PROCEEDING ICEBESS - SEP 2024
E-ISSN : 2528 - 671X

SOFTWARE DEVELOPMENT USING AGILE APPROACH

Sri Susanna Sembiring
Fakultas Ekonomi dan Bisnis, Universitas Negeri Yogyakarta, Indonesia
(srisusanna.2023@student.uny.ac.id)

Mahendra Adhi Nugroho
Fakultas Ekonomi dan Bisnis, Universitas Negeri Yogyakarta, Indonesia
(mahendra@uny.ac.id)

ABSTRACT

Fast and efficient software development is crucial for many organizations in today's digital age. One approach that is gaining popularity is the agile method. This research aims to 1) analyze the application of the Agile approach in software development, 2) identify the main advantages and challenges of the Agile methodology, and 3) compare the effectiveness of Agile with traditional software development methods. The research method used was a literature review, in which relevant literature on Agile software development was analyzed. The research findings show that Agile approaches can improve development speed, flexibility in responding to changing requirements, and customer engagement. However, it also faces challenges, such as the need for cultural change and adequate training for the development team. A comparison with traditional methods shows that Agile is more effective in terms of adaptability, collaboration, and customer satisfaction, but it may require more discipline and coordination. This research provides insights for organizations considering the adoption of Agile in their software development process.

Keywords: Agile, software development, traditional methods, literature review

INTRODUCTION

Information technology's rapid development has changed how organizations develop software (Cao et al., 2019). In this dynamic digital era, the need for software development methodologies that are flexible, adaptive, and responsive to change is becoming increasingly urgent (Cao et al., 2019). Agile approaches emerge as a solution to overcome the limitations of traditional development methods, such as the Waterfall model, which often cannot accommodate changing needs quickly (Camara et al., 2020).

Agile is an iterative and incremental approach to software development that emphasizes flexibility, team collaboration, and close customer interaction. (Javdani Gandomani & Ziaei Nafchi, 2016).. The Agile principles, formulated in the Agile Manifesto in 2001, have shifted the software development paradigm from a process- and documentation-focused approach to one more concerned with people, interaction, and value creation for the customer (Dingsøyr & Lassøyr, 2016). (Dingsøyr & Lassenius, 2016).

Since its inception, various Agile methodologies such as Scrum, Extreme Programming (XP), Kanban, and Feature-Driven Development (FDD) have been developed and widely adopted in the software industry. According to the State of Agile survey by VersionOne in 2021, 95% of organizations surveyed reported using Agile practices in their software development.

Agile software development methodologies have been gaining acceptance among mainstream software developers since the late 1990s when they were first established in Scrum, Feature-driven development (FDD), Extreme Programming, and other methodologies. (Dingsøyr et al., 2012).. Software development approaches guide developers through the software development process (de Borba et al., 2019).. To develop software, developers usually

PROCEEDING ICEBESS - SEP 2024
E-ISSN : 2528 - 671X

choose a software development approach, which usually divides the development process into phases, and for each phase, developers must apply the guidelines provided by the chosen approach for that phase. (de Borba et al., 2019). Agile proponents claim that the focal aspects of light and agile methods are simplicity and speed (Davis et al., 1988). (Davis et al., 1988). Therefore, in development work, the development group concentrates only on required functions immediately, executes them quickly, collects feedback, and reacts quickly to business and technology changes. (Anggrian & Geni, 2024).. Next, we will discuss each of these Agile methods.

