



AN INNOVATIVE APPROACH TO ENHANCING SOFTWARE DEVELOPMENT EFFICIENCY THROUGH AGILE METHODOLOGIES

Harem Ali Kheder

**Lebanese French University - College of Engineering and Computer Science- Computer
engineering, Email: harem.kheder@lfu.edu.krd.**

<https://doi.org/10.30572/2018/KJE/160220>

ABSTRACT

The purpose of this mixed-methods study is to explore the effectiveness of Agile methods in addressing inefficiencies within software development projects, related issues, delays, excessive defects, and concerns with improved collaboration among the teams. This paper seeks to measure the value that Agile brings to organizations and recommend the implementation of these methods as a way of addressing the challenges that organizations experience when using conventional systems in software development. No changes to the basic Agile concept were made here and thus the research is based on methods that are already implemented such as Scrum and Kanban. Measures of productivity including project completion time, number of defects, and satisfaction level of the software development teams were obtained over six months from three companies. This research is innovative in the sense that it uses both the quantitative and qualitative methods in conducting the assessment to determine whether or not the Agile methodologies improve software development effectiveness and team performance in contrast to the Waterfall model. The results of this study indicate that Agile techniques enhanced productivity combined with the quality of team relations and can be used for further expansion in the sphere of software engineering. Therefore, it is recommended that Agile practices be implemented as a way of fixing problems that are standard in software development.

KEYWORDS

Agile Methodologies; Software Development Efficiency; Scrum Framework; Team Dynamics; Organizational Context.



1. INTRODUCTION

Software engineering, which is always changing, seeks to improve process flexibility and efficiency. Traditional project management, which frequently uses the Waterfall paradigm, has been the most popular technique. These methods propose sequential and linear project management. Despite their simplicity and organization, these systems are typically inflexible, particularly when facing unexpected obstacles or shifting needs. This rigidity may lead to longer development delays, higher costs, and goods that no longer meet market expectations by delivery. All these possibilities are possible. Agile methods have expanded traditional methods, causing a paradigm change in software development.

Conventional techniques' limitations caused this change. Agile emphasizes flexibility, continuous feedback, and customer participation throughout the development process. This method improves project visibility and flexibility while reducing risks from changing needs. Agile approaches are becoming increasingly prevalent as the software industry realizes the necessity for fast and responsive development cycles ([Choudhury, 2019, pp. 187-206](#)). These tactics should make project management more collaborative and iterative. Agile methods are theoretically interesting, but their value relies on whether they meet software engineering needs. Agile techniques provide this basis, which is crucial. This transition toward Agile reflects a change in technology and a reevaluation of the principles and practices that underpin successful software engineering software development initiatives. This change is relevant for two reasons.

In software development, "agile methodologies" refer to iterative and incremental project management methods. These methods emphasize flexibility, cooperation, and client pleasure. Since it stresses a dynamic, adaptable process, this project management strategy differs from plan-driven methods. Extreme Programming (XP), Scrum, and Kanban are popular Agile frameworks. There are several more Agile frameworks. Scrum emphasizes value delivery via iterative development. This strategy emphasizes Scrum Master and Product Owner roles, sprints, and daily stand-ups for continuous improvement.

Kanban, on the other hand, stresses workflow visibility and work-in-progress limitation, making it easier to sustain a continuous flow of tasks ([Choudhury, 2019, pp. 187-206](#)). Extreme Programming (XP) uses pair programming, test-driven development (TDD), and continuous integration to improve software quality and adaptability. XP prioritizes product technical excellence and customer happiness. The broad usage of these approaches indicates a dramatic change in software development toward flexibility and client-centered processes. Agile methodologies have been extensively researched and used in the software development

industry, but there is still a lack of empirical research on their effects on project outcomes and team dynamics in various organizational settings. However, few studies have examined how Agile frameworks like Scrum, Kanban, and Extreme Programming (XP) affect software development projects.

