

Article

Challenges in Agile Software Maintenance for Local and Global Development: An Empirical Assessment

Mohammed Almashhadani ¹, Alok Mishra ^{2,*}, Ali Yazici ³ and Muhammad Younas ⁴

¹ Graduate School of Nature and Applied Sciences, Atilim University, Incek, 06830 Ankara, Türkiye; mhycs90@yahoo.com

² Faculty of Engineering, Norwegian University of Science and Technology, 2815 Gjovik, Norway

³ Department of Software Engineering, Atilim University, Incek, 06830 Ankara, Türkiye; ali.yazici@atilim.edu.tr

⁴ Information Technology, Government College University Faisalabad (GCUF), Faisalabad 38000, Pakistan

* Correspondence: alok.mishra@ntnu.no

Abstract: Agile methods have gained wide popularity recently due to their characteristics in software development. Despite the success of agile methods in the software maintenance process, several challenges have been reported. In this study, we investigate the challenges that measure the impact of agile methods in software maintenance in terms of quality factors. A survey was conducted to collect data from agile practitioners to establish their opinions about existing challenges. As a result of the statistical analysis of the data from the survey, it has been observed that there are moderately effective challenges in manageability, scalability, communication, collaboration, and transparency. Further research is required to validate software maintenance challenges in agile methods.

Keywords: software maintenance; software maintenance life cycle; agile software maintenance



Citation: Almashhadani, M.; Mishra, A.; Yazici, A.; Younas, M. Challenges in Agile Software Maintenance for Local and Global Development: An Empirical Assessment. *Information* **2023**, *14*, 261. <https://doi.org/10.3390/info14050261>

Academic Editor: Aneta Poniszewska-Maranda

Received: 13 February 2023

Revised: 20 April 2023

Accepted: 21 April 2023

Published: 27 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/> 4.0).

1. Introduction

According to the IEEE definition [1], software engineering describes a framework for all phases of the software development life cycle (SDLC) whereas software maintenance is about software modification after delivery, the correction of faults, and performance improvement. Development refers to creating a product from scratch, i.e., a “new product”. In contrast, software maintenance refers to continuous support to fix bugs or adapt the product after it is delivered to the end user [2,3]. Software maintenance is different from software development due to the dependence of maintenance on program comprehension [4]. The software maintenance life cycle has models such as Osborn, Boehm, Iterative Enhancement, IEEE, and reuse-oriented [4,5]. Most of these models are based on traditional software development technologies [5,6]. In Figure 1, the stages of software development and software maintenance are shown.

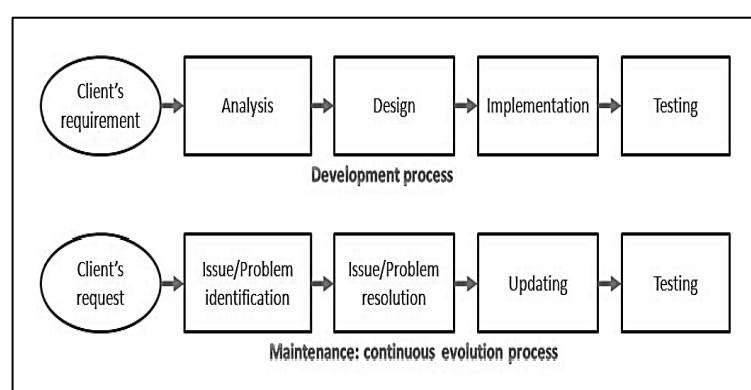


Figure 1. A comparison between the SDLC and the SMLC process [5].

The maintenance process is an indispensable part of maintaining the software due to constant environmental changes, and the software needs to adapt to such environments [7]. The main objective of the maintenance process is to keep the system stable and working according to users' requirements. There are many reasons to perform software maintenance, such as fixing errors, preventing system failure, and performing system updates to keep pace with changes [8]. Changing the system according to customer needs (for example, by improving performance) and adding a new functionality are among the other reasons for software maintenance.

According to [9], the need for the maintenance process can be summarized under the following categories:

- Fixing bugs: detecting and removing bugs to keep the system running;
- Adapting the system: adapting the system to continue operations within the changing business environment;
- Supporting users: providing users with backup and assistance when needed.

Agile software maintenance is positioned between traditional maintenance and agile methods [4]. Agile maintenance has many benefits, which can be acquired by using different agile practices extracted from a variety of agile families [10,11]. Despite the many benefits of agile software maintenance, several studies [12–21], have reported different kinds of challenges. In this study, these challenges are identified and categorized based on the literature. Verification of the challenges is carried out by conducting a survey.

The rest of the paper is organized as follows. Section 2 introduces the background information about challenges faced by agile professionals in the local and global environment. In Section 3, the research method is discussed. Section 4 describes the research hypothesis. Section 5 presents the results and the validity of the study. Section 6 provides a discussion of the survey results. In Section 7, threats to validity and limitations are reported. Finally, we conclude with future research directions.

2. Background

Agility is based on iterative development which can be considered as an umbrella activity for different agile methods. Agile methods are used to overcome the difficulties encountered by traditional software development methodologies by focusing on customer satisfaction, changing requirements, encouraging small self-organizing teams, and collaborating directly with customers [22].

Several agile practices have been used in the maintenance process: planning games, small releases, refactoring, pair programming, collective code ownership, continuous integration/automated release, test-driven development, iteration planning, 40 h weeks, standup meetings, coding standards, etc. [10,23–25].

Many studies have reported several benefits of using the agile approach for the maintenance process, which contribute to increasing product quality [26,27], speeding the process, improving communication [12,13], improving productivity [13,28], and so on.

In contrast, many studies [12,14–21] have stated the challenges that teams face while using agile maintenance both in local (on premises) and global environments (distributed). The studies [12,14,15] reported the different challenges faced by teams in the local environment while using agile software maintenance. These challenges are summarized as follows.

