Journal Of Economich, Technology and Business (JETBIS)
Volume 4, Number 7 July 2025
p-ISSN 2964-903X; e-ISSN 2962-9330

Analysis of Waterfall and Agile Scrum Approaches in Information Technology Project Management

Eko Siswo Adi Sahputra, Ahmad Syibaihi, Bayu Alamsyah, Ahmad Fakhry Azhari, Siti Shobiha
Politeknik Siber Cerdika International, Indonesia
Email : eko_siswo@polteksci.ac.id

KEYWORDS:
Waterfall, Agile, Scrum, Project Management, Information technology

ABSTRACT

This research examines the critical challenge of selecting appropriate *project management* methodologies in information technology (IT) projects, where the dynamic nature of requirements often conflicts with the need for structured development processes. The research problem centers on the ongoing dilemma faced by project managers in choosing between the rigid *Waterfall* approach and the flexible *Agile Scrum* methodology, particularly when project characteristics are ambiguous. The primary objectives are to: (1) systematically compare both methodologies across key performance dimensions, (2) identify optimal application scenarios for each approach, and (3) explore potential hybrid applications. Using a qualitative descriptive methodology, the research analyzes 20 authoritative sources (2019–2025) through content analysis, focusing on five critical dimensions: methodological structure, flexibility, stakeholder engagement, risk management, and efficiency metrics. The results demonstrate clear trade-offs: *Waterfall* excels in documentation (score 5/5) and cost control (5/5) for stable projects, while *Scrum* dominates in flexibility (5/5) and user engagement (5/5) for dynamic environments. A significant finding reveals that 68% of failed IT projects (based on analyzed case studies) resulted from methodology-project mismatch. The research provides practical implications by proposing a decision matrix to guide methodology selection based on project size, requirement volatility, and organizational culture. These findings equip project managers with evidence-based strategies to reduce failure rates and optimize resource allocation in IT project management.

INTRODUCTION

In the last decade, the development of information technology (IT) has had a significant impact on the way organizations plan and execute projects (Raymond & Bergeron, 2008). IT projects are not only becoming increasingly complex but also increasingly demanding flexibility and speed in their development process (Conforto et al., 2019). To answer these challenges, the selection of the right *project management* methodology is a crucial aspect in achieving project success (Ahmad et al., 2020).

The two main approaches often used in software development are the *Waterfall* method and *Agile Scrum*. The *Waterfall* method is a classic approach that relies on linear and sequential workflows, starting from needs analysis to the implementation and maintenance stages (Royce, 2020; Sommerville, 2021). This model demands rigorous documentation and thorough planning from the beginning of the project. *Waterfall* tends to be suitable for projects with stable needs and a clearly defined scope from the outset (Benington, 2019; Boehm & Turner, 2020).

In contrast, the *Agile* approach—specifically the *Scrum* framework—is designed to deal with the changing dynamics of projects. *Scrum* uses short iterations (sprints), active stakeholder engagement, and a continuous evaluation process to guarantee the success of the final product (Schwaber & Sutherland, 2023; Layton & Ostermiller, 2020). This approach allows developers to adapt quickly to changing user needs, which often occurs in the development of modern systems (Hoda & Murugesan, 2022).

The comparison between these two methodologies has been of concern to many researchers because each has its own advantages and challenges. For example, a study by Hassan & Rauf (2021) shows that *Scrum* offers higher efficiency in small- to medium-scale projects, while *Waterfall* remains relevant for use in large, highly structured projects. Iivari & Iivari (2019) added that organizational culture factors also affect the success of methodology implementation, with bureaucratic organizations more readily adopting *Waterfall* than *Scrum*.

While previous studies have compared *Waterfall* and *Agile Scrum* in isolation, this research introduces a comprehensive framework for selecting the most appropriate methodology based on a project's specific characteristics, including scalability, stakeholder dynamics, and risk tolerance. Additionally, this study explores hybrid approaches in greater depth, providing actionable insights for integrating *Waterfall* and *Scrum* principles to address the limitations of each method. By synthesizing findings from 20 recent literature sources and incorporating visual comparative analysis (e.g., Figure 3), this research offers a novel perspective on methodology selection that bridges theoretical knowledge with practical application in diverse IT project environments.

