IJIST
P ISSN : 2618-1630
E ISSN : 2909-6130

Role of Agile Methodologies for Ensuring Quality in Complex Systems: A Systematic Literature Review

Ihsan Ullah[1], Jaweria Kanwal[1], Fatima Gillani[1], Iqra Shahzad[1]

[1]National University of Modern Languages

* **Correspondence:** Jaweria Kanwal, jaweria.kanwal@numl.edu.pk

Citation |Ullah. I, Kanwal. J, Gillani. F, Shahzad. I, "Role of Agile Methodologies for Ensuring Quality in Complex Systems: A Systematic Literature Review," Int. J. Innov. Sci. Technol., vol. 4, no. 4, pp.1042-1054, 2022.

DOI | https://doi.org/10.33411/IJIST/2022040409

Received | Aug 26, 2022. **Revised** | Oct, 22,2022. **Accepted** | Oct 27, 2022. **Published** | Nov 05, 2022.

In software development, the selection of a software process model set the base for the success of a software product. An inappropriate selection may lead to a delay in project release, introduce defects and make the project difficult to update. This lack of quality characteristics may lead to the risk of losing customer expectations as well as the failure of the project itself. In the case of complex systems, the problems become more severe. To meet such expectations, agile methodologies are used to ensure quality in software and meet customers' expectations. There is currently no literature that gives insights into the role of agile methodologies in ensuring quality in complex systems. The purpose of this paper is to evaluate the effectiveness and the impact of agile methodology in achieving the quality of complex systems.

For this, we perform a Systematic Literature Review (SLR) and define a review protocol. By performing a thorough search and screening, we selected 39 papers related to agile methods and complex systems. Our analysis shows that complex systems have various requirements of quality attributes some of the complex systems mainly focus on security, reliability, and efficiency whereas other emphasizes safety, response time, and maintainability. Our analysis also shows that agile methodologies are widely used for the development of complex systems because ensuring the quality requirements of complex systems is not possible with the use of traditional methods of software development.

Keywords: Agile methodology; software quality; software quality attributes; complex systems; complex systems quality

Acknowledgment.

This research work was supported by the National University of Modern Languages, Pakistan.

Project details. Nil

Author's Contribution.

Author1: performed article search and screening, results analysis, and writing. Author 2: supervision, writing. Author 3: article search, writing. Author 4:.article search results in analysis

Conflict of interest:

There exists no conflict of interest for publishing this manuscript in IJIST.

Introduction

Software development is a set of resource-constrained, goal-directed cooperative innovation and communication activities [1]. Agile software methodology, a relatively new approach to software development that was originally introduced in the late 1990s and is now widely considered "mainstream" software engineering throughout the world, has been prevalent in the IT industry for more than 3 decades. Using self-organizing, cross-functional teams and the customer as partners, agile approaches are a collection of software development techniques centered on incremental and iterative development [2]. Agile approaches have been shown to be significantly more agile and flexible than conventional software development [3]. It is a time-boxed iterative approach and encourages rapid and flexible responses to changes [4].

In the current world, we are highly dependent on technology in all aspects of our daily life and technology is changing day by day. With the change in environment, the demands of the customer regarding software requirements also change [5] which results in the complexity of the software products. Software systems are required to be interdependent, maintainable, secure, and usable. Complex software systems are challenging because uncertainty is high, requirements are not clear and high efficiency is required. Meeting the quality criteria of complex systems can be much more challenging. Agile methodologies come to the rescue which ensures quality in software products.

Principles and practices of agile methodologies may help cope with the complexity and uncertainty of complex systems and ensure enduring quality in these systems. Different practices of agile development methods ensure various quality characteristics in software systems. Meeting customers' expectations which is the highest priority of agile methodologies, the agile development methods i.e., Scrum, and extreme programming, can be beneficial to maintain the quality of the product.

However, the software development community still knows little about the role of agile in complex software products and how agile practices can achieve quality characteristics in software systems and especially in complex systems. We are motivated to fill in this knowledge gap and embarked on mapping out the published evidence available about agile practices for complex products. For this purpose, we conduct a Systematic Literature Review (SLR) by answering the following research questions.

RQ1. Which quality attributes are required in complex software systems?

RQ2. What agile methods are commonly used for the development of complex software systems?

RQ3. How do agile methodologies ensure product quality in complex systems?

The purpose of this paper is to evaluate the effectiveness and impact of agile methodology in complex software systems. Our main focus is to identify and analyze how various quality characteristics are achieved in complex systems using agile methodologies.

Related Work

In this section, we present relevant literature in the fields of product quality of complex systems using different agile practices. Different studies discussed different methods of agile for achieving a quality product in a complex system. Different problems were found in complex systems and how agile methods are applied to mitigate those problems by their inherited problem-solving nature as compared to the traditional methods. In a study, Setiadi et al. [6] implemented Scrum for online transaction processing in hospitals. The reason for implementing Scrum is because of the frameworks of agile development which can address complex adaptive problems, by increasing the productivity and creativity of all stakeholders [6].