In [10,12,14,15], a problem related to iterative development (problems in applying tasks within the sprint) was defined. This problem can cause a lack of transparency between team members, weakness in project manageability, and an increase in the need for more software infrastructure.

Another challenge was reported in [12,14,29], which focuses on work objectives. This challenge can lead to a lack of transparency between team members, which affects the project's quality.

Many studies [12,14,29] have stated the problem of teams working closely. The implications of this problem can affect transparency and collaboration between team members, which will affect the project's quality as result.

Studies [12,14,29] referred to challenges related to customer involvement and face-to-face communication, which can affect the application of one of the agile principles. These problems can negatively affect communication and collaboration between team members and cause weakness in manageability due to the different views between the client and the work team.

One of the drawbacks of agile methods is light documentation [10,12,14,28,30]. This problem can affect project manageability and thus decrease the project's quality.

Frequent testing is one of the challenges reported by studies [12,14]. This problem is affected by two main factors: scalability and providing software infrastructure.

Motivation through collective ownership and knowledge transfer through openness are two other problems reported in [12,14]. These challenges can be caused by a lack of communication and collaboration among team members, leading to weakness in project transparency.

The combination of agile methods and global software engineering (GSE), known as distributed agile software development, has been applied in many agile methods such as extreme programming (XP), scrum, and so on. Many advantages related to quality, cost, and time can be gained from this hybrid approach [31].

Despite the benefits achieved by this combination, many other agile software maintenance challenges in the global environment have been reported, such as problems regarding communication [16–18], control [18–20], and trust [19–21].

In this study, the survey method (a quantitative method) will be applied to verify the challenges faced by the agile software maintenance teams in the local and global environment.

3. Research Method

A survey method was used to identify the challenges that teams faced in local and global environments. Surveying is a method of questioning knowledge, attitudes, and behaviors [32]. Moreover, the survey was carried out through direct communication with software professionals. In this article, the steps proposed by Kitchenham [33] are used to conduct a survey to verify the challenges faced by software maintenance teams.

The survey method is utilized to gather the necessary data to support theoretical research on the practical side and to formulate the research questions properly. In order to design the survey, techniques, and measurements used in relevant studies [25,34] were effectively used.

Among the many factors reported in the literature studies, this study focuses only on manageability, scalability, software infrastructure, communication, collaboration, and transparency.

As the initial step of the survey, in Tables 1 and 2, the challenges mentioned above in the local (on-premise) and global (distributed) environments are summarized, respectively, based on the related studies.

Table 1. Challenges in the local environment.

Agile Maintenance Challenges	Classification of the Challenges According to Quality Factors
1. Iterative development [10,12,14,15].	Transparency: Lack of transparency in a project leads to unclear views concerning that project. Manageability: Lack of manageability leads to conflict in the sprint. Software Infrastructure: The lack of infrastructure affects the development and maintenance process iteratively.
2. Focusing on work objectives [12,14,29].	Transparency: Lack of transparency in the project leads to unclear views regarding the project.

Table 1. Cont.

Agile Maintenance Challenges	Classification of the Challenges According to Quality Factors
3. Close team work [12,14,29].	Transparency: Lack of transparency in the project can affect team collaboration. Collaboration: Lack of collaboration results in poor teamwork.
4. Close customers involvement [12,14,29].	Collaboration and Communication: Lack of collaboration and communication between the customers and the teams will affect the project's quality. Manageability: Lack of manageability could lead to conflict between customers' requirements and the maintenance team.
5. Face-to-face communication [12,14,29].	Collaboration and Communication: Lack of collaboration and communication between the customers and the teams can affect the project's quality.
6. Light documentation [10,12,14,28,30].	Manageability: Lack of manageability could lead to conflict between the customers' requirements and the maintenance team.
7. Frequent testing [12,14].	Scalability: The lack of scalability may affect the frequent testing process, which is considered an essential factor in the maintenance process. Software Infrastructure: The lack of infrastructure provision affects the frequent testing.
8. Motivation through collective ownership [12,14].	Communication and collaboration: Lack of collaboration and communication between teams will discourage collective ownership.
9. Knowledge transfer through openness [12,14].	Collaboration and Communication: Lack of collaboration and communication between teams will discourage knowledge transfer. Transparency: Lack of transparency in a project is likely to affect knowledge transfer among the maintenance team members.

Table 2. Challenges in the global environment.

Agile Maintenance Challenges	Classification of the Challenges According to Quality Factors
1. Challenges regarding Communication [16–20,23,35].	Communication: Lack of communication between teams will lead to obstacles in collaboration within a global environment. Manageability: Lack of manageability will lead to conflicts in performing tasks.
2. Challenges regarding control [18–20,35,36].	Manageability: Lack of manageability will lead to poor control over the project. Transparency: Lack of transparency in a project is likely to affect the degree and quality of control over a project.
3. Challenges regarding trust [17,19,20,35,37].	Communication: Lack of communication will lead to a lack of trust. Collaboration: Lack of collaboration between teams will lead to a lack of trust. Manageability: Excessive monitoring can lead to low trust.

In the next step, the survey is designed using Google Forms. This simple survey tool can be used for multiple purposes, such as designing and analyzing surveys, without requiring prior programming knowledge.

This survey uses the design steps given in [38] as follows:

1. Determine the survey sections;
 2. Determine the question type for each section;
 3. Design the sequence of the questions for each section.
1. **Determine survey sections**

The survey questions are determined per the challenges identified in the related studies for the distributed (global) environment and the local (on-site) environment.

2. **Determine the question type for each section**

For this purpose, we chose multiple-choice questions, followed by an open question for the responders at the end of each section.