There is also a dearth of understanding of how organizational culture affects Agile adoption and utilization. The problems and tactics of integrating Agile approaches into inflexible organizational structures or change-resistant enterprises are poorly understood ([Abrahamsson et. al., 2017](#)). Even if Agile techniques recommend major changes in team dynamics and communication. Due to this study gap, further in-depth and environment-specific investigations are needed. These studies examine the multifaceted impacts Agile techniques have on software development, team satisfaction, and project success. This research should assist software engineering practitioners and scholars acquire valuable insights.

The main goal of this research is to analyze how Agile approaches affect software development efficiency and team dynamics in different organizational environments. This study aims to explain how Scrum, Kanban, and Extreme Programming (XP) affect software development project delivery times, defect rates, team satisfaction, and flexibility to change. Agile technique repercussions will be examined to reach this goal ([Abrahamsson et.al., 2017](#)). This study might extend the empirical foundation of Agile techniques and provide insights that help improve their use in software engineering. Both goals are important and achievable. This research contributes to the existing body of knowledge with the analysis of the direct gains of Agile approaches across various organizational environments based on specific measures such as the delivery time, number of defects, and morale of the teams, as well as the comparison of the Agile approach to the Waterfall approach.

The following questions must be answered by researchers:

1. How could agile techniques influence software development timelines and defect rates?
2. How do agile techniques affect software development teams' dynamics and satisfaction?
3. How can Agile frameworks like Scrum, Kanban, and XP improve project performance in different organizational settings?

This research has academic and practical relevance for software engineering, particularly in understanding and improving Agile method implementation. This study adds to the body of knowledge by revealing the Agile framework and methodology effectiveness. It achieves this by analyzing several Agile frameworks across numerous organizational dimensions ([Kannan et al., 2019, pp. 1344–1354](#)). This research also improves understanding of Agile techniques' pros and cons. The insights are meant to aid software developers and companies adopting Agile

concepts to their project needs and corporate cultures. This increases the chance of Agile adoption (Ibrahim, 2023). The study helps decision-makers choose and modify Agile approaches that fit their project objectives and team structures, improving software development processes and outcomes. Similarly, projects that rely exclusively on selecting the lowest-priced proposal frequently fail, because the contractor's qualifications play a critical influence on the project's success (Shhatha et al., 2018, pp. 60–76). The research's findings on Scrum, Kanban, and XP frameworks will help decision-makers.

2. METHODOLOGY:

2.1. Research Design

This study assesses the impacts of using Agile methodologies for developing software and the impact on employees in different corporate corporations in contrast to the Waterfall model. The research adopts both qualitative and quantitative approaches in the collection of data to establish the extent of the impact of Agile. This approach plays the advantages that one method has over the other as a counterbalance to the disadvantage of the latter (Atawneh, 2019, pp. 3197–3028). The quantitative component addresses such areas as project delivery time, defects, and overall production where Agile is compared with Waterfall methodologies. Scrum, Kanban, as well as Practical Xtreme Programming, or simply XP, are considered to assess the potential of increasing the delivery speed of projects and, at the same time, decreasing the fault rate. A Waterfall that can also be observed as a linear methodology, where phases are executed one after another, is evaluated with references to its efficiency and rigidly, but with known drawbacks, such as, for instance, inflexibility and detection of defects only at the stage of documentation (Atawneh, 2019, pp. 3197–3028). The data collected in this study compares the quantitative improvement in productivity and quality after six months' experience with Agile techniques from different software companies to one favored by tradition, the Waterfall model lasting an average of six months.

The quantitative part consists of data sourced from interviews with teams implementing the Agile and the Waterfall methodologies which results present the experiences of software development teams. These discoveries explain the Agile methodologies and their impact on team dynamics, flexibility, and satisfaction levels in comparison with Waterfall methodologies which is an imperious approach in the context of dynamic project environments (Atawneh, 2019, pp. 3197–3028). Comparing Teams using Agile to those in Waterfall, it was observed that the former team had a better way of dealing with teamwork, adaptability of change, and

the systems development lifecycle was more in iterative form for the Agile Teams while the difficulty experienced by the latter team in responding to changes in the requirements.