A study conducted by Abdelhamid et al. [7] discussed how product quality is achieved using agile methods in software products by different techniques of requirements verification. Pair programming and test-driven development are the best choices to be used to ensure the quality of the product [7]. Scrum aims to develop high-quality products that can be used to be completed on time for complex systems.

For critical problem domains such as core business systems which require a long-term evolution, XP is used to deliver high-quality software because it facilitates in a changing environment [8]. This study discussed various methods for achieving quality using agile methods by Scrum and extreme programming considering them as a problem-solver for the complex problem domain. Quality assurance is one of the major components of software development, but agile methodologies lack high-quality assurance during the development however iterative and incremental development with the proper testing before the release still produces high-quality software products.

In [9] Fernando Selleri Silva proposed a reference model to improve the quality of software products by keeping quality assurance practices the highest priority for productivity and quality of the product. The model took into account the guidelines provided in CMMI and MPS, BR maturity models. Parita Jain [10] stated that there is a need for a quality model that must ensure that moving from traditional development to agile development leads to the ever-improving quality of the software.

Legowo [11] stated that Agile Software Methodology with Scrum is very precisely implemented in the case of developing a Quality Assurance System. In that study, the choice of Agile development methodology with Scrum was considered very appropriate, specifically in the development process. Finally, the results showed that the Agile Software methodology with Scrum is the right method to increase the efficiency and effectiveness in Quality Assurance System application development

For critical systems, the critical quality factors should be kept in the highest priority due to their critical nature but functional requirements are given more attention than quality requirements which can result in a lack of quality in some of the critical systems, more attention is given to functional requirements than to quality requirements [12]. The reason behind this is to ensure the highest critical qualities are the highest priority for a critical system.

Aggarwal et al. [13] focus on the measures of quality for agile software by highlighting the role of Quality Assurance within the agile development process. Testing as a key component is discussed with a brief understanding that in agile no separate testing team exists. The authors have also provided some metrics for measuring quality in agile as a future scope.

High-quality assurance of software is somehow compromised for agility but iterative and incremental combined development still produced high-quality software. Proper Testing is performed before release to make sure the product meets its quality or not [14].

Material And Methods

This study was carried out according to the standards set forth by Kitchenham and Charters [15][16] for conducting a Systematic Literature Review (SLR). To conduct SLR, review we formulate research questions, define and validate review protocols, search and selection of research papers performed, and screening of papers conducted through inclusion/exclusion criteria [17][18][19]. Our research methodology involves different activities such as the selection of processes, search resources, search keyword selection, and data extraction methods. The detail of the workflow is depicted in Figure 1. Detail description of our methodology is as follows.

a) Search Process

The section includes the processes for searching for the literature, discussing the search resources and search keywords, inclusion and exclusion criteria data extraction strategy, and data synthesis method.

b) Search Resources

The search included electronic sources, and the main focus for the search string was Agile and complex software products both combined keywords, and those keywords are modified according to our specific requirements for search in digital libraries, the digital library which was chosen for conducting search strategy was: The IEEExplore Digital library, The ACM Library, and Science Direct [20].

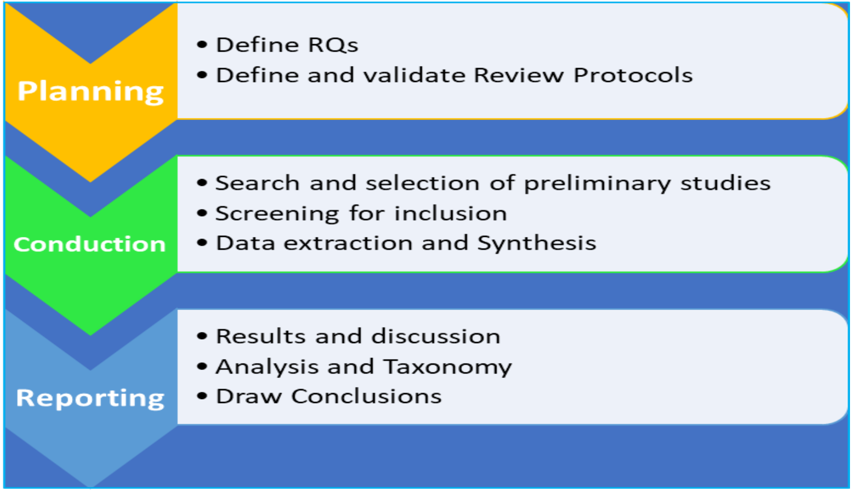
Planning
• Define RQs
• Define and validate Review Protocols
Conduction
• Search and selection of preliminary studies
• Screening for inclusion
• Data extraction and Synthesis
Reporting
• Results and discussion
• Analysis and Taxonomy
• Draw Conclusions

Figure 1: Overview of SLR stages

c) Search Criteria

The search criteria used for the SLR in this review consist of a string related to keywords such as agile, complex software products, and quality attributes. An example of a search done in the electronic databases is shown below:

"Agile" and "Complex software Product"

Based on the search options provided by each database, we manually created the search string in each one. We approached each database search as a process of learning and experimenting.