3. Design the sequence of the questions for each section.

The questions in the survey are categorized into two main parts, namely, general (demographic) information about the respondent and questions related to agile software maintenance challenges, as follows:

Section A—General Information

Section B—Twenty questions (details of the questions can be found in Appendix A) These were related to agile software maintenance challenges and distributed, according to factors, into five sections:

- (B1) Manageability: 4 questions;
- (B2) Scalability: 4 questions;
- (B3) Software Infrastructure: 5 questions;
- (B4) Communication and Collaboration: 3 questions;
- (B5) Transparency: 4 questions.

The survey was designed and the respondents chose one or more of the three options (local, global, none), to specify whether there are any challenges in the local and/or global environments.

After designing and identifying the questions, as a preliminary study, the survey was shared with the authors to discuss the relevance of the questions based on the related literature and the survey questions were finalized.

Then, a pilot study was carried out by sending the survey to five software professionals. After receiving their feedback and considering critical comments, the survey was finalized and deployed.

4. Hypothesis

The primary purpose of this study is to verify the challenges faced by agile professionals during the maintenance process. Therefore, using the challenges described in Tables 1 and 2, the research hypotheses are formulated.

The independent variable is the use of agile maintenance in an organization, and the dependent variable is the level of agile maintenance challenges according to five factors (manageability, scalability, software infrastructure, communication and collaboration, and transparency) based on the literature studies.

- **Manageability:** the ability to organize and manage resources, such as human resources, in a way that enables the completion of the project through a commitment to the specific content, considering quality factors such as traceability and control, to achieve the agile principle that states “Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely” [22]. Based on this, the following hypothesis is built:

Hypothesis (H1) :*There are challenges with respect to management in agile maintenance projects.*

Hypothesis (H2): *There are no challenges with respect to management in agile maintenance projects.*

Hypothesis (H1) is an alternative hypothesis and Hypothesis (H2) is the null hypothesis.

- **Scalability:** the ability to scale out resources such as storage, networks, processors, and so on. According to [22], “Continuous attention to technical excellence and good design enhances agility”. In maintenance, more resources are necessary to achieve the process when adding new system functionality. The hypothesis for this factor is

Hypothesis (H3): *There are challenges with respect to scaling up resources in agile maintenance projects.*

Hypothesis (H4): *There are no challenges with respect to scaling up resources in agile maintenance projects.*

Hypothesis (H3) is an alternative hypothesis and Hypothesis (H4) is the null hypothesis.

- **Software Infrastructure:** The first principle in the agile manifesto is “Our highest priority is to satisfy the customer through early and continuous delivery of valuable software” [22]. In order to promote the principle, the delivery process must take place early, which necessitates speed in configuring the infrastructure, because traditional environments require time and cost and face problems in providing the necessary resources. Therefore, another hypothesis was constructed as follows:

Hypothesis (H5): *There are challenges with respect to providing the necessary infrastructure in agile maintenance projects.*

Hypothesis (H6): *There are no challenges with respect to providing the necessary infrastructure in agile maintenance projects.*

Hypothesis (H5) is an alternative hypothesis and Hypothesis (H6) is the null hypothesis.

- **Communication and Collaboration:** One of the Agile principles is “Business people and developers must work together daily throughout the project” [22]. So, communication and collaboration are critical factors in agile methods. As illustrated in previous sections, there are challenges concerning communication and collaboration among team members, both between each other and with customers. For this reason, the following hypothesis was constructed:

Hypothesis (H7): *There are challenges with respect to communication and collaboration in agile maintenance projects.*

Hypothesis (H8): *There are no challenges with respect to communication and collaboration in agile maintenance projects.*

Hypothesis (H7) is an alternative hypothesis and Hypothesis (H8) is the null hypothesis.

- **Transparency:** During the systematic literature review and from the survey, several challenges regarding transparency were identified. Consequently, a hypothesis was constructed as follows:

Hypothesis (H9): *There are challenges with respect to transparency in agile maintenance projects.*

Hypothesis (H10): *There are no challenges with respect to transparency in agile maintenance projects.*

Hypothesis (H9) is an alternative hypothesis and Hypothesis (H10) is the null hypothesis.

5. Results

The data collected from the survey respondents were reviewed and then verified in terms of validity and reliability, followed by identifying the research technique, research community, research sample, and research instrument. Additionally, these data were analyzed using appropriate statistical analysis methods. To achieve precision in the analysis, we include a degree of interpretation for these results; for this purpose, measurements, classifications, and interpretation methods are used to extract the key conclusions. The descriptive-analytical method is employed to achieve the objective of the study.

5.1. Research Sample

The sample targeted in the survey is made up of organizations that use agile maintenance activities. After deploying the survey using Google Forms, the responses were collected, and the total sample size was found to be 56 participants, 42 of which (i.e., 75%) work at organizations using agile methods for software maintenance.

The following sections are a comprehensive description of the research sample, described by the required statistical methods according to several demographic variables, including country, experience, type of the organization, the nature and duration of use of agile, and the maintenance process.

5.1.1. Characteristics of the Sample Study

The sample size is calculated according to many characteristics described in the following steps.

1. Distribution of the study sample according to country

The results are as follows: 2.4% from Brazil, 19% from India, 2.4% from Iraq, 11.9% from Jordan, and 52.4% from Pakistan, as the largest category, followed by 9.5% from Turkey and 2.4% from the United Kingdom.

2. Distribution of the study sample according to experience

Based on the results, 28.6% of the professionals' experience was between 0 and 4 years, 23.8% 5 and 7 years, 26.2% 8 and 10 years, as the largest, and finally, 21.4% over 10 years.

3. Distribution of the study sample according to the size of an organization

Accordingly, 50% of the professionals work at micro-size organizations, which is the largest category of the sample, 28.6% at small-size organizations, 4.8% at medium-size organizations, and 16.7% at large-size organizations.

