cloud computing. *Res Art Int J Curr Eng Technol*. 2014;4(3):1690-1693.
S42. Qumer A, Henderson-Sellers B. An evaluation of the degree of agility in six agile methods and its applicability for method engineering. *Inf Softw Technol*. 2008;50(4):280-295. doi:10.1016/j.infsof.2007.02.002
S43. Younas M, Jawawi DNA, Mahmood AK, Ahmad MN, Sarwar MU, Idris MY. Agile Software Development Using Cloud Computing: A Case Study. *IEEE Access*. 2020;8:4475-4484. doi:10.1109/access.2019.2962257
S44. Yu L, Mishra A. Risk Analysis of Global Software Development and Proposed Solutions. *Automatika*. 2010;51(1):89-98. doi:10.1080/00051144.2010.11828358
S45. Manuja M. "Moving Agile based projects on Cloud," in *IEEE International Advance Computing Conference (IACC)*, 2014, vol. pp. 1392–1397.
S46. Choudhari J, Suman U. "Iterative Maintenance Life Cycle Using eXtreme Programming," *2010 International Conference on Advances in Recent Technologies in Communication and Computing*, Oct. 2010, doi: 10.1109/artcom.2010.52.
S47. Choudhari J, Suman U. Extended iterative maintenance life cycle using eXtreme programming. *ACM SIGSOFT Softw Eng Notes*. 2014;39(1):1-12. doi:10.1145/2557833.2557845
S48. Choudhari J, Suman U. An Empirical Evaluation of Iterative Maintenance Life Cycle Using XP. *ACM SIGSOFT Softw Eng Notes*. 2015;40(2):1-14. doi:10.1145/2735399.2735406

How to cite this article: Almashhadani M, Mishra A, Yazici A. Software maintenance practices using agile methods towards cloud environment: A systematic mapping. *J Softw Evol Proc*. 2024;36(11):e2698. https://doi.org/10.1002/smr.2698