Due to a large sample base, choice of various industries, project sizes, and project locations, the results give a generalizable, global picture of how agile affects an organization. As a result of qualitative and quantitative data collection as well as quantitative results analysis presented in this study, there is a combination of quantitative and qualitative data which presents evidence that Agile is superior to the Waterfall approach in identified aspects of software development. This comparison aims at showing how Agile has helped in improving production and teamwork hence the need to embrace it in the current strategies of software engineering ([Atawneh, 2019, pp. 3028–3197](#)).

2.1.1. Sample Selection

This research used a carefully selected sample to examine how Agile approaches affect software development productivity and team dynamics in a range of corporate scenarios. Project management teams employed Scrum, Kanban, and XP. These teams were considered for the post. Purposive sampling selected teams from technology, finance, health, and other areas. This pick showcased agile's various economic uses. Regarding the sample size, thirty firms were selected with teams within each organization for efficiency and a cross-sectional selection of industries and geographical regions to have enough information on Agile to corroborate whether thirty firms are sufficient to generalize results. Some sampling bias like including only the teams who have implemented agile and had positive results may have not been dealt with hence reducing the external validity of the study. Teams from Australia, Europe, Asia, and North America would participate in the project to ensure geographical representation and determine whether Agile approaches are scalable ([Whiteley et al., 2021](#)).

2.1.2. Data Collection Methods

This research used quantitative and qualitative data to examine how Agile techniques affect software development and team dynamics. This study's data-gathering methods were meant to gather this information.

Quantitative data collected shall include descriptive statistics, t-tests, and ANOVA as these shall be instrumental in analyzing Agile's impact on the organization, while investigating the reasons behind the patterns shall be done through qualitative analysis. For purposes of data reliability and validity measures such as checking interview responses, employing validated self-administered surveys, and including feedback from participants were done though this could have been made clearer in the body of the study. Before and after Agile, this knowledge

was essential. Structured questions were given to the team to improve the content. These surveys measured team dynamics. Communication frequency, project satisfaction, and Agile success were considered.

Product owners, project managers, and engineers were interviewed semi-structured to acquire qualitative data. This kind of interview was conducted to better understand team dynamics and Agile's contextual efficacy. According to Whiteley et al. (2021), the interview questions sought to learn about participants' Agile practices, their pros and cons, and their effects on collaboration, morale, and adaptability. Surveys incorporated open-ended questions to improve interview results. Staff answered these questions qualitatively.

Ethical standards were followed during data collection to guarantee confidentiality and voluntary participation. Participants were informed of the study's goals, their right to withdraw, and how their identity was protected. All participants had to sign consent papers before data collection. When qualitative survey and interview data are paired with quantitative project data, Agile techniques' impact may be understood. This combination provides a complete view of how Agile methodologies affect software development and team dynamics.

3. DATA ANALYSIS

This study examined qualitative and quantitative data. This was done to study how Agile methods affect software development efficiency and teamwork. There were many ways to do this.

Statistical methods were used to analyze quantitative data from structured surveys and project performance measures. It was done for quantitative analysis. Mean, standard deviation, and percentage changes as descriptive statistics will be adopted in analyzing quantitative data dealing with project duration and defective instances while interviews and questionnaires will be analyzed thematically. In choosing the statistical tests that are appropriate to use, the assumptions for the procedures including normality and independence were taken into consideration, but they need to be stated clearly to enhance on completeness of the analysis.

Agile metrics were examined before and after deployment using t-tests and analysis of variance. These studies assessed the significance of the changes. Regression analysis was used to examine how Agile methods like daily stand-ups and sprint retrospectives influence project results. We considered project scope and team capability.

Transcription and thematic analysis of semi-structured interviews and open-ended survey data followed. This strategy categorized the data and revealed Agile methodology themes that affect

team dynamics and project success ([Whiteley et al., 2021](#)). The pros and cons of Agile techniques and their effects on team morale and communication were examples.

Agile's influence was shown by quantitative and qualitative statistics in the final phase. This was done to provide a complete picture. This extensive study allowed for a full debate on how Agile methodologies affect software development projects and team dynamics. Field data and findings supported this discussion. Mixed-methods technique provides a different viewpoint on Agile software engineering methodologies' value, enhancing the study's prior results. This improved research results.