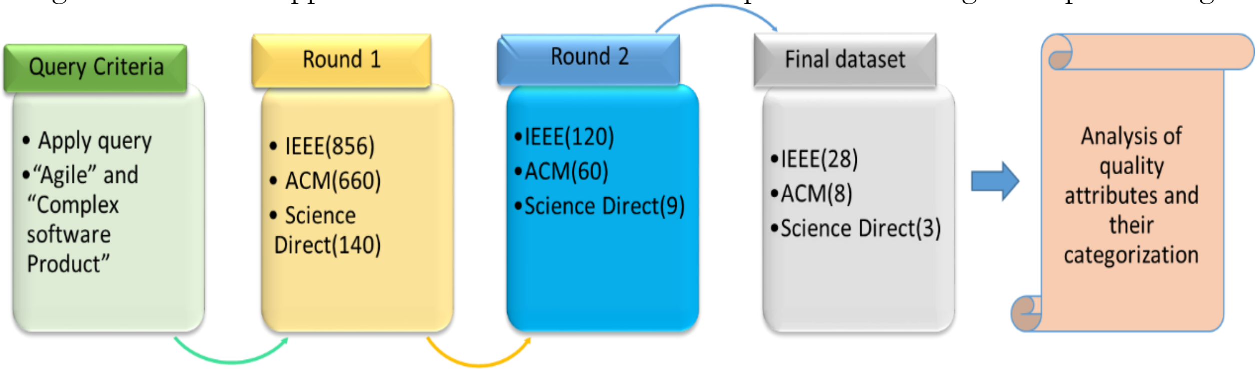

Figure 2: Detail process of paper selection for SLR

d) Study selection criteria

We use the following inclusion and exclusion criteria to assess the papers for SLR. inclusion/ is presented in Figure 2.

i. Inclusion criteria

For inclusion criteria, the paper must have the following features

- ➢ Available on digital libraries (IEE Explore, ACM, Science Direct digital libraries).
- ➢ Focused on the implementation of agile for complex systems.
- ➢ Focused on Scrum or Extreme Programming or Agile processes for critical systems.
- ➢ Do not be a prior systematic literature review paper.
- ➢ Published between the years 2008 and 2021.
- ➢ Written in the English Language.
- ➢ Non-redundant since the main focus is to study Agile methods and their applications in complex systems, and complex software product quality.

ii. Exclusion criteria

Any paper not having any of the inclusion criteria was excluded after reading the remaining papers carefully. A list of papers that were included and excluded was preserved. Multiple publications of comparable research were excluded to minimize results bias.

The SLR protocol was evaluated for internal consistency confirmation, the research questions are derived from the search strings, the data extraction forms allow for answering the research question(s), and the data analysis procedure is adequate to answer the research questions.

Table 1: Summary of included papers

No.	Digital Database	Retrieved	The final number of studies
1	IEEE	120	28
2	ACM	60	8
3	Science Direct	9	3

e) Final selection

To guarantee that the publications addressed our research questions, the following screening criteria were employed. 1) and agile practices for complex systems? 2) Does a paper discuss any quality attributes of complex systems?

The following two criteria were included relating to the quality of each paper's description for extra quality evaluation. 3) Does the paper's goal appear to be stated clearly? 4) Does the article appropriately address the background of the Agile methods and complex systems?39 papers were selected by carrying out the quality assessment based on those quality assessment criteria. The summary of those selected papers is given in Table 1.

Table 2:Quality attributes found in the studies

No.	Quality Attributes	Studies
1	Correctness	[5][7][10][12][14][16][17] [18]–[20][21]
2	Consistency	[5][7][10][12][14] [16] [17] [18]–[20] [22]
3	Security	[10][12][14] [16] [17][18], [19][23]
4	Accuracy	[7][10][12][14] [16] [17][18], [19][21]
5	Reliability	[5][7][10][12][14] [16] [17] [18]–[20] [24], [25] [21][26]–[28] [22]
6	Response time	[10][12][14] [16] [17] [18], [19][21] [27]
7	Maintainability	[5][10][12][14] [16][18]–[20] [21] [26], [27] [22]
8	Stability	[18] [20] [21]
9	Interoperability	[10][12][14] [16] [18]–[20] [24] [21][26], [27]
10	Data integrity, Bug-free at runtime, completeness, safety	[18] [20] [25][26]

f) Data extraction

Data extraction was carried out on the 39 studies that satisfied the inclusion and exclusion criteria the data of each study were recorded in a textual file. The recorded files included agile methodologies implementation and the complex systems quality attributes included in any of those studies. The data extraction process faced some difficulties because the quality achievements were not reported in certain studies. Due to this fact, all data from the studies were extracted by both authors. All disagreements were resolved by discussion.

