Impact of Agile-Hybrid Methods and Approaches on Software Quality: A Systematic Literature Review

Žana Zekić, Zlatko Stapić
Faculty of Organization and Informatics
University of Zagreb
Pavlinska 2, 42000 Varaždin, Croatia
{zzekic, zstapic}@foi.hr

Abstract. *The paper presents results of performing a systematic literature review process aiming to answer what combinations of agile methods and practices are used in hybrid approaches with reported impact on software quality and how the use of agile and hybrid approaches impacts the software quality. The review included more than 3900 original scientific studies published in the last five years and it yielded 15 papers dealing with this topic. The total of seven combinations of agile methods are reported to have an impact on software quality as well as 22 agile practices impacting on 26 software quality attributes.*

Keywords. Agile-Hybrid Methods, Software Quality, Systematic literature review, SLR, Software development

1 Introduction

Today, many organizations declare themselves as agile, but it is difficult to find pure agile development in practice. Many of them use a hybrid software development approach, which combines agile and/or traditional methods. Agile development puts emphasis on quality over quantity, but the practice has shown that it is difficult to achieve a high level of product quality in a dynamic environment.

In order to identify the factors (agile practices and attributes) having the most significant influence on mentioned quality, the aim of this research is to find scientific studies that explore various combinations of agile methods and their impact on software quality.

In order to achieve that, systematic literature review will be performed. However, we know that there are many systematic literature reviews (SLRs) that address the topic of agile development and software quality. Research from Hoda et al. (2017) showed that SLRs that cover the topic of agile software development are on the rise. They cover different aspects of agile development. Studies represent the benefits and challenges of agile and hybrid methods. On the other hand, Alsaqaf et al. (2017) in their study focus on identifying the challenges in the engineering of quality requirements in agile projects. They identified 12 challenges that harm the quality and proposed 13 solutions for dealing with quality requirements in agile projects. Similarly, Fitriani et al. (2016) conducted systematic literature review (from now on: SLR) about challenges in agile software development. Their study identified 30 challenges in agile software development and showed that these challenges significantly affect quality. Furthermore, Thangiah and Basri (2016) as well as Arcos-Medina and Mauricio (2019) analysed additional factors influencing software quality: testing and agile development process. Thangiah and Basri identified that different testing techniques in agile development can have significant influence on software quality, while Arcos-Medina and Mauricio performed SLR presenting aspects of software quality applied to the process of agile software development, and their work identified 118 critical success factors, 137 agile practices, 102 metrics, and 70 quality models.

However, none of these studies were exploring the factors influencing the software quality in development context in which the agile-hybrid approaches are used. This makes the rationale for performing our study as necessary and relevant.

This paper presents the results of systematic literature review on the above-mentioned topic and is organized as follows: research methodology stating research questions is presented in the second section, third section provides research results naming identified agile-hybrid methods and the impact of agile practices on quality attributes, which are then analysed and discussed in section four, and finally, section six represents the conclusion of the study.

2 Research methodology

The systematic literature review has been selected as a research method to analyse the impact of agile-hybrid approaches on software quality. SLR is a method that helps identifying, evaluating and interpreting available research relevant to a stated research question.

According to Kitchenham and Charters (2007), SLR has three main phases: planning the review, conducting the review and reporting the review emphasis announced on yearly base, addressing both, research theory and practice.

2.1. Planning the review

The first phase of systematic literature review is the phase of planning the review. In this phase, we need to specify the research question, develop a review protocol and validate the review protocol (Brereton et al., 2007).

To find papers that study various combinations of agile methods and their impact on software quality we set the following research questions:

- RQ1: What combinations of agile methods and practices are used in hybrid approaches with reported impact on software quality?
- RQ2: How the use of agile and hybrid approaches impacts the software quality?

Study sources that will be used in this research have been recommended by Hoda et al. (2017) as the most relevant sources to software engineering based on their previous research and experience. The list of relevant database sources along with the search string used to obtain the data are presented in Table 1.