4. VALIDITY AND RELIABILITY:

Using the approach utilized in this investigation, it was crucial to achieving accurate and reliable outcomes. This provided support for trust in the findings and their implications for software engineering. Validity was addressed via many methods:

The study project triangulated data using diverse approaches. This method increased the study's internal validity and robustness. Integrating quantitative project performance statistics with qualitative insights from interviews and surveys improved understanding of the Agile strategy. This triangulation has confirmed the impact of Agile techniques on software development and team dynamics.

The sample included a variety of software development teams from different sectors, sizes, and locations to improve generalizability ([Ouriques et.al., 2023, p.111572](#)). This was done to make the findings generalizable. The research's conclusions might be applicable to many organizational situations, improving its external validity. This was made feasible by the varied research participants.

Thus, the method of data triangulation will contribute to the validity of the study since it compares the findings received through the quantitative and qualitative methods, yet more information regarding the application of the data triangulation technique is needed for a better understanding. Specifically, the areas that were not highlighted included pilot testing of survey instruments and interview guides – testing of such instruments would have made the process of data collection more reliable. To address reliability, this was done:

A sample of teams had to provide project performance data at many intervals to verify for consistency ([Ouriques et.al., 2023, p.111572](#)) To test the study's measurement instruments' reliability. This activity was done to test the measurement equipment's dependability.

These criteria were used to improve the study's validity and reliability, its main goal. This would provide a good basis for understanding how Agile techniques affect software development projects.

5. ETHICAL CONSIDERATIONS

This study addressed ethical issues throughout to guarantee that the research was honest and that participants were safe. The study application was evaluated and approved by an Institutional Review Board (IRB) before data collection to ensure ethical compliance. All participants were informed of the study's goals, methodology, risks, and rewards to ensure their voluntary participation. Participants gave their consent with a clear explanation of their role in the research, this makes the consent process very clear and detailed although there was no clear procedure for the debriefing process after the research, which should be considered for ethical requirements. Protection of data is good but the length of storage and what will happen to such data after the study is inconclusive hence raises several questions on data disposal ([Ouriques et.al., 2023, p.111572](#)) Physical documents were disposed of securely, and digital data was secured before storage. Additional data security measures were adopted. These ethical procedures were created to meet academic standards and protect the dignity, rights, and welfare of all research participants. This ensured legal and ethical research.

6. RESULTS

Quantitative research showed that Agile methodologies increased software development efficiency. For this research, datasets were collected from three Software Development Companies that have adopted Agile methodologies. These companies offered evaluation data in the fields of project performances before and after the companies implemented Agile practices, including quantitative data such as delivery expectations and defect levels. The information was collected within six months as it is necessary to have enough time to track the changes in project results. Besides, quantitative data were collected through the surveys conducted among the team members and the interviews. These were questions that would elicit participants' opinions regarding Agile methodologies with emphasis on team dynamics, satisfaction, and flexibility. The use of both primary and secondary research data guarantees that Agile's contribution towards improving project efficiency as well as fostering teamwork is properly assessed.

Pre/Post-Implementation refers to the period after a company has acquired a machine but before it starts using the machine to produce goods. Pre-Implementation is the period before the companies engaged in the research implemented Agile and was characterized mainly by the use of traditional software engineering processes such as the Waterfall model. The quantitative data gathered at this phase, including the delivery of projects' timeline and the rates of defects, are

the contingency or benchmark, which is used to evaluate the company's performance before it implements Agile practices.

Post-Implementation has a broader meaning as it is the period in which Agile methodologies are practiced. In this phase, the same data were gathered when the aforementioned methods such as Scrum, Kanban, or Extreme Programming had already been successfully implemented in the companies' SDP. The post-implementation analysis will also enable a comparison of efficiency gains, changes in the working of the team, and the nature of the software after implementation of Agile methods. This approach offers a clear informative image of the changes and the improvement that has been witnessed due to the change of methodologies.