g) Synthesis of findings

The study followed the synthesis method to combine the results of the studies. The most common method which was used is the Meta-ethnography approach, by combining the results of different studies to create an understanding of phenomena under study greater than the individual ethnographic studies [21], in the first stage of following this approach, the results of those studies were collected in the tables for the agile methodologies used and the product quality attributes of complex systems, in the second phase the findings were compared, the study revealed different approaches used different agile methodologies and different quality attributes of the complex systems which are mentioned in the result section. Finally, the findings were translated, compared, and synthesized to meet our research questions. The synthesis of the studies is given in Table 2 for the quality attributes and in Table 3 for the Agile Method used in the studies.

Result And Discussion

This research involves SLR in the fields of product quality of complex systems using different agile practices. Different studies discussed different methods of agile for achieving a quality product in a complex system. This section presents the results of the SLR by answering the following research questions.

RQ1: Which quality attributes are required in complex software systems?

In the selected studies, various quality attributes were re-discussed according to the complexity of the systems and their requirements in the domain they are using, but most of the major quality attributes which were mentioned in those papers are shown in Figure 3.

Different papers discussed different quality attributes according to their systems and implementations in that area but the major and most commonly discussed quality attributes from the studies are correctness consistency, security, extensibility/responsiveness to change, response time, maintenance, reliability, accuracy, and performance.

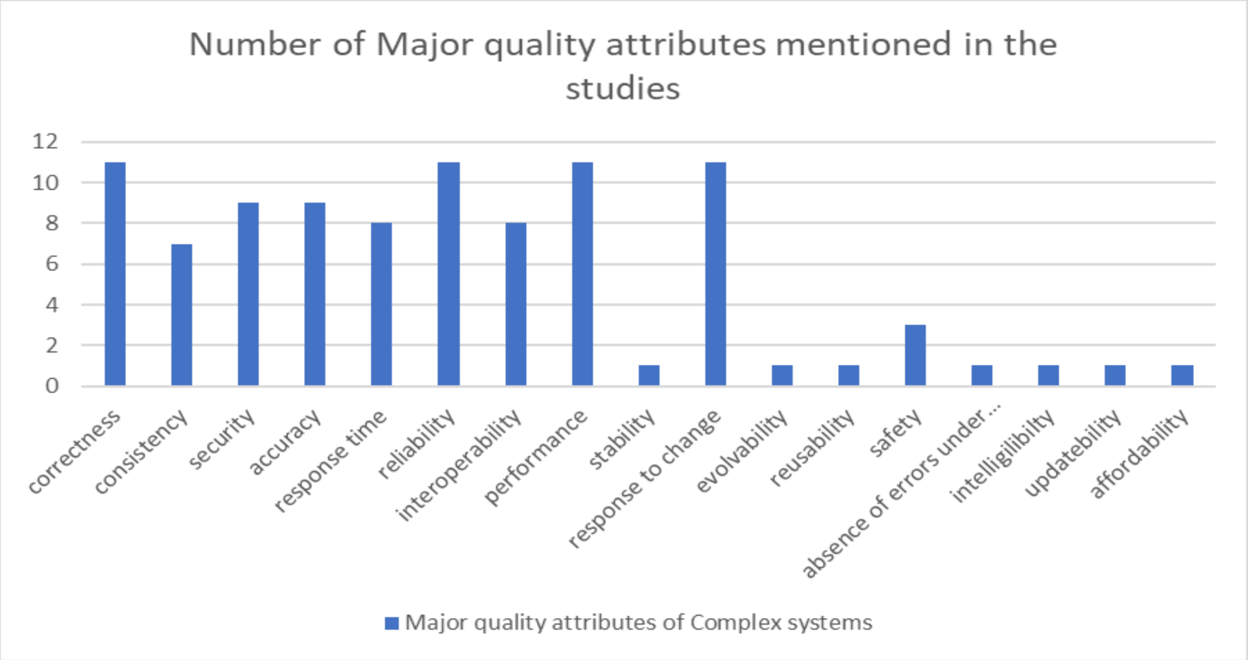

Figure 3: Number of quality attributes in complex systems mentioned in studies

The identified quality attributes can be categorized into four categories to understand the type of quality characteristics required in complex systems. We categorized them into 4 main categories i.e. SAFETY, PERFORMANCE, CHANGEABILITY, and RELIABILITY as shown in Figure 4. From this, the most important and frequently mentioned Quality attributes identified are in category B. So attributes in Class B will be of high priority in complex system applications practicing agile approaches as shown in Figure 3.

Category A describes the security and safety characteristics of software. The security of complex systems is one of the highest quality factors, especially in the domain of hospital management and banking systems. Therefore it must be as secure as possible. Category B represents quality attributes related to the performance of a software system. The complex software system should be accurate (i.e. having not a single error in any value), have better response time (should be "hard" in real time which means it should respond in that specific time without any delay), and correct (verification of specifications of what the software will do, and do correctly in a safe manner).