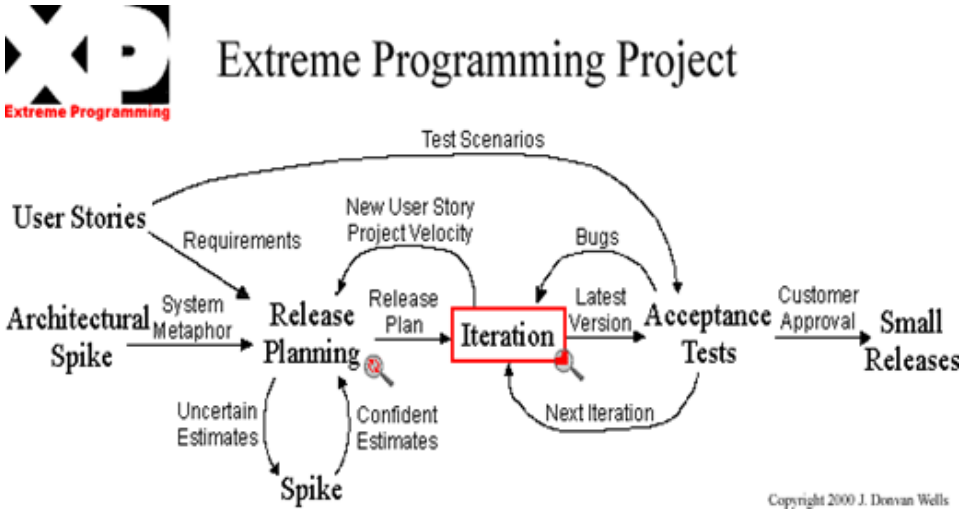
Scrum

Scrum is a framework for developing and maintaining complex products (Camara et al., 2020). Scrum applies an iterative and incremental approach focusing on collaboration, adaptability, and stakeholder engagement. Scrum focuses on short iterations, team collaboration, and adaptability to changing needs. There are three leading roles in Scrum: Product Owner: Responsible for maximizing the business value of the developed product (Conboy, 2009). Scrum Master: Responsible for ensuring the Scrum team can work effectively. Scrum Team: This team consists of multifunctional professionals who work together to produce "done" product increments for every sprint.

Extreme programming (XP)*:*

Extreme programming (XP) is a collection of well-known software engineering practices. Extreme programming aims to enable successful software development despite unclear or constantly changing software requirements. Extreme programming is a software development discipline that organizes people to produce high-quality software more productively. XP seeks to reduce the cost of changing requirements by having multiple short development cycles instead of long ones.

The third part of the manuscript, "Method, Data, and Analysis," is designed to describe the nature of the data. The method should be well elaborated and enhance the model, the approach to the analysis, and the steps taken. Equations should be numbered as we illustrate. This section typically has the following sub-sections: Sampling (a description of the target population, the research context, and units of analysis; the sample; and respondents' profiles); data collection; and measures (or measurements).