4. Distribution of the study sample according to organization's nature

On this basis, it has been observed that 11.9% of the sample is employed by national organizations, 83.3% by international organizations, which is the largest category of the sample, and 4.8% by other categories.

5. Distribution of the study sample according to the duration of using Agile

It is evident that 52.4% of the sample have been using agile methods for 0 to 4 years, which is the largest category of the sample, 16.7% for 5 to 7 years, 14.3% for 8 to 10 years, and 16.7% have done so or more than 10 years.

6. Distribution of the study sample according to the maintenance process

It can be observed that 11.9% of the professionals work on-site (local), 21.4% by remote access to the customer servers (global), 57.1% depend on the nature of the fault and use either the local or global environment, which is the largest category of the sample, 4.8% operate as support, and finally, 4.8% choose either local, company-remote, or cloud-provided remote work.

5.1.2. Survey Validity

After the preparation step, the survey was reviewed to check the validity and reliability before analyzing the results.

The Pearson correlation coefficient is calculated between each paragraph (question) and the remaining ones (see Table 3). Additionally, the internal consistency among the corresponding section (five factors) is calculated as shown in Table 4.

Table 3. Pearson correlation coefficient among the survey questions.

No	Correlation Coefficient						
1	0.897 **	6	0.811 **	11	0.872 **	16	0.703 **
2	0.799 **	7	0.705 **	12	0.727 **	17	0.863 **
3	0.786 **	8	0.859 **	13	0.742 **	18	0.770 **
4	0.618 **	9	0.796 **	14	0.679 **	19	0.704 **
5	0.704 **	10	0.607 **	15	0.731 **	20	0.647 **

** All correlations among the variables are above 0.6, which is considered a high correlation, according to Table 5 below.

Table 4. Pearson correlation coefficient among the axis (five factors).

No	Factors	No of Items	Correlation Coefficient
1	Manageability	4	0.909
2	Scalability in agile software maintenance	4	0.949
3	Software infrastructure	5	0.979
4	Communication and Collaboration	3	0.888
5	Transparency	4	0.952

As shown in Table 4, all correlation values among the variables are above 0.9, which displays a very high correlation according to the scale provided in Table 5. The correlation between the two variables is noteworthy (2-tailed). According to [39], this is an acceptable score. As per the Pearson correlation coefficients between each paragraph (20 questions), shown in Table 3, and the axis (five factors group questions), given in Table 4, there is excellent validity regarding the internal consistency.

Table 5. The scale of Pearson correlation coefficients [40].

Scale of Correlation Coefficient	Value
$0 < r \leq 0.19$	Very low correlation
$0.2 \leq r \leq 0.39$	Low correlation
$0.4 \leq r \leq 0.59$	Moderate correlation
$0.6 \leq r \leq 0.79$	High correlation
$0.8 \leq r \leq 1.0$	Very high correlation

The Kaiser–Meyer–Olkin (KMO) and Bartlett's tests are applied for five factors to measure how data are suited to factor analysis (see Table 6). The results above reveal KMO scores of 0.715, 0.559, 0.801, 0.656, and 0.807, which are above 0.5. Bartlett's test value of significance is at <0.01 , and according to Kaiser [41], the KMO scores and Bartlett's test values are considered acceptable.

Table 6. KMO and Bartlett's test for the five sections (five factors group questions).

Factors	KMO	Kaiser–Meyer–Olkin measure of sampling adequacy		0.715
		Approx. chi-square	Df	
1. Manageability	Bartlett's test of sphericity	Approx. chi-square	Df	25.038
				<0.001
				6
2. Scalability	Bartlett's test of sphericity	Approx. chi-square	Df	0.559
				<0.001
				6
3. Software Infrastructure	Bartlett's test of sphericity	Approx. chi-square	Df	38.477
				<0.001
				10
4. Communication and Collaboration	Bartlett's test of sphericity	Approx. chi-square	Df	49.426
				<0.001
				3
5. Transparency	Bartlett's test of sphericity	Approx. chi-square	Df	0.656
				<0.001
				6

5.1.3. Survey Reliability

To verify the survey's reliability, a Cronbach's alpha test was applied. In Table 7, the results of the test are summarized.

Table 7. Cronbach's alpha coefficients.

Factors	No of Items	Cronbach's Alpha
Manageability	4	0.794
Scalability	4	0.812
Software infrastructure	5	0.888
Communication and Collaboration	3	0.877
Transparency	4	0.939
Total	20	0.969

From Table 7, it can be concluded that all Cronbach's alpha values are above 0.7, and the reliability of the survey is assured according to [42].

5.2. Descriptive Statistical Analysis

To identify the challenges in relation to the agile maintenance factors, the chi-square, mean, standard deviation, rank, and degree for each question are calculated within the dimension, using descriptive statistics and the SPSS software. The following illustrates the intervals for the three-point (low, medium, high) Likert scale [43]:

- Low: 1.00–1.66
- Medium: 1.67–2.33
- High: 2.34–3.00

In Table 8, the level of challenge (high, medium, low) is determined using the mean values.

Table 8. Agile maintenance challenge levels.