Table 1. Study sources and search string

Digital databases	IEEExplore ACM Digital library Springerlink Science Direct Web of Science
Search string	((agile OR hybrid) AND (“software development” OR “system development”) AND (“software quality ” OR “product quality” OR “system quality”))

The following procedure is going to be used for study selection: I) after performing an automatic search based on a defined search string; II) relevant papers are going to be extracted by applying inclusion criteria on the title and on the abstract; III) papers are also going to be filtered by introduction and conclusion and duplicates will be removed, finally IV) the data will be extracted and quality assessment criteria are going to be applied based on the information obtained from full-text review.

Inclusion criteria are defined as follows:

- Study is an original scientific paper.
- Study focuses on the impact of agile methods and their combinations on software quality.
- Study is published after 2015.
- Study is written in English.
- Study is published in a journal or conference.

Furthermore, in order to filter the rest of the articles, we set questions for quality assessment, shown in Table 2. All answers ought to be positive in order for study to pass to the data extraction phase.

Table 2. Questions for quality assessment

Does the paper focus on the impact of the agile-hybrid approach on software quality?	YES/NO
Does the study explain the usage of agile or hybrid approaches?	YES/NO
Does the study provide factors that affect software quality?	YES/NO
Are the results related to research questions?	YES/NO

2.2. Conducting the review process

Table 3. represents the results of the selection process. In the first phase, 3959 studies were extracted after performing an automatic search in defined databases. In the second phase, we extracted 92 papers by applying inclusion criteria on the title and abstract. 52 studies were a result of the third phase where we filtered papers by applying inclusion criteria on introduction and conclusion. In the fourth phase, we read full-text and applied quality assurance criteria.

Table 3. Results of selection process

Source	**Phase 1**	**Phase 2**	**Phase 3**	**Phase 4**
IEE Explore	926	12	9	5
ACM Digital library	626	19	9	1
Springerlink	1490	18	8	1
Science Direct	791	18	10	3
Web of Science	126	25	16	5

The total of 15 papers passed all inclusion and quality assessment criteria and were used in the data extraction phase. Studies were analysed in order to answer research questions. Title, type of the study, and major results and conclusions were extracted from the studies.

3 Results

Research results of the systematic literature review process will be presented in relation to research questions.

Combinations of agile methods and practices (**RQ1**) which are used in hybrid approaches with reported impact on software quality were the first focus of our study and we identified 7 combinations of agile methods that are reported in scientific literature to have an impact on software quality. Table 4 shows the summary of agile-hybrid methods found in the literature.

Table 4. Summary of agile-hybrid methods found in the literature

Agile-hybrid methods	Study
Agile with Dynamic Quality Control	Tommy et al. (2015)
Hybrid agile model (Scrum - XP - Lean Software Development)	Neelu and Kavitha (2021) Jain et al. (2018) Kasims et al. (2018) G. Arcos-Medina and Mauricio (2020)
SDLC - Safe	Pradhan and Nanniyur (2021)
Test-driven development (TDD) Iterative test-last development (ITL)	Santos et al. (2021)
Q-Rapids as Rapid software development	Martínez-Fernández et al. (2018)
Waterfall - Agile	Pradhan and Nanniyur (2019) Pradhan et al. (2019) Bianchi et al. (2020)
Parallel agile (PA) and the incremental commitment spiral model (ICSM)	Boehm et al. (2019)

As for the research question on how the use of agile and hybrid approaches impacts the software quality (RQ2), the review showed that 22 agile practices have significant impact on software quality. The table below enumerates the practices and quality attributes they impact. The total number of 26 quality attributes are identified. Some of them are being impacted by different agile practices and are being repeated in the table.