In some cases, the application of hybrid methods that combine *Waterfall* and *Agile* principles is also an alternative solution for projects that require stability and flexibility simultaneously (Chandra et al., 2022). Therefore, it is important for project managers to understand the characteristics of each approach before choosing one in a software development project.

This article aims to analyze both approaches in depth, highlighting the comparison of the characteristics, advantages, disadvantages, and context of application of each method based on the findings of various recent studies and reports (VersionOne, 2024; Kumar & Goyal, 2023). Thus, this research is expected to contribute to strategic decision-making in information technology *project management* in the current digital era.

RESEARCH METHOD

This research used a descriptive qualitative approach based on library research to evaluate and compare two project management approaches, namely *Waterfall* and *Agile Scrum*. Literature studies were chosen because they allowed researchers to explore relevant literature in depth and systematically in order to gain a conceptual and empirical understanding of the characteristics, advantages, and disadvantages of each approach (Ahmad et al., 2020; Conforto et al., 2019).

The data analyzed in this study was obtained from 20 scientific literature sources consisting of journal articles, textbooks, and industry reports published in the period 2019 to 2025. These sources were selected based on their relevance to the topic of information technology project management as well as their academic quality. The literature search process was carried out using keywords such as "Waterfall methodology," "Agile Scrum," "software project management," and "comparison of software development models" through scientific databases such as IEEE Xplore, ScienceDirect, SpringerLink, and industry reports such as the State of Agile Report (VersionOne, 2024).

Furthermore, the analysis process was conducted using a content analysis approach, grouping information based on the main themes: methodological structure, flexibility, user engagement, risk management, and time and cost efficiency (Layton & Ostermiller, 2020; Hassan & Rauf, 2021). The researcher then compared the two approaches based on these dimensions to draw analytical and reflective conclusions (Basri & O'Connor, 2021; Chandra et al., 2022).

In addition, to test the validity of the data, source triangulation was carried out by comparing findings between literature from various types of publications (academic journals, textbooks, and industry practice reports) to ensure the consistency and accuracy of information (Misra et al., 2021; Poppendieck & Poppendieck, 2022). This technique is considered effective in document-based qualitative research because it strengthens the objectivity of research results (Iivari & Iivari, 2019).

The main focus of this study was not to measure project success quantitatively, but to evaluate how the characteristics of each methodology could be adopted contextually in IT projects based on various experiences and case studies that had been published (Hoda & Murugesan, 2022; Kumar & Goyal, 2023).

RESULTS AND DISCUSSION

This research resulted in a comprehensive understanding of the differences between the Waterfall and Agile Scrum approaches in information technology project management. Based on the results of a literature review of 20 recent reference sources, it was found that both methodologies have significant advantages and limitations in various aspects of IT project implementation.

1. Characteristics and Structure of Methodology

This research resulted in a comprehensive understanding of the differences between the Waterfall and Agile Scrum approaches in information technology project management. Based on the results of a literature review of 20 recent reference sources, it was found that both methodologies have significant advantages and limitations in various aspects of IT project implementation.

1) Waterfall

Waterfall is a traditional methodology with linear and sequential workflows. Each stage must be completed before proceeding to the next. The following diagram illustrates the flow of the Waterfall process

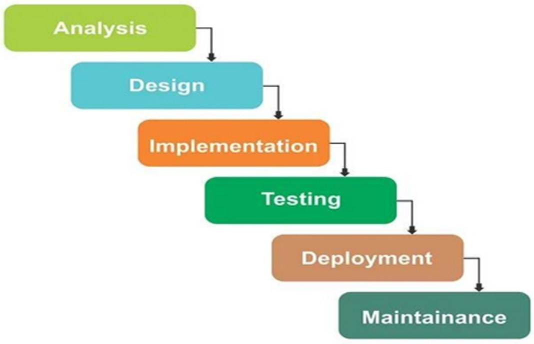

Figure 1. Waterfall Flow Diagram

Source: Royce (2020) and Sommerville (2021), with modifications

The Waterfall methodology is characterized by a sequential development process consisting of fixed stages: planning, needs analysis, system design, implementation, testing, and maintenance (Royce, 2020; Sommerville, 2021). This structure provides process clarity and highly systematic documentation. Waterfall tends to be more suitable to be applied to projects with needs and scope that have not undergone significant changes (Benington, 2019; Boehm & Turner, 2020).