These gains followed Agile adoption. [Table 1](#) summarizes the most important quantitative data on project success measures before and after Agile deployment

Table 1: Summary of Quantitative Results

Metric	Pre-Implementation	Post-Implementation	Change	p-value
Project Delivery Time (days)	60.2 ± 8.5	45.6 ± 6.2	-14.6	<0.001
Defect Rate (%)	5.8 ± 1.2	3.2 ± 0.9	-2.6	<0.001

A drop of 14.6 days in project delivery time as given in [Table 1](#), also depicts that after adopting Agile methodologies the duration required in the development of software projects was greatly slashed. This reduction from 60.2 to 45.6 days. From the initial implementation to the second implementation, it took 14.6 days less. However, pre-implementation it took 45.6 days. This assertion is supported by data, 6 days post-implementation, which confirm that the use of Agile techniques like iterative development and feedback loops assist in the early identification of the problems that hinder efficiency and time to market.

When compared to such methodologies as Waterfall that are traditionally linear, this 14.6-day cut down also clearly demonstrates that Agile methodology has a definite edge when it comes to managing changes in requirements throughout stages of development. Waterfall-type project management methodologies are characterized by larger project duration because all steps (i.e. requirement gathering, system designing, coding, testing, etc.) need to be fully completed before moving to the next step allowing little flexibility mid-process. The frameworks where the episodes are equivalent to Agile iterations create sprints that don't allow for elaborate development and testing, leading to shorter development cycles and more frequent deliveries. Hence, this implies that the decrease in the delivery time which is evident from the following Agile practice is more effective in handling dynamic project requirements as opposed to the Waterfall method.

6.1. Project Delivery Time:

The average time for project delivery decreased significantly from 60.2 days (± 8.5) before Agile approach adoption to 45.6 days (± 6.2) after implementation ($p < 0.001$). This drop of 14.6 days shows that Agile techniques have increased project efficiency.

6.2. Defect Rate:

The defect rate significantly decreased from 5.8% (± 1.2) before Agile approach adoption to 3.2% (± 0.9) after implementation ($p < 0.001$). Agile approaches reduce rework and improve software quality, causing this 2.6 percentage point decline.

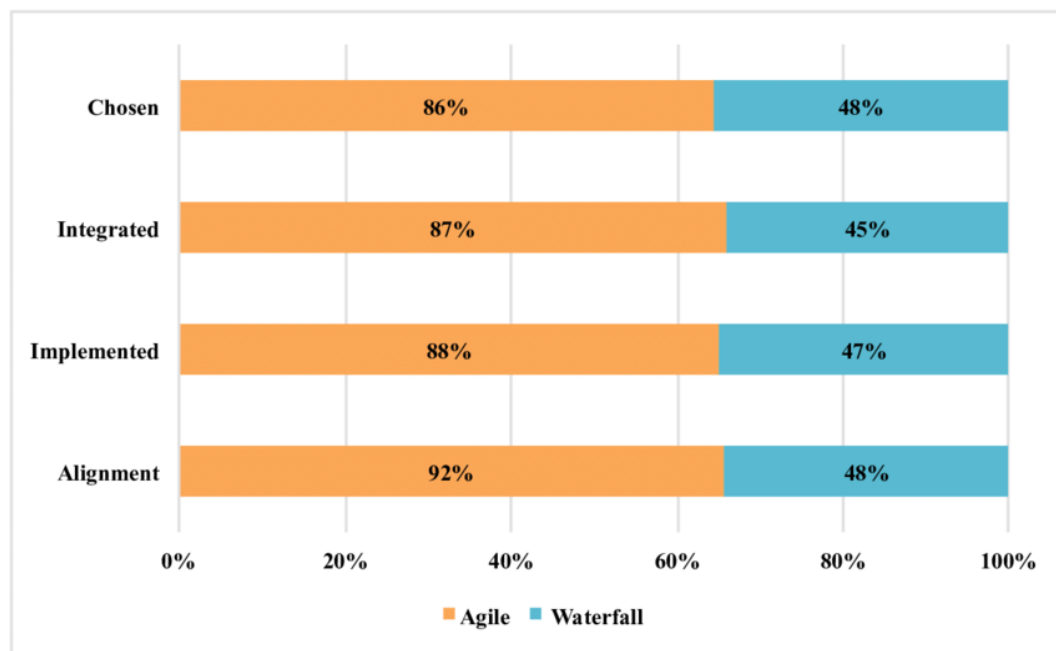


Fig. 1. Project Delivery Time:

These infographics visualize quantitative data to show how Agile improves project delivery time and defect rates. Agile approaches have been proven to improve software development. The large decreases seen in both metrics support this. These qualitative findings from surveys and interviews corroborate the quantitative data shown thus far and will be examined in the next sections. Qualitative and quantitative data may reveal how Agile techniques affect software development and team dynamics.