Category C represents quality attributes related to software changeability [22][23][24]. It represents that software should be responsive to change (should be easy to upgrade and its functionalities should be easily extended without taking too much down or losing data), maintainable (should be updateable easily) [25][26][29]. Category D represents quality attributes related to the reliability of a software system. Complex software systems should be consistent (should be consistent with their results/outputs), reliable (should be reliable to be used), and stable (should be operable in various environments).

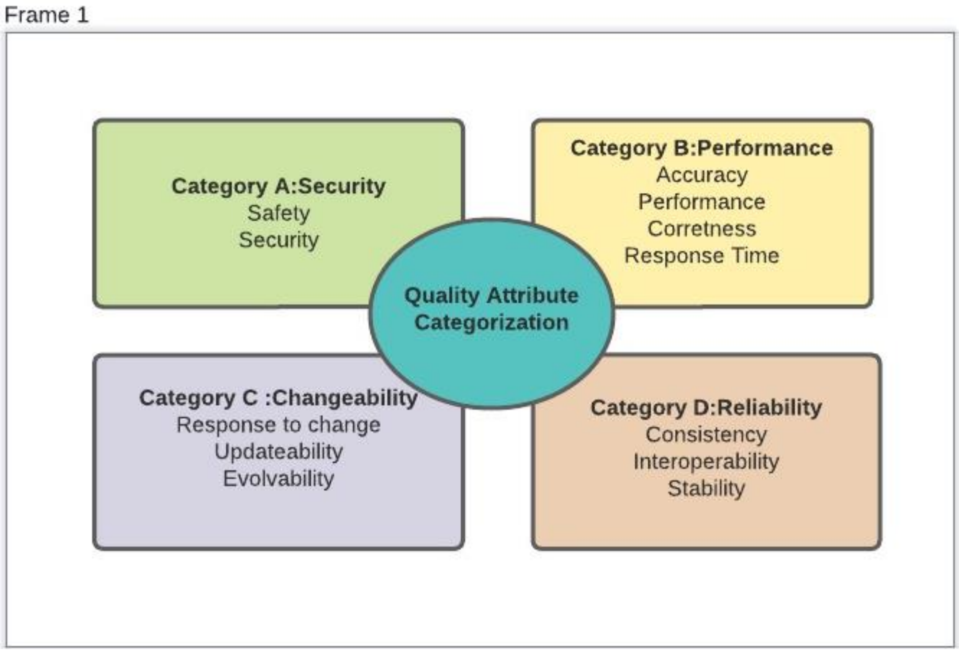

Figure 4: Categorization of quality attributes in complex software systems

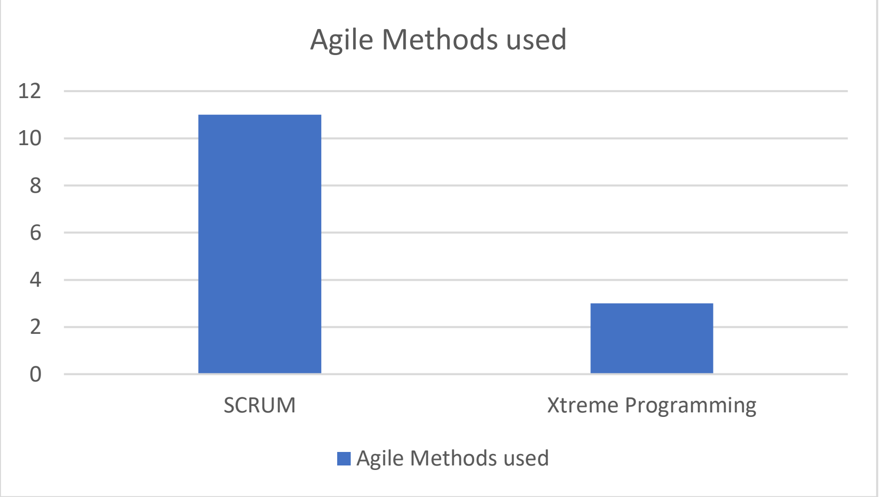

Figure 5: Agile methods used in complex systems in mentioned studies

RQ2: What agile methods are commonly used for the development of complex software systems?

To achieve quality characteristics in complex software systems, agile methodologies selected by the software companies/process managers will help in suggesting appropriate agile methods for achieving quality characteristics in new complex systems. Table 4 shows the studies in which agile methodologies were used for the development of complex systems.

There are some research studies where Scrum is combined with a variety of other frameworks to get the benefits of agile methods [14]. In some studies, Scrum and XP are combined to get the full benefits of the principles and values of agile methods which proved to be productive for their complex systems [9].