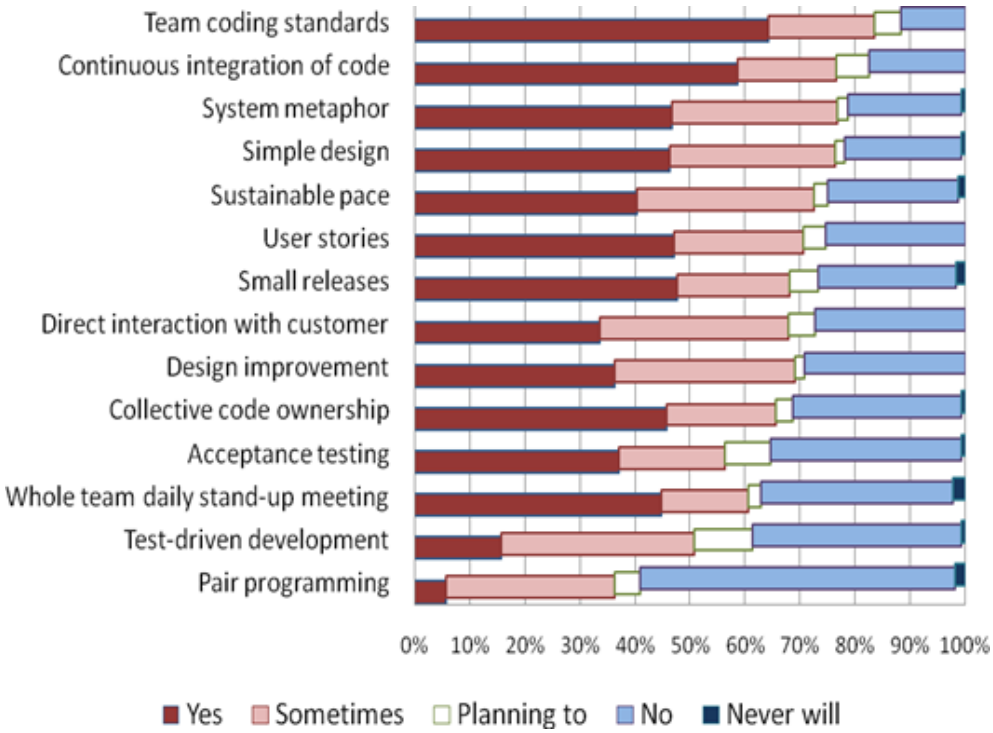


***Picture* 1. Extreme programming (XP) methodology**

Feature-driven development (FDD):

PROCEEDING ICEBESS - SEP 2024
E-ISSN : 2528 - 671X

Feature-based development *(*FDD) is a model-based short iteration process comprising five primary activities. For accurate state reporting and tracking of software development projects, milestones marking the progress achieved on each feature are defined. This section provides an overview of the activities. In the figure on the right, the meta-process model for this activity is shown. During the first two sequential activities, the shape of the overall model is established. The FDD approach embodies iterative development with practices believed to be effective in the industry. The specific mix of these ingredients makes the FDD process unique to each case. It emphasizes quality aspects throughout the process and includes frequent and tangible deliverables and accurate monitoring of project progress.



***Picture* 2 : Percentage Usage for Agile Practices**

However, Agile implementation is not without its challenges. Organizations often face difficulties changing work cultures, adjusting team structures, and managing stakeholder expectations. In addition, the effectiveness of Agile in various project and organizational contexts is still a subject of debate among practitioners and researchers.

This study thus aims to 1) analyze the application of Agile approaches in software development, 2) identify the main advantages and challenges of Agile methodologies, and 3) compare the effectiveness of Agile with traditional software development methods. Through a systematic literature review, this study is expected to provide valuable insights for organizations considering Agile adoption or seeking to optimize existing Agile implementations.

METHOD, DATA, AND ANALYSIS

This research uses a literature study method by collecting, reading, and analyzing various reliable reference sources related to software development with an agile approach. These sources include scientific journals, books, and articles from reputable websites. The data obtained was then used to answer the research objectives. From the reference search data conducted, 57 results were obtained from 2009-2023. However, researchers selected five

PROCEEDING ICEBESS - SEP 2024
E-ISSN : 2528 - 671X

references based on the title, problem formulation, objectives, and research results after the selection process.

RESULT AND DISCUSSION

Analyzing the Application of the Agile Approach in Software Development

In the study, Camara et al. (2020) showed that implementing Agile practices in global software development (GSD) teams involves various techniques such as daily meetings, collaboration through software, and communication tools that support distributed team interactions. These practices aim to increase team knowledge, identify obstacles, and plan the work. This research also emphasizes the vital role of the Scrum Master and team members in effectively adopting these practices in the GSD context. Furthermore, in their research study, Cao et al. (2009) stated that the application of Agile practices, such as pair programming and iterative development, is highly dependent on the organizational context and top management support. For example, the successful implementation of Agile methods at FinanApp was hampered after new management who did not support the method took over. The method suggests that the appropriateness of the appropriation and alignment with the organizational culture are critical to the success of Agile implementation. Mangalaraj et al. (2009) mention using methods to understand the factors that influence the acceptance of Extreme Programming (XP) in various project groups in the organization. It was found that XP acceptance involves modifying recommended practices to meet the specific needs of the project and organization. These modified processes are then accepted and institutionalized through frequent use. Fruhling & De Vreede (2006) showed that applying eXtreme Programming (XP) principles in developing an emergency response system was successfully carried out by adapting 12 core XP principles. Despite not having customers on-site, the development team implemented intensive communication and close collaboration with users. This approach enabled flexible and responsive development to changing needs, resulting in a system that operated well in the real environment. Wang et al. (2019) showed that the application of Agile approaches often involves a combination of lean principles. The analysis results show that although there are variations in the way Agile is implemented, many organizations have successfully adopted Agile practices by integrating lean elements to improve efficiency and reduce waste in the software development process.