Table 5. The impact of agile practices on quality attributes

Agile practices	Quality attributes	Study
Testing, Daily defect review meetings	Testability, Efficiency	Tommy et al. (2015)
Specification Requirement, Design, Implementation, Testing, Integration, Release	Availability, Scalability, Flexibility, Performance, Usability, Understandability	Neelu and Kavitha (2021)
Total no. of user stories, User story Accepted, Review effectiveness, Total no. of Sprints	Functionality, Reliability, Maintainability, Scalability	Jain et al. (2018)
Iteration (Sprint), Release every sprint, Automating the tests	Functionality, Testability, Efficiency	Kasims et al. (2018)
Testing cycles, Release	Number of defects, Annual failure rate	Pradhan et al. (2021)
Sprint	Overall quality	Bianchi et al. (2020)
Continuous integration, Pair programming, Stand-up meeting	Maintainability, Verifiability, Efficiency, Integrity, Usability, Testability,	Subih et al. (2019)

	Flexibility, Intra-operability	
Continuous integration	Time to develop, Introduced bugs, Time to deliver, Test quality, Documentation, Change management, Cost model,	Hamdan and Alramouni (2015)
Iterative practice, Prototyping, User testing, Automated testing	Testability, Verifiability, Efficiency	Berg et al. (2020)

4 Discussion

Nowadays, many companies want to implement systems in a minimum period of time at less cost (Subih et al., 2019). Because of that, they are using agile practices because they support flexibility, allow changes at any stage, and are focused on improving the quality of the software (Arcos-Medina & Mauricio, 2020) (Subih et al., 2019). Despite using new methodologies, numerous IT projects end with bad outcomes. They delivered over budget, in bad quality, and not meeting the deadlines. Also, a lot of time is wasted because of rework caused by quality issues (Kasims, 2018). This discussion focuses on analysing the connection between agile-hybrid methods and software quality.

Analysed papers have a different understanding of software quality. Some of them describe quality as meeting the customer needs and expectations (2018), while according to Arcos-Medina and Mauricio (2020) there are three types of software quality: Quality of design, quality of performance and quality of adoption. These qualities can be divided into quality attributes. Software quality measures how well the software is designed and how well it is working without faults (Jain et al., 2018).

As previously mentioned, adopting and adapting different approaches into agile-hybrid models, the benefits on software quality could be achieved. For instance, some studies focus on the testing process as a way to improve software quality. Tommy et al. (2015) states that the biggest challenge in software testing in agile methods is the less time and focus for the testing process, but they are with this study trying to prove that allocating more resources towards testing could influence on improving software quality without losing other benefits of agile methods. They presented the implementation of Dynamic Quality Control with agile methodology and its advantages. This hybrid of agile methodology and Dynamic Quality Control helps to improve uncovering hidden defects. Further, Pradhan et al. (2019) proposed a framework in which one of the objectives is to find more bugs in the early testing cycles. Results showed that the proposed metrics framework showed effectiveness in improving software quality. Furthermore, Q-Rapids quality model (Martínez-Fernández et al., 2018) as one of Rapid Software Development models, enables detecting problems faster and could be used in companies for comparing the quality of systems. Another example is an agile development method Test-driven development (TDD) which in correlation with Iterative Test-Last development (ITL) helps with improving software quality (Santos et al., 2021).

Some hybrid models use the best of each method to improve software quality. Hybrid agile model (Neelu & Kavitha, 2021) is a combination of Scrum, Extreme Programming, and Lean Software Development. Other authors do not use the same name, but are referring to the same agile combination (Jain et al., 2018) (Kasims, 2018) (Arcos-Medina & Mauricio, 2020). It is a quality driven agile development method. The biggest advantage of this model is the low defect rate, which results with high-quality software products.