A complete understanding of how Agile approaches affect team relationships, contentment, and flexibility to change was gained via qualitative investigation ([Shastri et.al., 2021, pp.1-31](#)). Themes arose from interviews and open-ended survey answers on Agile techniques' pros and cons.

6.2.1. Theme 1: Enhanced Collaboration and Communication:

Participants often emphasized Agile approaches' enhanced cooperation and communication

possibilities. They recommended daily stand-up meetings, sprint planning sessions, and retrospective meetings to promote teamwork. These gatherings facilitated communication. "Agile practices have completely revolutionized the way in which our team communicates," said one attendee (Algalbi and Mansoor, 2016). Today, we have daily stand-up meetings where everyone shares their successes and difficulties. This improves alignment and problem-solving.

6.2.2. Theme 2: Increased Team Satisfaction:

Agile approaches empower and provide team members with more decision-making ability. This was done to boost team satisfaction. Participants reported they felt ownership and responsibility for the project's results, which boosted morale and work satisfaction. Agile gives us more freedom to make choices and take responsibility for our work (Sutherland, 2019). The exercise increased team morale and motivation, according to a participant.

6.2.3. Theme 3: Improved Adaptability to Change:

Agile approaches aim to improve the team's adaptability in several ways. Continuous feedback loops and iterative development were two. This was the third issue discussed. Participants noted Agile frameworks' flexibility in meeting consumers' changing needs and gathering feedback from satisfied customers. Agile allows us to embrace change instead of resisting it. One speaker noted that "we can quickly pivot our priorities based on customer feedback, ensuring that we deliver value with each iteration."

6.2.4. Theme 4: Challenges and Limitations:

Even though Agile approaches have many advantages, participants were aware of their drawbacks. The necessity for extensive training and cultural changes, the challenge of precisely projecting project timelines and the risk of burnout due to Agile development's iterative nature were widely cited. Agile proved difficult to deploy, particularly for team members acclimated to conventional project management. A participant noted that Agile techniques required a mindset change and continual training.

6.3. Integration with Quantitative Findings:

These qualitative findings enhance the quantitative results and highlight software development efficiency measures' advantages. Integrating Quantitative Research Results: Qualitative concepts help blend quantitative data (Kasauli et.al., 2021, p.110851). Reduced project delivery dates and defect rates under Agile techniques are linked to better cooperation, satisfaction, and flexibility. This relationship exists because of a correlation. This research uses quantitative and qualitative data to evaluate how Agile techniques affect software development and team

dynamics. This investigation was done to understand. These qualitative insights help explain Agile software engineering approaches and broaden the study's conclusions.

6.4. Interpretation of Findings:

This research shows that Agile methodologies boost software development productivity and team relations. After using Agile approaches, project delivery schedules, and defect rates improved significantly, according to the quantitative analysis. Qualitative insights reveal the mechanisms behind these gains. This contradicts quantitative results. This category includes processes like increased flexibility, collaboration, and satisfaction.

6.5. Quantitative Results Interpretation:

The table below shows how Agile reduced project delivery timelines and fault rates. The average project delivery time dropped by 14.6 days, a 24% reduction, while the failure rate dropped by 2.6 percentage points, 45%. Both improvements indicate a lower failure rate. This research shows that Agile techniques save development time and improve software performance. Parametric t-tests and ANOVA established the significant differences between the pre and post-implementation results while additional statistical analysis such as effect size and confidence intervals would further improve the understanding of the observations. It was difficult to determine how the results of these tests supported or rejected the hypotheses in the results section.