Identifying Key Advantages and Challenges of Agile Methodologies

According to Camara et al. (2020) the main advantages of Agile methodologies in the context of GSD include increased flexibility, the ability to respond quickly to customer needs, and reduced development time thanks to better collaboration. However, challenges include communication, coordination, and control issues of the development process that can arise due to the different locations of teams. This research notes that although Agile is designed for closely located teams, many studies show that Agile practices can help overcome the problems associated with GSD. Cao et al. (2019) mentioned that the main advantages of Agile methodologies include the ability to adapt quickly to changing user needs and dynamic project environments. This method allows for faster development and frequent consultation with customers, which can increase user satisfaction. However, challenges include inadequate architectural planning, overemphasis on early deliverables, and low test coverage, leading to overlooked design issues in large projects. Mangalaraj et al. (2019) mentioned that the advantages of Agile methodologies, including XP, are their ability to increase flexibility and responsiveness to changing customer needs, as well as an emphasis on team collaboration and active customer involvement in the development process. However, the main challenge organizations face when adopting Agile methodologies is the significant burden posed by

PROCEEDING ICEBESS - SEP 2024
E-ISSN : 2528 - 671X

changing the work habits established by traditional approaches and variations in the pattern of adherence to XP practices within the organization. Fruhling & De Vreede (2019) highlight the main advantages of Agile methodologies, notably XP, which include flexibility in development, the ability to perform rapid prototyping, and reduced overhead. However, challenges include the need for constant communication and user engagement and potential difficulties in managing the expectations of various stakeholders with different needs. The research also noted that although the principles were not fully adopted, user engagement was still achieved through frequent meetings and ongoing communication. Wang et al. 2019) show advantages of Agile methodologies include improved team collaboration, responsiveness to change, and faster product delivery. However, challenges include difficulties in adapting Agile principles to different organizational contexts and issues related to the quality of data in experience reports used for analysis. The research also notes that the terms "Agile" and "lean" are often unclear, which can lead to confusion in their application.

Comparing the Effectiveness of Agile with Traditional Software Development Methods

Agile is more effective than traditional software development methods in the context of GSD (Camara et al. 2020). Agile allows teams to adapt quickly to changes and provide faster customer feedback, an essential advantage in a dynamic business environment. While traditional methods tend to be more rigid and focus on heavier documentation, Agile emphasizes team interaction and periodic delivery of working software, which increases customer satisfaction and development efficiency. Cao et al. (2019) mentioned that Agile methods are more effective in projects requiring flexibility and rapid adaptation, especially in dynamic contexts. However, traditional methods may be more appropriate for large and complex projects as they usually involve more formal planning and more structured architectural design. Research shows that while Agile can provide advantages in speed and responsiveness, scalability and architectural planning, challenges can reduce its effectiveness compared to traditional methods in specific contexts. Mangalaraj et al. (2019) mentioned that Agile methodologies, such as XP, emphasize software development's social and collaborative aspects, in contrast to traditional methods that focus more on planning and control. Research shows that Agile can be more effective in creating value through rapid adaptation to change. However, its successful implementation is highly dependent on the characteristics of the team and organizational environment. As such, Agile offers a more dynamic approach than traditional methods, which tend to be rigid. Fruhling & De Vreede (2019) mentioned that Agile methods, such as XP, are more effective in handling changing requirements and delivering faster results than traditional plan-driven software development methods. Traditional methods often have difficulty managing change and tend to exceed budget and schedule. In the context of this project, XP proved to be more suitable for small teams and projects that require rapid iteration. At the same time, traditional methods are more suitable for projects with stable and predictable requirements. Wang et al. (2019) show that Agile, when implemented well, can be more effective than traditional software development methods. Agile allows teams to adapt quickly to changing needs and provide faster feedback to stakeholders. In contrast, traditional methods are often more rigid and less responsive to change, which can result in delays in product delivery and an increased risk of project failure.