Related to our second research question, different studies analyzing the impact of agile practices on quality attributes are identified. For example, several studies have shown that Process Automation has a big impact on software quality. Hamdan and Alramouni (2015) analyse impact of Continuous Integration on software quality, and their study showed that practicing continuous integration results with significant improvement of software quality. Also, having a release each sprint allows receiving feedback faster, which can be used to improve software quality. To achieve regular releases, teams need to identify tools for improving and automating the deployment process (Kasims, 2018). If the process is automated, it is more likely to do releases more often. One very important aspect of agile development is Sprint which is also positively related to software quality (Bianchi et al., 2020), but authors did not mention the exact quality attributes they are referring to. Similarly, Berg et al. (2020) customize iterative practices to achieve quality. They defined a version of sprints where customers or potential users can give feedback. Furthermore, according to Pradhan and V. Nanniyur (2021) quality is an important factor in every sprint in which agile teams need to define quality criteria. These criteria help them to measure quality improvement on a sprint-to-sprint basis. Also, the focus is on finding defects in every iteration. They emphasize finding defects in the intern testing cycles. For this reason, the team can fix them earlier and improve software quality.

Our overall judgement is that the topic of this research is a rather new topic in the research community. The majority of identified papers are published during the last three years, and we can see that authors are taking rather different, but mostly

partial, approaches and points of view. We couldn't identify break-through research that would set a headstone for classification of impact of use of agile practices on quality attributes, and thus we find that additional, more systematic approaches in research and classification of mentioned practices and their influence of quality attributes should be undertaken.

5 Conclusion

The paper provides an overview of agile-hybrid approaches and agile practices with reported impact on software quality.

The systematic literature review that we have performed on more than 3900 original studies identified 15 papers dealing with this topic, and resulted in identification of 7 combinations of agile methods that are reported to have an impact on software quality. Additionally, we identified 22 agile practices which have significant impact on as many as 26 software quality attributes.

Aligned with the goals defined in the review protocol, this research focused only on available scientific evidence, thus not taking into the account other possible types of literature sources, such as "gray" literature, technical reports, dissertations, on-line appearances et cetera. Such narrow pool of types of sources becomes the main limitation of this work. The authors are planning to overcome this limitation by conducting another SLR including mentioned sources as well.

Although this study shows that agile methods in combination with other agile principles could help to improve software quality, we also found out that this research topic is rather new, that there are no break-through works, and that additional systematic research is necessary for detailed classification of agile-hybrid methods and agile practices and their influence of quality attributes.

References

Alsaqaf, W., Daneva, M., & Wieringa, R. (2017). Quality Requirements in Large-Scale Distributed Agile Projects – A Systematic Literature Review. In P. Grünbacher & A. Perini (Eds.), Requirements Engineering: Foundation for Software Quality (pp. 219–234). Springer International Publishing. https://doi.org/10.1007/978-3-319-54045-0_17

Arcos-Medina, G., & Mauricio, D. (2019). Aspects of software quality applied to the process of agile software development: A systematic literature review. International Journal of System Assurance Engineering and Management, 10(5), 867–897. https://doi.org/10.1007/s13198-019-00840-7

Arcos-Medina, G., & Mauricio, D. (2020). Identifying Factors Influencing on Agile Practices for Software Development. Journal of Information and Organizational Sciences, 44(1), 1–31. https://doi.org/10.31341/jios.44.1.1

Berg, V., Birkeland, J., Nguyen-Duc, A., Pappas, I. O., & Jaccheri, L. (2020). Achieving agility and quality in product development—An empirical study of hardware startups. Journal of Systems and Software, 167, 110599. https://doi.org/10.1016/j.jss.2020.110599

Bianchi, M., Marzi, G., & Guerini, M. (2020). Agile, Stage-Gate and their combination: Exploring how they relate to performance in software development. Journal of Business Research, 110, 538–553. https://doi.org/10.1016/j.jbusres.2018.05.003

Boehm, B., Rosenberg, D., & Siegel, N. (2019). Critical Quality Factors for Rapid, Scalable, Agile Development. 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), 514–515. https://doi.org/10.1109/QRS-C.2019.00101