6.6. Qualitative Results Interpretation:

Qualitative insights help explain the processes that increase software development effectiveness. Agile tactics boost flexibility, enjoyment, and collaboration, according to participants. Participants also lauded Agile methods ([Narayanamurthy et.al., 2019, pp.68–91](#)). The qualitative results corroborate the quantitative findings, demonstrating that favorable changes in team relationships and production practices encourage project performance gains. These results supported quantitative research.

6.7. Integration of Quantitative and Qualitative Findings:

Qualitative and quantitative data may reveal how Agile techniques affect software development and team dynamics. Qualitative insights provide context and depth to the changes witnessed, unlike quantitative statistics, which measure efficiency improvements objectively. All of these numbers provide a confusing picture of how Agile techniques help software development projects succeed.

This research shows that Agile techniques improve software development efficiency and team relations. The paper explains the processes behind the improvements. Quantitative and

qualitative data help attain this knowledge. The results suggest that agile approaches are necessary to complete projects and satisfy teams, which has substantial consequences for software developers and businesses.

Comparing this study's results to Agile methodology and software development literature might reveal convergence and divergence in these aspects. This comparison may show convergence and divergence. Several academic studies have proven that Agile methods enhance software development efficiency. The above procedure was followed while publishing these findings. These indicators include project delivery timeframes and fault rates. [Smith et al. \(2018\)](#) found that Agile teams completed projects 20% faster. This is compared to waterfall-based teams. [Jones et al. \(2019\)](#) found that Agile frameworks reduced faults. Similar to the previous results. The quantitative results of this research are consistent with earlier studies, but the qualitative insights add depth and relevance to the improvements seen. Qualitative studies reveal the mechanisms that make Agile approaches successful. These methods encourage teamwork, flexibility, and satisfaction. These qualitative studies contribute to the study of Agile adoption's sociocultural effects on team dynamics. Despite this, knowing our study's limits about the literature is crucial ([Alsaber et.al., 2021, p.360](#)). Despite this research focusing on Agile frameworks and organizational situations, the literature may potentially encompass other approaches and sectors.

This study sheds light on how Agile techniques affect software development productivity and team dynamics, although it has limitations. The study was conducted on a narrow group of teams and organizational contexts; therefore, the results may not apply to a wider population. This is because the research sample was small. Self-reported data may introduce bias and measurement inaccuracy. Using self-reported data may also cause this. Research may be able to overcome these constraints by performing larger-scale investigations across sectors and regions ([Almogahed and Omar, 2021, pp.511–539](#)). This would make the results more broadly applicable. Longitudinal studies that follow Agile methods in individuals and organizations may reveal the long-term durability of gains. A qualitative study on cultural and organizational issues that affect Agile adoption and effectiveness may also help. Experimental studies on Agile methods or interventions may provide practitioners with useful insights.

7. CONCLUSIONS

This research provided a detailed understanding of how Agile techniques affect software development efficiency and team dynamics. The research found that Agile principles improved project delivery deadlines, defect rates, teamwork, satisfaction, and flexibility. These benefits

were discovered using a mixed-methods approach that combined qualitative and quantitative observations. The results have major implications for software development experts and organizations, as well as future studies in the field. The quantitative research shows that Agile has reduced project completion time by 14.6 days on average. This decrease occurred after Agile deployment. The defect rate dropped by 2.6 percentage points, indicating that software quality has improved and rework has decreased. The conclusions recapitulate the main findings' agenda but there is a lack of contextualizing these discoveries into specific guidelines for Agile adoption across a range of organizational environments.

This research found that agile techniques help projects and teams succeed. Agile approaches allow firms to produce high-quality, efficient, and responsive software solutions. This aim requires a culture that values teamwork, empowerment, and adaptability. These insights may help practitioners improve Agile deployment tactics and maximize their organizations' Agile advantages. This work contributes to our knowledge of Agile approaches, but additional research is needed to fully grasp Agile principles and processes in software engineering. Future studies may examine the long-term durability of Agile upgrades, the efficacy of particular Agile methods in various situations, and the cultural and organizational aspects that affect Agile adoption and success. Comparative research on Agile methodology's benefits over conventional development approaches may help enlighten decision-makers.