Different studies use agile for achieving various quality attributes in a complex system. The most commonly used agile methods in our selected studies are Scrum and Extreme Programming for complex systems because these two methodologies are problem-solving as compared to the traditional methods [30]–[32]. In a study, a case study for an online transaction processing system is conducted where Scrum is used as a process model. Analysis of the study revealed that the reason for using Scrum is to achieve high-quality attributes like usability, reliability, and accuracy that are required for an online transaction processing system. Practices of Agile like implementing different techniques for requirements verification, pair programming, and Test-Driven Development ensure quality in the software.

In another study, extreme programming (XP) is applied to a complex problem domain. Analysis of the case study shows that XP addresses complex adaptive problems through its values, principles, and practices. It also increases the productivity and creativity of all stakeholders. Furthermore, its incremental approach allows product quality to improve with each deliverable. For critical problem domains such as core business systems that benefit from a long-term evolutionary approach, XP can be used to deliver high-quality software that keeps up with a changing environment [26]. The reason behind their success and popularity is their ability to deal with changing requirements and adopting to change [33].

Table 3: Agile methods used for complex systems

No.	Agile Method used	Studies
1	Scrum	[6] [12] [34] [35] [36][13][37][38][33] [32]
2	XP	[7][8] [39][40][28]
3	XP and Scrum along CMM	[9]
4	Scrum along RUP (Rational Unified Process)	[14]

RQ3: How do the agile methodologies ensure product quality in complex systems?

As we know that agile methodologies focus on customer satisfaction, adapting to change, and customer feedback, in this RQ, we will analyze how these practices result in improvement in the quality of products, especially in complex systems. As two methodologies i.e. extreme programming and Scrum are mainly used in the development of complex systems, we will analyze how these practices ensure quality in complex systems.

Our analysis shows that agile approaches use combinations of best practices to incorporate quality into the product, resulting in a new viewpoint on quality management. Many studies have supported and promoted the benefits of agile approaches in terms of quality [27], [31][34].

Agile methodologies overcome the conventional development methods' limitations and provide flexibility to adopt the changes in requirements at any stage [28]. Applying values from the principle of the agile method can enhance the success of complex systems implementation and minimizes the risk of failure. Due to its ability to welcome change, the principles of agile play a vital role in dealing with the complexity and uncertainty of large-scale projects [35]. Practices of extreme programming e.g. test-driven development and pair programming play a major role to ensure high quality in complex systems and reducing bugs [36].

From a development perspective, Agile software development offers the development team a lot of benefits of management and responsiveness in problem discovery and solving [37][38]. The composition of a team in agile practices has a great relationship with the practices available in high-performing teams and organizations. These practices provide a great development environment which results in a good quality product [39]. Other major benefits of agile methods include sprints in XP and Scrum, better communication practices, test-driven development, and customer feedback[40]. All these principles and the values in Scrum and XP ensure major quality attributes in complex systems.

Conclusion.

In this paper, we aim to provide detailed insight into applying agile methodologies to complex software systems. So, to determine the influence of agile methodology and frequently used quality attributes, we carried out a thorough review of the literature and identified the major quality attributes of complex software systems, the agile methodologies implementation in the complex systems, and analyzed how the agile methodologies meet those quality attributes of complex systems. Our analysis shows that complex systems have complex interdependencies of quality attributes. Some of the complex systems mainly focus on security, reliability, and efficiency whereas other emphasizes safety, response time, and maintainability which are for different domains. Various domains define quality attributes according to their applications. Most complex systems share common quality attributes i.e. reliability, correctness, security, response time, and performance. Our study shows that Scrum and extreme programming are the common agile methodologies used for developing complex software systems. The reason behind the use of the method is the features, principles, and values in both methodologies which are the key success factors to ensuring the quality attributes in complex systems.

In the future, we'll provide guidelines to suggest to developers which agile approach is more effective for complex systems in the view of experts.

References

[1] A. Cockburn, "The end of software engineering and the start of economic-cooperative gaming," Comput. Sci. Inf. Syst., vol. 1, no. 1, pp. 1–32, 2004, doi: 10.2298/csis0401001c.

[2] M. Cohn and D. Ford, "Introducing an agile process to an organization," Computer (Long. Beach. Calif)., vol. 36, no. 6, pp. 74–78, 2003, doi: 10.1109/MC.2003.1204378.

[3] A. Qumer and B. Henderson-Sellers, "A framework to support the evaluation, adoption and improvement of agile methods in practice," J. Syst. Softw., vol. 81, no. 11, pp. 1899–1919, Nov. 2008, doi: 10.1016/J.JSS.2007.12.806.

[4] J. A. Livermore, "Factors that impact implementing an agile software development methodology," Conf. Proc. - IEEE SOUTHEASTCON, pp. 82–86, 2007, doi: 10.1109/SECON.2007.342860.