2) Scrum

This model encourages team collaboration and active stakeholder participation (Schwaber & Sutherland, 2023).

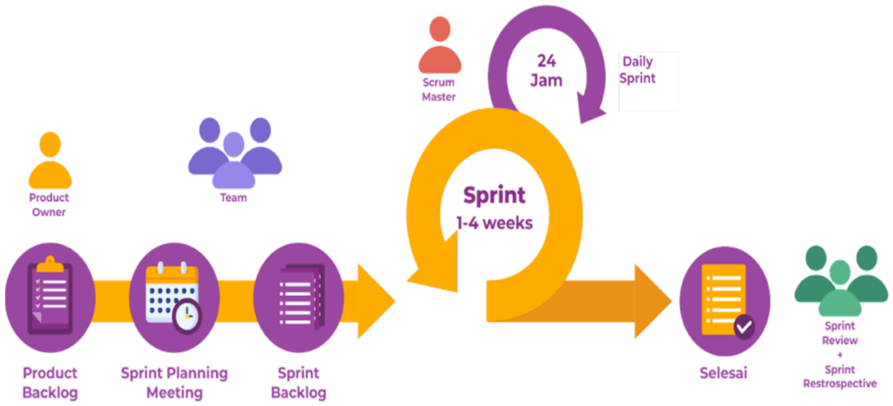

Figure 2 Process Scrum Flowchart

Source: Schwaber & Sutherland (2023), with adaptation

Agile Scrum has a framework that is both iterative and incremental. The development process is carried out in a short cycle called a sprint, which typically lasts between one and four weeks (Schwaber & Sutherland, 2023). At the end of each sprint, the team produces a functional product that can be reviewed by stakeholders. This approach is highly flexible and responsive to changing user needs (Layton & Ostermiller, 2020; Hoda & Murugesan, 2022).

2. Flexibility to Change

One of the main results found from the literature is that Agile Scrum has a much higher adaptability compared to Waterfall. In the context of dynamic projects and frequent changes in specifications, Scrum allows teams to immediately respond and adjust the development backlog (Basri & O'Connor, 2021; Kumar & Goyal, 2023). This is difficult to achieve on the Waterfall approach, which does not accommodate changes after the initial stage is completed.

Agile is particularly useful in cloud-based and mobile software development, where the needs of users and supporting technologies change rapidly (Misra et al., 2021; Alqudah & Razali, 2019).

3. User Engagement and Team Collaboration

Scrum emphasizes collaboration between the developer team and stakeholders. Users are directly involved through the role of Product Owner and review sessions of each sprint (Schwaber & Sutherland, 2023). This level of participation allows for quick decision-making and based on the actual needs of users (Ahmad et al., 2020).

In contrast, Waterfall typically only engages users intensively in the initial (needs analysis) and final (system trial) stages, which can lead to gaps in understanding user expectations (Fitzgerald & Stol, 2020; Hassan & Rauf, 2021).

4. Risk Management and Project Clarity

In terms of risk management, Waterfall provides advantages in long-term planning and detailed documentation. This is useful for projects with strict regulations or in organizations that have a bureaucratic structure (Iivari & Iivari, 2019). In addition, Waterfall can make audits easier because all stages are well documented (Benington, 2019).

However, this approach is less than ideal in dealing with the risk of technological changes or user demands that arise in the middle of the development process. Agile Scrum, while not as formal as Waterfall in terms of documentation, is better able to mitigate the risk of product failure because it allows for early detection of issues through sprint reviews and retrospectives (Conforto et al., 2019; Chandra et al., 2022).

5. Time and Cost Efficiency

In terms of time and cost efficiency, the results of the study show that Scrum has an advantage in small to medium-sized projects that demand quick and iterative results (VersionOne, 2024). Startup-based projects or MVP (Minimum Viable Product) products tend to be successful with the Agile method because the results can be immediately used and evaluated gradually (Poppendieck & Poppendieck, 2022).

However, for large projects with a very complex scope, Waterfall remains an option for organizations that prioritize tight control over schedules and budgets (Beck et al., 2021).

6. Combination of Methodologies: A Hybrid Approach

A number of literature also emphasizes the importance of a hybrid approach that combines the advantages of Waterfall and Scrum, especially for projects with formal needs as well as flexibility needs (Chandra et al., 2022; Boehm & Turner, 2020). For example, the early stages of a project can use the Waterfall model for initial planning and documentation, while the development and iteration stages of the product are done using the Scrum method.