Survey Question P_i # (Challenges) **	Response %				Chi-Square	Df	Sig	Mean	Std	Rank	Degree
	None	Local	Global	Local and Global							
Manageability											
P1	21.4	33.3	28.6	16.7	2.762	3	0.430	2.24	0.790	1	Medium
P2	33.3	33.3	26.2	7.1	7.714	3	0.052	2.00	0.826	3	Medium
P3	26.2	47.6	11.9	14.3	13.429 *	3	0.004	2.00	0.733	4	Medium
P4	31.0	35.7	16.7	16.7	4.857	3	0.183	2.02	0.811	2	Medium
Scalability											
P5	28.6	23.8	35.7	11.9	5.048	3	0.168	2.19	0.862	3	Medium
P6	19.0	23.8	42.9	14.3	7.905 *	3	0.048	2.38	0.795	1	High
P7	23.8	28.6	35.7	11.9	5.048	3	0.168	2.21	0.842	2	Medium
P8	33.3	26.2	26.2	14.3	3.143	3	0.370	2.07	0.867	4	Medium
Infrastructure											
P9	19.0	31.0	26.2	23.8	1.238	3	0.744	2.31	0.780	2	Medium
P10	23.8	31.0	23.8	21.4	0.857	3	0.836	2.21	0.813	3	Medium
P11	23.8	38.1	19.0	19.0	4.095	3	0.251	2.14	0.783	4	Medium
P12	21.4	23.8	31.0	23.8	0.857	3	0.836	2.33	0.816	1	Medium
P13	28.6	40.5	11.9	19.0	7.714	3	0.052	2.00	0.796	5	Medium
Communication and Collaboration											
P14	31.0	14.3	40.5	14.3	8.476 *	3	0.037	2.24	0.906	2	Medium
P15	35.7	31.0	11.9	21.4	5.619	3	0.132	1.98	0.841	3	Medium
P16	26.2	19.0	35.7	19.0	3.143	3	0.370	2.29	0.864	1	Medium
Transparency											
P17	31.0	23.8	28.6	16.7	2	3	0.572	2.14	0.872	3	Medium
P18	23.8	28.6	31.0	16.7	2	3	0.572	2.24	0.821	1	Medium
P19	23.8	28.6	23.8	23.8	0.286	3	0.963	2.24	0.821	2	Medium
P20	33.3	28.6	26.2	11.9	4.286	3	0.232	2.05	0.854	4	Medium

* Significant chi-square value. ** P_i 's are defined in Appendix A.

Some essential points should be clarified before interpreting the results, namely, the interpretation of the significant and non-significant chi-square values. According to [44], if there are no statistically significant differences among the options of none, local, global, and local and global, this is referred to as a 'non-significant' chi-square. In other words, the respondents have rated the options equally as challenges, which implies that all options have the same degree of challenge.

In contrast, if there are statistically significant differences among these options, this is called a 'significant' chi-square, whereby the respondents indicated that one of the options is more important than the others.

5.2.1. Manageability

The statistics in Table 8 revealed challenges facing agile maintenance teams concerning management; the results of the chi-square were as follows: 2.762, 7.714, 4.857, respectively for P1, P2, and P4 (non-significant), and 13.429* for P3 (significant).

Additionally, the results show that the degree is medium, the mean 2.07, and the standard deviation 0.79. Thus, the ratio of challenges faced by the agile maintenance team regarding manageability is low. The results indicate homogeneity of respondents' opinions regarding the administrative challenges facing agile maintenance professionals.

5.2.2. Scalability

The statistics in the above table revealed challenges facing agile maintenance teams regarding scalability and interoperability, showing that the chi-square values are 5.048, 5.048, 3.143, respectively for P5, P7, and P8 (non-significant), and 7.905* for P6 (significant).

Additionally, the results show the mean as 2.21 and the standard deviation as 0.84. Accordingly, the challenging factors faced by agile professionals regarding scalability are low, indicating homogeneity in the respondents' views on these issues.

5.2.3. Software Infrastructure

The results related to infrastructure challenges indicate that the chi-square values are 1.238, 0.857, 4.095, 0.857, and 7.714, respectively, for P9, P10, P11, P12, and P13 (non-significant). At the same time, the results regarding scalability are a mean of 2.20 and a standard deviation of 0.80. Accordingly, the challenges agile professionals face regarding infrastructure are low, indicating homogeneity in the respondents' views on these issues.

5.2.4. Communication and Collaboration

The results related to communication and collaboration challenges are chi-square values of 8.476* for P14, which is significant, and 5.619, 3.143, respectively, for P15, and P16, which are non-significant. Additionally, the degree is medium, the mean is 2.17, and the standard deviation is 0.87; therefore, the low value indicates homogeneity in the respondents' views concerning these issues.

5.2.5. Transparency

The results shown in Table 8 related to transparency challenges are chi-square values of 2, 2, 0.286, and 4.286, respectively for P17, P18, P19, and P20, which are non-significant. At the same time, the results show that the respondents' opinions about the transparency challenges faced by agile maintainers were homogeneous. Where the degree is medium, the mean is 2.17, and the standard deviation is 0.84.

6. Discussion

An industrial survey was used in this study to identify the existing challenges concerning agile maintenance. Questions with different viewpoints were organized and deployed for agile professionals to establish their opinions about these challenges.

Many studies [12,14,15,29,45] have shown the agile management challenges facing agile teams, such as iterative development, focusing on work objectives, the team working closely, and knowledge transfer through openness. These were confirmed by the conducted questionnaire, as the questionnaire indicated the existence of management problems; the Chi-square values for management challenges were 2.762, 7.714, and 4.857, respectively, which are non-significant, and 13.429*, which is considered significant. The mean was 2.07, and the standard deviation was 0.79. Thus, the degree of agile management challenge is low, indicating homogeneity in the opinions of the study sample members on manageability challenges.

The agile method is effective in small projects. However, maintenance teams face some challenges concerning scalability in large projects [34,37,46–49]. The challenges were also confirmed by professionals in the survey, with a mean of 2.21. The standard deviation is 0.84, and the chi-square values are 5.048, 5.048, and 3.143, respectively, which are non-significant, and 7.905*, which is significant.

The literature studies [50–54] reported many challenges regarding communication among team members in a global environment. The survey confirmed these challenges, in which the results showed that the degree of communication challenges is medium; the mean is (2.17). The standard deviation is (0.87), and the chi-square values are 8.476*, which is significant, and 5.619, 3.143, respectively, which are non-significant. According to the participants' opinions, the communication and collaboration challenges faced by agile maintainers are minimal. The results confirm the existence of problems in the distributed environment regarding communication and collaboration.