CONCLUSIONS

This literature review shows that applying Agile methods in software development has various advantages, such as flexibility, quick adaptability, and increased team collaboration. However, challenges such as communication difficulties in distributed teams, adaptation to different organizational contexts, and the need for intensive communication are also faced.

Compared to traditional methods, Agile is more effective in dynamic environments that require rapid change, although traditional methods may be more appropriate for large projects with more formalized planning. Successful implementation of Agile depends mainly on management support and fits the organization's culture.

REFERENCE

Anggrian, S., & Geni, B. Y. (2024). Kepegawaian Menggunakan Metode Waterfall (Studi Kasus : Pt . Dola Usaha Indonesia). *Jurnal Mahasiswa Teknik Informatika*, *8*(1), 1029–1035.

Camara, R., Alves, A., Monte, I., & Marinho, M. (2020). Agile Global Software Development: A Systematic Literature Review. *ACM International Conference Proceeding Series*, 31–40. https://doi.org/10.1145/3422392.3422411

Cao, L., Mohan, K., Xu, P., & Ramesh, B. (2019). A Framework for Adapting Agile Development Methodologies. *European Journal of Information Systems*, *18*(4), 332–343. https://doi.org/10.1057/ejis.2009.26

Conboy, K. (2009). Agility From First Principles: Reconstructing the Concept of Agility in Information Systems Development. *Information Systems Research*, *20*(3), 329–354. https://doi.org/10.1287/isre.1090.0236

Davis, A. M., Bersoff, E. H., & Comer, E. R. (1988). A Strategy for Comparing Alternative Software Development Life Cycle Models. *IEEE Transactions on Software Engineering*, *14*(10), 1453–1461. https://doi.org/10.1109/32.6190

de Borba, J. C. R., Trabasso, L. G., & Pessôa, M. V. P. (2019). Agile Management in Product Development. *Research Technology Management*, *62*(5), 63–67. https://doi.org/10.1080/08956308.2019.1638488

Dingsøyr, T., & Lassenius, C. (2016). Emerging Themes in Agile Software Development: Introduction to the Special Section on Continuous Value Delivery. *Information and Software Technology*, *77*, 56–60. https://doi.org/10.1016/j.infsof.2016.04.018

Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, *85*(6), 1213–1221. https://doi.org/10.1016/j.jss.2012.02.033

Fruhling, A., & De Vreede, G. J. (2019). Field Experiences With EXtreme Programming: Developing an Emergency Response System. *Journal of Management Information Systems*, *22*(4), 39–68. https://doi.org/10.2753/MIS0742-1222220403

Javdani Gandomani, T., & Ziaei Nafchi, M. (2016). Agile Transition and Adoption Human-Related Challenges and Issues: A Grounded Theory approach. *Computers in Human Behavior*, *62*, 257–266. https://doi.org/10.1016/j.chb.2016.04.009

Mangalaraj, G., Mahapatra, R., & Nerur, S. (2019). Acceptance of Software Process Innovations- The Case of Extreme Programming. *European Journal of Information Systems*, *18*(4), 344–354. https://doi.org/10.1057/ejis.2009.23

Wang, X., Conboy, K., & Cawley, O. (2019). "Leagile" Software Development: An Experience Report Analysis of the Application of Lean Approaches in Agile Software Development. *Journal of Systems and Software*, *85*(6), 1287–1299. https://doi.org/10.1016/j.jss.2012.01.061