Brereton, P., Kitchenham, B. A., Budgen, D., Turner, M., & Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. Journal of Systems and Software, 80(4), 571–583. https://doi.org/10.1016/j.jss.2006.07.009

Fitriani, W. R., Rahayu, P., & Sensuse, D. I. (2016). Challenges in agile software development: A systematic literature review. 2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS), 155–164. https://doi.org/10.1109/ICACSIS.2016.7872736

Hamdan, S., & Alramouni, S. (2015). A Quality Framework for Software Continuous Integration. Procedia Manufacturing, 3, 2019–2025. https://doi.org/10.1016/j.promfg.2015.07.249

Hoda, R., Salleh, N., Grundy, J., & Tee, H. M. (2017). Systematic literature reviews in agile software development: A tertiary study. Information and Software Technology, 85, 60–70. https://doi.org/10.1016/j.infsof.2017.01.007

Jain, P., Sharma, A., & Ahuja, L. (2018). The Impact of Agile Software Development Process on the Quality of Software Product. 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 812–815. https://doi.org/10.1109/ICRITO.2018.8748529

Kasims, G. (2018). Applying Lean to Improve Quality in Software Development Projects. Proceedings of the 2nd International Conference on Business and Information Management, 130–134. https://doi.org/10.1145/3278252.3278254

Kitchenham, B., & Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering.

Martínez-Fernández, S., Jedlitschka, A., Guzmán, L., & Vollmer, A. M. (2018). A Quality Model for Actionable Analytics in Rapid Software Development. 2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 370–377. https://doi.org/10.1109/SEAA.2018.00067

Neelu, L., & Kavitha, D. (2021). Estimation of software quality parameters for hybrid agile process model. SN APPLIED SCIENCES, 3(3). https://doi.org/10.1007/s42452-021-04305-0

Pradhan, S., & Nanniyur, V. (2021). Large scale quality transformation in hybrid development organizations – A case study. Journal of Systems and Software, 171, 110836. https://doi.org/10.1016/j.jss.2020.110836

Pradhan, S., & Nanniyur, V. (2019). Back to Basics—Redefining Quality Measurement for Hybrid Software Development Organizations. 2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE), 402–412. https://doi.org/10.1109/ISSRE.2019.00047

Pradhan, S., Nanniyur, V., Melanahalli, P., Palla, M., & Chulani, S. (2019). Quality Metrics for Hybrid Software Development Organizations – A Case Study. 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), 505–506. https://doi.org/10.1109/QRS-C.2019.00097

Santos, A., Vegas, S., Dieste, O., Uyaguari, F., Tosun, A., Fucci, D., Turhan, B., Scanniello, G., Romano, S., Karac, I., Kuhrmann, M., Mandic, V., Ramac, R., Pfahl, D., Engblom, C., Kyykka, J., Rungi, K., Palomeque, C., Spisak, J., … Juristo, N. (2021). A family of experiments on test-driven development. EMPIRICAL SOFTWARE ENGINEERING, 26(3). https://doi.org/10.1007/s10664-020-09895-8

Subih, M. A., Malik, B. H., Mazhar, I., Izaz-ul-Hassan, Sabir, U., Wakeel, T., Ali, W., Yousaf, A., Bilal-bin-Ijaz, Nawaz, H., & Suleman, M. (2019). Comparison of Agile Method and Scrum Method with Software Quality Affecting Factors. INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS, 10(5), 531–535.

Thangiah, M., & Basri, S. (2016). A Preliminary Analysis of Various Testing Techniques in Agile Development—A Systematic Literature Review. 2016 3RD INTERNATIONAL CONFERENCE ON COMPUTER AND INFORMATION SCIENCES (ICCOINS), 600–605.

Tommy, R., Mhaisekar, M., Kallepally, S., Varghese, L., Ahmed, S., & Somaraju, M. D. (2015). Dynamic Quality Control in Agile Methodology for Improving the Quality. In 2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS) (pp. 233–236). Ieee.