Additionally, previous studies [12,14,17–21,35,36] reported challenges regarding transparency, such as trust, control, and knowledge transfer through openness. The survey results confirmed these challenges, where the chi-square values are 2, 2, 0.286, and 4.286,

respectively, which are non-significant; the mean was 2.24, the standard deviation was 0.821, and the degree was medium.

One of the most critical challenges facing the agile teams is the availability of infrastructure, which is stated by [12,14], and confirmed by the questionnaire, where according to the results, the mean was 2.33, the standard deviation was 0.816, and the degree was medium, while the chi-square values were 1.238, 0.857, 4.095, 0.857, and 7.714, respectively (non-significant).

7. Threats to Validity and Limitations

In this section, some aspects of the threats to the validity of the survey are discussed. Accordingly, sampling bias, response bias, selection bias, question order, and question-wording bias are as follows:

- **Sampling bias:** The sample size (56) of the survey may not be enough to generalize the results. However, based on the literature on the software engineering domain [55,56], this sample size is appropriate for generalizing the survey results.
- **Response bias:** In some cases, the respondents may not provide accurate or honest responses to the survey questions. To overcome this threat, some illogical answers are excluded.
- **Selection bias:** This happens when the sample of participants does not represent the survey population. In this study, participants are from different countries.
- **Question order and question-wording bias:** To avoid these, a pilot test is conducted to ensure that the questions are formulated in the correct order and questions are framed in the right manner.

Like any other empirical research, limitations must be acknowledged. First, there might be variations in the respondents' backgrounds/profiles. The survey design population may also impact the outcome of the survey. Sample size also affects the results. As it is known, the larger sample affects the objectivity of the results and gives an accurate idea of the community's opinion. Although samples were collected from different countries, the sample sizes of those countries were not quite close to each other, which makes it difficult to compare the situations in different countries. It would have been more interesting to explore if there are any differences in terms of the outcomes in light of the background of the respondents' profiles, sectors, and countries.

8. Conclusions and Future Research

This study was conducted to identify agile software maintenance challenges faced in five sections based on quality factors. A survey was carried out following a series of steps, and the questions were shared with agile professionals. The collected data were analyzed using statistical methods. The results reveal challenges in agile software maintenance, irrespective of local and global environments, concerning management, scalability, software infrastructure, communication and collaboration, and transparency. The survey's reliability was ensured through a Cronbach's alpha test.

As per the literature review, the factors under survey have challenges; however, the results of the survey revealed them to be of a medium degree.

Author Contributions: Conceptualization, A.M., M.A.; Methodology, A.M., M.A., A.Y. and M.Y.; Validation, M.A., A.Y. and M.Y.; Formal analysis, A.M., M.A., A.Y. and M.Y.; Investigation, A.M., M.A. and A.Y.; Date Duration, A.M.; Writing—original draft, A.M., M.A. and A.Y.; Writing—review & editing, A.M., M.A., A.Y. and M.Y.; Supervision, M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not Applicable.

Conflicts of Interest: Authors declare no conflict of interest.

Appendix A

The Challenges in Manageability

P1: Challenges related to the availability of experts for configuring software and hardware resources and agile maintenance experts.

P2: Lack of technical support for the improvement of agile maintenance practices.

P3 Weakness in managing maintenance team projects.

P4: Challenges related to lack of management commitment to support agile maintenance members.

The Challenges in Scalability

P5 Challenges related to software's ability to interact with other systems in an organization.

P6 Challenges related to frequent planning of interactions among team members.

P7 Challenges related to organizing large-size projects in agile software maintenance.

P8 Challenges related to documentation of large-size projects in agile software maintenance.

The Challenges in Infrastructure

P9 Challenges related to configuration infrastructure for implementing software maintenance.

P10 Challenges related to the availability of infrastructure and resources in agile software maintenance projects.

P11 Challenges related to necessary software tools for artifact management in agile software maintenance.

P12 Challenges related to the necessary testing server for both frequent and automated tests in agile software maintenance.

P13 Challenges related to the necessary server for frequent delivery of software.

The Challenges in Communication and Collaboration

P14 Challenges concerning geographically distributed teams to reduce the issues related to communication, coordination and collaboration, etc., among team members.

P15 Challenges related to using advanced tools and techniques for continuous communication between team members.

P16 Challenges related to obtaining customer feedback and involvement in projects.

The Challenges in Transparency

P17 Challenges related to providing source code management in agile software maintenance.

P18 Challenges related to sharing data, code, built results and test reports in agile software maintenance.

P19 Challenges related to traceability mechanism for project artifacts in agile software maintenance.

P20 Challenges related to the client's monitoring the progress of software maintenance in agile projects.

References

1. Thayer, R.H. Glossary Software Engineering. *IEEE Softw.* **2003**, *20*, c3. [[CrossRef](#)]
2. Sommerville, I. *Software Engineering*; Pearson: Boston, MA, USA; Munich, Germany, 2011.
3. Pressman, R.S. *Software Engineering: A Practitioner's Approach*, 7th ed.; Engineering & Mathematics; McGraw Hill: New York, NY, USA, 2010.
4. Devulapally, G.K. Agile in the context of Software Maintenance: A Case Study'. Thesis, Blekinge Institute of Technology, Karlskrona, Sweden, 2015. Available online: <https://www.semanticscholar.org/paper/Agile-in-the-context-of-Software-Maintenance-%3A-A-Devulapally/20693500d3c2d7184d86be6f2aac619a31351557> (accessed on 1 February 2023).
5. Liguò, Y.; Mishra, A. Risk Analysis of Global Software Development and Proposed Solutions. *Automatika* **2010**, *51*, 89–98. [[CrossRef](#)]
6. Choudhari, J.; Suman, U. Iterative Maintenance Life Cycle Using EXtreme Programming. In Proceedings of the 2010 International Conference on Advances in Recent Technologies in Communication and Computing, Washington, DC, USA, 16–17 October 2010; Available online: <https://ieeexplore.ieee.org/xpl/conhome/5654667/proceeding> (accessed on 16 December 2022).
7. Calhoun: The NPS Institutional Archive Faculty and Researcher Publications Faculty and Researcher Publications. The State of Software Maintenance. 1987, Volume 1, pp. 303–310. Available online: <http://hdl.handle.net/10945/40278> (accessed on 16 December 2022).