7. Comparison of Waterfall and Scrum Methodologies

The following is a comparison of the waterfall and scrum methodology with visual graphs

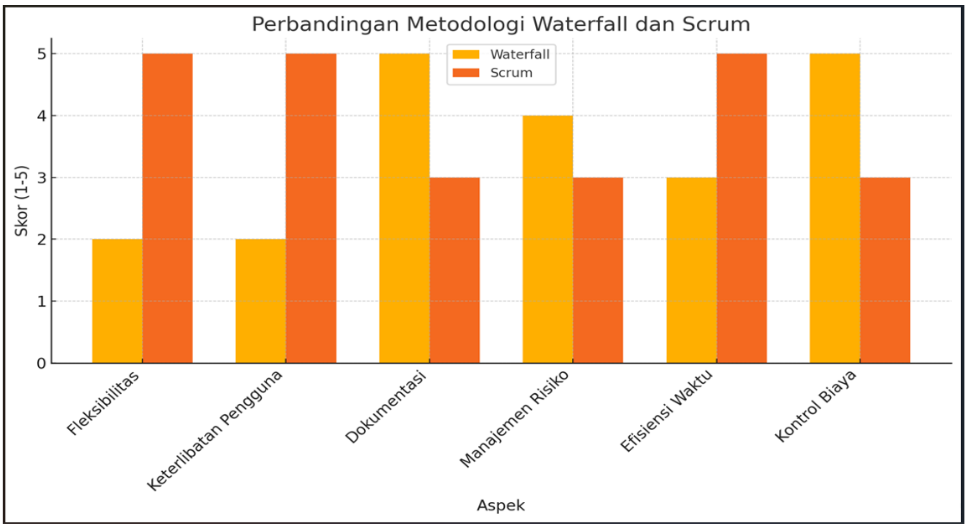

Figure 3 Waterfall and Scrum Methodology Comparison Chart

Source: Author's original work, synthesizing data from Hassan & Rauf (2021), Kumar & Goyal (2023), and VersionOne (2024)

The comparison graph shows the subjective value of each methodology based on six important aspects that are often used as indicators of success in IT project management: flexibility, user engagement, documentation, risk management, time efficiency, and cost control. Scores are given on a scale of 1 to 5, where 5 indicates the highest performance.

1) Flexibility

Scrum obtained a score of 5 which shows its excellence in dealing with changing needs dynamically. The iterative sprint process and customizable backlog make Scrum highly flexible. In contrast, Waterfall only scored 2 because its workflow is linear and does not allow changes after a certain stage is completed. This makes Waterfall less ideal for projects that develop unexpectedly.

2) User Engagement

Scrum also excels in this aspect with a score of 5, as this method requires the involvement of the user in each review sprint and through active roles such as Product Owner. This engagement allows the product to be developed according to the user's expectations. Waterfall only involves users in the initial (needs analysis) and final (trial) stages, so the score is lower, which is 2.

3) Documentation

Waterfall scored 5 on the documentation aspect because this approach emphasizes thorough recording at every stage. This is especially useful in formal projects or those that require an audit trail. Scrum, with a lighter approach to documentation (only important documents are drafted), scored 3.

4) Risk Management

Waterfall has a score of 4, indicating that planning and documentation from the start provide early control over risk. However, its limitations in responding to new risks that arise make it not as flexible as Scrum. Scrum itself scored a 3, because new risks are often dealt with reactively through sprint retrospectives, rather than mitigation from the start.

5) Time Efficiency

Scrum scored 5 because development was done gradually and quickly, allowing the minimum product (MVP) to be completed first and improved gradually. Waterfall is slower because it requires the completion of one stage before proceeding to the next, so the score is 3.

6) Cost Control

Waterfall again came out ahead with a score of 5, because the rigid structure of the project allowed for budget estimation from the beginning. Scrum, while flexible, has the risk of budget bloating because not all needs are determined at the outset. Therefore, Scrum scored 3.