[5] J. Kanwal, O. Maqbool, H. A. Basit, and M. A. Sindhu, "Evolutionary Perspective of Structural Clones in Software," IEEE Access, vol. 7, no. January, pp. 58720–58739, 2019, doi: 10.1109/ACCESS.2019.2913043.

[6] T. Setiadi and S. B. Premapasha, "Scrum implementation for online transaction processing

(oltp) in hospital management," Proceeding 2018 12th Int. Conf. Telecommun. Syst. Serv. Appl. TSSA 2018, Jul. 2018, doi: 10.1109/TSSA.2018.8708797.

[7] S. Abdalhamid, A. O. M. Mohammed, and A. Mishra, "Agile and quality: A systematic mapping study," Proc. 2019 Int. Conf. Comput. Sci. Renew. Energies, ICCSRE 2019, Jul. 2019, doi: 10.1109/ICCSRE.2019.8807763.

[8] R. S. Wingo and M. M. Tanik, "Using an agile software development methodology for a complex problem domain," Conf. Proc. - IEEE SOUTHEASTCON, vol. 2015-June, no. June, Jun. 2015, doi: 10.1109/SECON.2015.7132890.

[9] F. S. Silva, F. S. F. Soares, A. L. Peres, I. M. De Azevedo, P. P. Pinto, and S. R. De Lemos Meira, "A reference model for agile quality assurance: Combining agile methodologies and maturity models," Proc. - 2014 9th Int. Conf. Qual. Inf. Commun. Technol. QUATIC 2014, pp. 139–144, Dec. 2014, doi: 10.1109/QUATIC.2014.25.

[10] A. L. and S. A. Jain P., "Current state of the research in agile quality development | IEEE Conference Publication | IEEE Xplore," 3rd International Conference on Computing for Sustainable Global Development, 2016.

[11] M. B. Legowo, B. Indiarto, and D. Prayitno, "Agile Software Methodology with Scrum for Developing Quality Assurance System," Proc. - 2019 2nd Int. Conf. Comput. Informatics Eng. Artif. Intell. Roles Ind. Revolut. 4.0, IC2IE 2019, pp. 104–109, Sep. 2019, doi: 10.1109/IC2IE47452.2019.8940831.

[12] A. I. M. Leite, "An Approach to Support the Specification of Agile Artifacts in the Development of Safety-Critical Systems," Proc. - 2017 IEEE 25th Int. Requir. Eng. Conf. RE 2017, pp. 526–531, Sep. 2017, doi: 10.1109/RE.2017.43.

[13] A. Agarwal, N. K. Garg, and A. Jain, "Quality assurance for product development using Agile," ICROIT 2014 - Proc. 2014 Int. Conf. Reliab. Optim. Inf. Technol., pp. 44–47, 2014, doi: 10.1109/ICROIT.2014.6798281.

[14] M. Tanveer, "Agile for large scale projects - A hybrid approach," 2015 Natl. Softw. Eng. Conf. NSEC 2015, pp. 14–18, Feb. 2016, doi: 10.1109/NSEC.2015.7396338.

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

[16] N. Yildirim and A. Varol, "A research on security vulnerabilities in online and mobile banking systems," 7th Int. Symp. Digit. Forensics Secur. ISDFS 2019, pp. 1–5, 2019, doi:

10.1109/ISDFS.2019.8757495.

[17] L. Lopez, J. Partanen, P. Rodriguez, and S. Martinez-Fernandez, "How practitioners manage quality requirements in rapid software development: A survey," Proc. - 2018 1st Int. Work. Qual. Requir. Agil. Proj. QuaRAP 2018, pp. 14–17, 2018, doi: 10.1109/QuaRAP.2018.00009.

[18] D. Schneider and M. Trapp, "Runtime safety models in open systems of systems," 8th IEEE Int. Symp. Dependable, Auton. Secur. Comput. DASC 2009, pp. 455–460, 2009, doi: 10.1109/DASC.2009.111.

[19] P. Subsorn and S. Limwiriyakul, "A case study of internet banking security of mainland Chinese banks: A customer perspective," Proc. - 2012 4th Int. Conf. Comput. Intell. Commun. Syst. Networks, CICSyN 2012, pp. 189–195, 2012, doi: 10.1109/CICSyN.2012.43.

[20] H. M. Chen and R. Kazman, "Architecting ultra-large-scale green information systems," 2012 1st Int. Work. Green Sustain. Software, GREENS 2012 - Proc., pp. 69–75, 2012, doi: 10.1109/GREENS.2012.6224259.

[21] G. W. Noblit and R. D. Hare, "Meta-ethnography : synthesizing qualitative studies," p. 88, 1988.

[22] J. Kanwal, H. A. Basit, and O. Maqbool, "Structural clones: An evolution perspective," 2018 IEEE 12th Int. Work. Softw. Clones, IWSC 2018 - Proc., vol. 2018-January, pp. 9–15, Mar. 2018, doi: 10.1109/IWSC.2018.8327313.