8. Akhlaq, U.; Yousaf, M. Impact of Software Comprehension in Software Maintenance and Evolution. 2010. Available online: <https://www.diva-portal.org/smash/get/diva2:829443/FULLTEXT01.pdf> (accessed on 14 March 2023).
9. Grubb, P.; Takang, A.A. *Software Maintenance*; World Scientific: Singapore, 2003.
10. Poole, C.; Huisman, J.W. Using Extreme Programming in a Maintenance Environment. *IEEE Softw.* **2001**, *18*, 42–50. [CrossRef]
11. Kajko-Mattsson, M.; Nyfjord, J. A Model of Agile Evolution and Maintenance Process. In Proceedings of the 2009 42nd Hawaii International Conference on System Sciences, Waikoloa, HI, USA, 5–8 January 2009.
12. Heeager, L.T.; Rose, J. Optimising Agile development practices for the maintenance operation: Nine heuristics. *Empir. Softw. Eng.* **2015**, *20*, 1762–1784. [CrossRef]
13. Mishra, D.; Mishra, A. A review of non-technical issues in global software development. *Int. J. Comput. Appl. Technol.* **2011**, *40*, 216–224. [CrossRef]
14. Ibrahim, K.S.K.; Yahaya, J.; Mansor, Z.; Deraman, A. The Emergence of Agile Maintenance: A Preliminary Study. In Proceedings of the International Conference on Electrical Engineering and Informatics (ICEEI) 2019, Bandung, Indonesia, 9–10 July 2019.
15. Abdullah, S.; Subramaniam, M.; Anuar, S. Improving the Governance of Software Maintenance Process for Agile Software Development Team. *Int. J. Eng. Technol.* **2018**, *7*, 113–117.
16. Beecham, S.; Noll, J.; Richardson, I. Using Agile Practices to Solve Global Software Development Problems—A Case Study. In Proceedings of the 2014 IEEE International Conference on Global Software Engineering Workshops, Shanghai, China, 18 August 2014. [CrossRef]
17. McHugh, O.; Conboy, K.; Lang, M. Agile Practices: The Impact on Trust in Software Project Teams. *IEEE Softw.* **2012**, *29*, 71–76. [CrossRef]
18. Paasivaara, M.; Durasiewicz, S.; Lassenius, C. Distributed Agile Development: Using Scrum in a Large Project. In Proceedings of the 2008 IEEE International Conference on Global Software Engineering, Bangalore, India, 17–20 August 2008. [CrossRef]
19. Lee, S.; Yong, H.-S. Distributed Agile: Project management in a global environment. *Empir. Softw. Eng.* **2009**, *15*, 204–217. [CrossRef]
20. Jalali, S.; Wohlin, C. Agile Practices in Global Software Engineering—A Systematic Map. In Proceedings of the 2010 5th IEEE International Conference on Global Software Engineering, Princeton, NJ, USA, 23–26 August 2010. [CrossRef]
21. Yap, M. Follow the sun: Distributed extreme programming development. In Proceedings of the AGILE Conference, Denver, CO, USA, 24–9 July 2005.
22. Manifesto, A. Manifesto for Agile Software Development. Available online: <http://agilemanifesto.org/> (accessed on 16 December 2022).
23. Jain, N. Offshore Agile Maintenance. In Proceedings of the AGILE 2006 (AGILE'06), Minneapolis, MN, USA, 23–28 July 2006. Available online: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1667596> (accessed on 16 December 2022).
24. Svensson, H.; Host, M. Introducing an agile process in a software maintenance and evolution organization. In Proceedings of the Ninth European Conference on Software Maintenance and Reengineering, Manchester, UK, 21–23 March 2005.
25. Choudhari, J.; Suman, U. Extended iterative maintenance life cycle using eXtreme programming. *ACM SIGSOFT Softw. Eng. Notes* **2014**, *39*, 1–12. [CrossRef]
26. Rudzki, J.; Hammouda, I.; Mikkola, T. Agile Experiences in a Software Service Company. In Proceedings of the 2009 35th Euromicro Conference on Software Engineering and Advanced Applications, Patras, Greece, 27–29 August 2009. [CrossRef]
27. Pino, F.J.; Ruiz, F.; Garcia, F.; Piattini, M. A Software Maintenance Methodology for Small Organizations: Agile_MANTEMA. *J. Softw. Evol. Process* **2011**, *24*, 851–876. [CrossRef]
28. Kumar, B. The Sway of Agile Processes over Software Maintainability. *Int. J. Comput. Appl.* **2015**, *109*, 25–29. [CrossRef]
29. Highsmith, J.; Cockburn, A. Agile Software Development: The Business of Innovation. *Computer* **2001**, *34*, 120–127. [CrossRef]
30. Prochazka, J. Agile Support and Maintenance of IT Services. *Inf. Syst. Dev.* **2011**, *1*, 597–609. [CrossRef]
31. Kaur, P.; Sharma, S. Agile Software Development in Global Software Engineering. *Int. J. Comput. Appl.* **2014**, *97*, 39–43. [CrossRef]
32. Leedy, P.D.; Ormrod, J.E. *Practical Research. Planning and Design*, 11th ed.; Pearson: Boston, MA, USA, 2018; Volume 1.
33. Kitchenham, B.A.; Pfleeger, S.L. Personal Opinion Surveys. In *Guide to Advanced Empirical Software Engineering*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2008; pp. 63–92. [CrossRef]
34. Younas, M.; Jawawi, D.N.A.; Ghani, I.; Shah, M.A.; Khurshid, M.M.; Madni, S.H.H. Framework for Agile Development Using Cloud Computing: A Survey. *Arab. J. Sci. Eng.* **2019**, *44*, 8989–9005. [CrossRef]
35. Talluri, M.; Haddad, H.M. Best Managerial Practices in Agile Development. In Proceedings of the 2014 ACM Southeast Regional Conference, Kennesaw, Georgia, 28–29 March 2014. [CrossRef]
36. Awar, K.B.; Sameem, M.S.I.; Hafeez, Y. A Model for Applying Agile Practices in Distributed Environment: A Case of Local Software Industry. In Proceedings of the 2017 International Conference on Communication, Computing and Digital Systems (C-CODE), Islamabad, Pakistan, 8–9 March 2017. [CrossRef]
37. Kircher, M.; Jain, P.; Corsaro, A.; Levine, D. Distributed Extreme Programming. In Proceedings of the XP2001—eXtreme Programming and Flexible Processes in Software Engineering, Villasimius, Italy, 21–23 May 2001.
38. Burgess, T.F. A general introduction to the design of questionnaires for survey research. *Guide Des. Quest.* **2001**, *30*, 411–432.