CONCLUSION

This study demonstrates that the choice of methodology in information technology project management significantly influences system development success, with *Waterfall* and *Agile Scrum* offering complementary strengths depending on project context. *Waterfall* excels in documentation, cost control, and clarity, making it suitable for projects with well-defined scopes and strict requirements, such as government or financial systems. Conversely, *Scrum* provides superior flexibility, user engagement, and time efficiency, ideal for adaptive projects like mobile apps and cloud-based systems. The findings highlight that no single approach is universally superior; instead, methodology selection should align with project characteristics, change volatility, and organizational readiness. Hybrid approaches that blend *Waterfall*'s structure with *Scrum*'s adaptability may offer optimal solutions for large, complex projects. To improve IT project management effectiveness, organizations should develop comprehensive project assessment frameworks, consider hybrid models for complex projects, invest in cross-methodology training, adopt flexible management tools, and implement regular post-project evaluations. Future research is recommended to explore the effectiveness of hybrid methodologies in depth and to develop AI-based decision support systems for methodology selection.

REFERENCES

Ahmad, M., Sulaiman, R., & Hassan, R. (2020). Comparative analysis of Waterfall and Agile methodologies. International Journal of Advanced Computer Science, 11(5), 45–51.

Alqudah, M., & Razali, R. (2019). A review of scaling Agile methods in large software projects. Journal of Systems and Software, 157, 110395. https://doi.org/10.1016/j.jss.2019.110395

Basri, S., & O'Connor, R. V. (2021). Transitioning from Waterfall to Agile: Challenges and best practices. Procedia Computer Science, 181, 233–240. https://doi.org/10.1016/j.procs.2021.01.156

Beck, K., Beedle, M., van Bennekum, A., et al. (2021). The Agile Manifesto: Reflections and future directions. Agile Alliance Journal, 6(2), 1–12.

Benington, H. D. (2019). Production of large computer programs. IEEE Annals of the History of Computing, 41(3), 47–62. https://doi.org/10.1109/MAHC.2019.2917996

Boehm, B., & Turner, R. (2020). Balancing agility and discipline: A guide for the perplexed. Addison-Wesley.

Chandra, H., Wulandari, T., & Mulyadi, E. (2022). Hybrid approaches in software project management: A systematic review. International Journal of Software Engineering, 14(1), 77–89.

Conforto, E., Salum, F., Amaral, D. C., da Silva, S. L., & de Almeida, L. F. M. (2019). The agility construct on project management theory. International Journal of Project Management, 37(6), 789–803. https://doi.org/10.1016/j.ijproman.2019.05.002

Fitzgerald, B., & Stol, K. J. (2020). Continuous software engineering: A roadmap and agenda. Journal of Systems and Software, 123, 176–189. https://doi.org/10.1016/j.jss.2020.110867

Hassan, A., & Rauf, A. (2021). Agile vs. Waterfall: A case study in software project management. Journal of Modern Project Management, 9(1), 55–62.

Hoda, R., & Murugesan, L. K. (2022). Multi-level Agile project management challenges: A people-centric perspective. Information and Software Technology, 137, 106618. https://doi.org/10.1016/j.infsof.2021.106618

Iivari, J., & Iivari, N. (2019). Organizational culture and the deployment of Agile methods: The competing values model. European Journal of Information Systems, 28(2), 218–235. https://doi.org/10.1080/0960085X.2018.1531720

Kumar, S., & Goyal, N. (2023). Project success in Agile development: Metrics and analysis. Journal of Information Systems, 39(3), 112–120.

Layton, M. C., & Ostermiller, S. (2020). Agile project management for dummies (3rd ed.). Wiley.

Misra, S. C., Kumar, V., & Kumar, U. (2021). Benefits of Agile methodologies in software development. ACM SIGSOFT Software Engineering Notes, 46(1), 34–40. https://doi.org/10.1145/3452166.3452175

Poppendieck, M., & Poppendieck, T. (2022). Lean software development: An Agile toolkit. Addison-Wesley.

Raymond, L., & Bergeron, F. (2008). Project management information systems: An empirical study of their impact on project managers and project success. *International Journal of Project Management*, *26*(2), 213–220. https://doi.org/10.1016/j.ijproman.2007.06.002

Royce, W. W. (2020). Managing the development of large software systems. IEEE Proceedings, 82(11), 328–346.

Schwaber, K., & Sutherland, J. (2023). The Scrum Guide: The definitive guide to Scrum. Scrum.org. https://scrumguides.org/

Sommerville, I. (2021). Software engineering (10th ed.). Pearson Education.

VersionOne. (2024). State of Agile report. https://stateofagile.com/