[23] C. Kaufmann, A. Kock, and H. G. Gemünden, "Emerging strategy recognition in agile portfolios," Int. J. Proj. Manag., vol. 38, no. 7, pp. 429–440, Oct. 2020, doi: 10.1016/J.IJPROMAN.2020.01.002.

[24] O. Michael, L. T. Lun, and K. E. Keong, "Methodology to improve Safety Critical SoC based platform: A case study," IEEE Int. Reliab. Phys. Symp. Proc., vol. 2021-March, pp. 12–15, 2021, doi: 10.1109/IRPS46558.2021.9405223.

[25] E. P. Jharko, "The methodology of software quality assurance for safety-critical systems," 2015 Int. Sib. Conf. Control Commun. SIBCON 2015 - Proc., 2015, doi: 10.1109/SIBCON.2015.7147057.

[26] M. Younas, D. N. A. Jawawi, I. Ghani, and R. Kazmi, "Non-Functional Requirements Elicitation Guideline for Agile Methods," J. Telecommun. Electron. Comput. Eng., vol. 9, no. 3-4 Special Issue, pp. 137–142, 2017.

[27] M. N. Aydin, F. Harmsen, K. Van Slooten, and R. A. Stegwee, "An agile information systems development method in use," Turkish J. Electr. Eng. Comput. Sci., vol. 12, no. 2, pp. 127–138, 2004.

[28] S. Al-Saqqa, S. Sawalha, and H. Abdelnabi, “Agile software development: Methodologies and trends,” Int. J. Interact. Mob. Technol., vol. 14, no. 11, pp. 246–270, 2020, doi: 10.3991/ijim.v14i11.13269.

[29] J. Kanwal, “Refactoring Patterns Study in Code Clones during Software Evolution,” 2017 IEEE 11th Int. Work. Softw. Clones, no. July 2019, pp. 1–2, 2017, doi: 10.1109/IWSC.2017.7880508.

[30] L. Williams, “A Survey of Agile Development Methodologies,” Univercsty West Engl., pp. 209–227, 2007.

[31] B. Henderson-Sellers and M. K. Serour, “Creating a Dual-Agility Method: The Value of Method Engineering,” https://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/jdm.2005100101, vol. 16, no. 4, pp. 1–24, Jan. 1AD, doi: 10.4018/JDM.2005100101.

[32] A. Taufiq, T. Raharjo, and A. Wahbi, “Scrum evaluation to increase software development project success: A case study of digital banking company,” 2020 Int. Conf. Adv. Comput. Sci. Inf. Syst. ICACSIS 2020, pp. 241–246, Oct. 2020, doi: 10.1109/ICACSIS51025.2020.9263235.

[33] J. of C. S. IJCSIS, “Cost Efficient Scrum Process Methodology to Improve Agile Software Development,” IJCSIS Vol 18 No. 4 April 2020 Issue, Jan. 2020.

[34] A. Braz, C. M. F. Rubira, and M. Vieira, “Development of Complex Software with Agile Method,” Proc. - 2015 Agil. Conf. Agil. 2015, pp. 97–101, 2015, doi: 10.1109/Agile.2015.18.

[35] R. Tommy, M. Mhaisekar, S. Kallepally, L. Varghese, S. Ahmed, and M. D. Somaraju, “Dynamic quality control in agile methodology for improving the quality,” 2015 IEEE Int. Conf. Comput. Graph. Vis. Inf. Secur. CGVIS 2015, pp. 233–236, 2016, doi: 10.1109/CGVIS.2015.7449927.

[36] S. M. Faizi, S. Rahman, and K. Hopkins, “Implementing Large Enterprise Resource Planning Systems with Agile Methods,” ICIET 2019 - 2nd Int. Conf. Innov. Eng. Technol., pp. 23–24, 2019, doi: 10.1109/ICIET48527.2019.9290587.

[37] T. Mcbride and M. Lepmets, “Quality assurance in agile safety-critical systems development,” Proc. - 2016 10th Int. Conf. Qual. Inf. Commun. Technol. QUATIC 2016, pp. 44–51, Jan. 2017, doi: 10.1109/QUATIC.2016.016.

[38] D. Oxenham, “Agile approaches to meet complex system of system engineering challenges: A defence perspective,” 2010 5th Int. Conf. Syst. Syst. Eng. SoSE 2010, 2010, doi: 10.1109/SYSOSE.2010.5544092.

[39] H. Zan, L. Pan, and Y. Wang, "Design & implementation of laboratory information management system based on agile method," 2012 2nd Int. Conf. Consum. Electron. Commun. Networks, CECNet 2012 - Proc., pp. 2490–2493, 2012, doi: 10.1109/CECNet.2012.6201497.

[40] M. Younas et al., "Elicitation of Nonfunctional Requirements in Agile Development Using Cloud Computing Environment," IEEE Access, vol. 8, pp. 209153–209162, 2020, doi: 10.1109/ACCESS.2020.3014381.