39. Gopularam, B.P.; Yogeesh, C.B.; Periasamy, P. Highly Scalable Model for Tests Execution in Cloud Environments. In Proceedings of the 2012 18th International Conference on Advanced Computing and Communications (ADCOM), Bangalore, India, 14–16 December 2012. Available online: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6563584> (accessed on 14 March 2023).
40. Selvanathan, M.; Jayabalan, N.; Saini, G.K.; Supramaniam, M.; Hussin, N. Employee Productivity in Malaysian Private Higher Educational Institutions. *PalArch's J. Archaeol. Egypt/Egyptol.* **2020**, *17*, 66–79.
41. Kaiser, H.F. An Index of Factorial Simplicity. *Psychometrika* **1974**, *39*, 31–36. Available online: <https://link.springer.com/article/10.1007/BF02291575> (accessed on 16 December 2022). [CrossRef]
42. Beckett, C.; Eriksson, L.; Johansson, E.; Wikström, C. Multivariate Data Analysis (MVDA). In *Pharmaceutical Quality by Design: A Practical Approach*; Wiley & Sons: Hoboken, NJ, USA, 2017.
43. Pimentel, J. A Note on the Usage of Likert Scaling for Research Data Analysis Some of the Authors of This Publication Are Also Working on These Related Projects: Collaboration Project View Project Personal Project View Project. *USM R D* **2010**, *18*, 109–112.
44. Salkind, N.J. *Statistics for People Who (Think They) Hate Statistics*; Sage: London, UK, 2000.
45. Yamato, Y.; Katsuragi, S.; Nagao, S.; Miura, N. Software maintenance evaluation of agile software development method based on open stack. *IEICE Trans. Inf. Syst.* **2015**, *98*, 1377–1380. [CrossRef]
46. Tuli, A.; Hasteer, N.; Sharma, M.; Bansal, A. Empirical Investigation of Agile Software Development. *ACM SIGSOFT Softw. Eng. Notes* **2014**, *39*, 1–6. [CrossRef]
47. Nazir, A.; Raana, A.; Khan, M.F. Cloud Computing ensembles Agile Development Methodologies for Successful Project Development. *Int. J. Mod. Educ. Comput. Sci.* **2013**, *11*, 5. [CrossRef]
48. Braithwaite, K.; Joyce, T. XP Expanded: Distributed Extreme Programming. In Proceedings of the Extreme Programming and Agile Processes in Software Engineering, Sheffield, UK, 18–23 June 2005; pp. 180–188. [CrossRef]
49. Almudarra, F.; Qureshi, B. Issues in Adopting Agile Development Principles for Mobile Cloud Computing Applications. *Procedia Comput. Sci.* **2015**, *52*, 1133–1140. [CrossRef]
50. Berczuk, S. Back to Basics: The Role of Agile Principles in Success with an Distributed Scrum Team. In Proceedings of the Agile 2007 (AGILE 2007), Washington, DC, USA, 13–17 August 2007. [CrossRef]
51. Simons, M. Internationally Agile. The Challenges of Offshore Development. InformIT. Available online: <https://www.informit.com/articles/article.aspx?p=25929> (accessed on 15 March 2002).
52. Ramesh, B.; Cao, L.; Mohan, K.; Xu, P. Can Distributed Software Development Be Agile? *Commun. ACM* **2006**, *49*, 41. [CrossRef]
53. Farmer, M. DecisionSpace Infrastructure: Agile Development in a Large, Distributed Team. In Proceedings of the Agile Development Conference, Salt Lake City, Salt Lake City, UT, USA, 22–26 June 2004. Available online: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1359801> (accessed on 4 October 2022).
54. Therrien, E. Overcoming the Challenges of Building a Distributed Agile Organization. In Proceedings of the Agile 2008 Conference, Toronto, ON, Canada, 4–8 August 2008. Available online: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4599507> (accessed on 16 December 2022).
55. Mahmood, S.; Anwer, S.; Niazi, M.; Alshayeb, M.; Richardson, I. Key factors that influence task allocation in global software development. *Inf. Softw. Technol.* **2017**, *91*, 102–122. [CrossRef]
56. Tiwari, N. Using the analytic hierarchy process (AHP) to identify performance scenarios for enterprise application. *Comput. Meas. Group Meas. It* **2006**, *4*.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.