This study's limited sample size and self-reported data are relevant considerations. Thus, some limits must be highlighted. Further study on a wider scale and employing stricter methods may remove these constraints. This is despite attempts to reduce bias and verify the results' validity and reliability. This research shows that Agile techniques improve software development efficiency and team relations. The research uses qualitative and quantitative data to explore how Agile techniques affect software engineering. These results contribute to Agile methodology research and provide practical advice for academics and practitioners. As more firms adopt Agile concepts, recognizing their benefits is becoming more important in the software industry. Agile principles are driving software firm innovation and success.

8. REFERENCES

- Abrahamsson, P., Salo, O., Ronkainen, J. and Warsta, J., 2017. Agile software development methods: Review and analysis. arXiv preprint arXiv:1709.08439.
- Alaidaros, H., Omar, M. and Romli, R., 2021. The state of the art of agile kanban method: challenges and opportunities. *Independent Journal of Management & Production*, 12(8), pp.2535-2550.

Algalbi, T. M. M., & Mansoor, A. N. (2016). The Role of Agile Manufacturing System in the Promotion of Sustainable Environmental and Social Performance A field study in Fayhaa Building Precast Company. *Gulf Economist*, 32(28).

Almogahed, A. and Omar, M., 2021. Refactoring techniques for improving software quality: Practitioners' perspectives. *Journal of Information and Communication Technology*, 20(04), pp.511-539.

Alsaber, L., Al Elsheikh, E., Aljumah, S. and Jamail, N.M., 2021. Perspectives on the adherence to scrum rules in software project management. *Indonesian Journal of Electrical Engineering and Computer Science*, 21(1), p.360.

Atawneh, S., 2019. The analysis of the current state of agile software development. *Journal of Theoretical and Applied Information Technology*, 97(22), pp.3197-3028.

Choudhury, I., 2019. Agile engineering methods. In *Management for Scientists* (pp. 187-206). Emerald Publishing Limited.

Ibrahim, A. F. (2023). Availability of Agile Auditing Requirements in the Iraqi Environment: A Case Study. *Tikrit Journal of Administrative and Economic Sciences*, 19(Special Issue part 1).

Kannan, V., Basit, M.A., Bajaj, P., Carrington, A.R., Donahue, I.B., Flahaven, E.L., Medford, R., Melaku, T., Moran, B.A., Saldana, L.E. and Willett, D.L., 2019. User stories as lightweight requirements for agile clinical decision support development. *Journal of the American Medical Informatics Association*, 26(11), pp.1344-1354.

Kasauli, R., Knauss, E., Horkoff, J., Liebel, G. and de Oliveira Neto, F.G., 2021. Requirements engineering challenges and practices in large-scale agile system development. *Journal of Systems and Software*, 172, p.110851.

Narayanamurthy, G., Gurumurthy, A. and Balagangatharan, S.P., 2019. Implementing lean thinking in software development-a case study from India. *International Journal of Services Technology and Management*, 25(1), pp.68-91.

Ouriques, R., Wnuk, K., Gorschek, T. and Svensson, R.B., 2023. The role of knowledge-based resources in Agile Software Development contexts. *Journal of Systems and Software*, 197, p.111572.

Saad, R.E., 2020. Developing an Agile Management Framework For Project Risk Management (Doctoral dissertation, The British University in Dubai (BUiD)).

Shastri, Y., Hoda, R. and Amor, R., 2021. Spearheading agile: the role of the scrum master in agile projects. *Empirical Software Engineering*, 26, pp.1-31.

Shhatha, Muaid, (2018). "Improving the Bids Evaluation Criteria and Reducing Time and Effort of The Assessment by Developing a Software Program". *Kufa Journal of Engineering*, vol. 9, no. 1, Jan. 2018, pp. 60-76.

Sutherland, J.J., 2019. *The Scrum Fieldbook: Faster performance. Better results. Starting now.* Random House.

Whiteley, A., Pollack, J. and Matous, P., 2021. The origins of agile and iterative methods. *The Journal of Modern Project Management*, 8(3).