# **Research and Practice of Agile Software Development Methods**

#### Zhaoda Zhen

Taiyuan University of Technology, Taiyuan, Shanxi Province, China

zhenzhaoda5077@link.tyut.edu.cn

Abstract. In the rapidly changing software development environment, agile methods have gradually become the key to improving project efficiency and quality. This study arises from the challenge posed by traditional development models, which struggle to adapt to swiftly changing market demands. It focuses on the theme of agile software development and aims to explore its application advantages in practical projects. Through literature review and comparative analysis, this research delves into the core principles and practical applications of agile methods. The research results show that agile methods can significantly improve the response speed and delivery quality of development teams, verifying their importance in the context of the new era. In short, agile development is the preferred strategy for addressing complex and ever-changing requirements, and its flexibility and efficiency have brought revolutionary changes to the software development industry.

Keywords: agile software, software development, demand management, Iterative development.

#### 1. Introduction

In recent years, agile software development methods have received widespread attention and research in both academic and industrial circles both domestically and internationally. Numerous scholars and experts have engaged in comprehensive theoretical discussions and practical analyses regarding agile methods, contributing valuable insights and recommendations.

In the industrial sector, an increasing number of companies are adopting agile methods for software development, achieving notable commercial success. These enterprises have not only improved the efficiency and quality of software development by implementing agile methods, but also enhanced team cohesion and innovation capabilities. At the same time, agile methods provide organizations with greater flexibility and rapid market response capabilities, which helps them gain an advantage in fierce market competition.

The research on agile software development methods has benefited from significant contributions by prior scholars. For example, Barney Sebastian et al. analyzed the shortcomings of traditional software development in 2003 and pointed out the necessity of introducing agile patterns. Dusan Bozic[1] and Milos Panto applied agile development methods to software development in the power system in 2010, significantly improving the efficiency of traditional industry software development .. However, despite these foundational theoretical contributions and practical case studies, gaps remain in the in-depth application of agile software development methods in specific fields such as finance and healthcare, and the integration of agile methods with emerging technologies such as big data and artificial intelligence,

© 2024 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

which need further exploration and improvement. Therefore, this article provides an overview of the characteristics and current application status of agile software, as well as suggestions for its future development direction. This paper aims to provide a solid theoretical foundation for the advancement and utilization of agile software.

# 2. Overview and Characteristics of Agile Software Development

# 2.1. Definition and Background

Agile software development methodology is a programming approach that emphasizes rapid and iterative development. This method originated from an improvement on the traditional waterfall development approach, especially in environments with uncertain and frequently changing requirements [2]. Agile methods break down large software development tasks into smaller, more manageable parts through short iteration cycles, thereby improving the ability to respond to changes. It is an effective means of providing value to customers quickly, efficiently, and consistently throughout the software development lifecycle (SDLC).

# 2.2. Main features

The main characteristics of agile software development methods include short cycle iterations, high-frequency delivery, lightweight branching and merging strategies, automated testing, continuous integration, and continuous deployment. This approach prioritizes human collaboration, emphasizing close collaboration and face-to-face communication between the programmer team and business experts. Compared to traditional waterfall development, agile methods are more flexible, able to quickly adapt to changes in requirements, and improve software quality through frequent deliveries. In addition, agile methods also emphasize simple design and development principles to minimize complexity and improve maintainability [3].

## 2.3. Common frameworks

In agile software development, there are various common frameworks and methods, such as Scrum, Kanban, Extreme Programming (XP), Lean Software Development, etc. Scrum is an iterative and incremental framework that emphasizes the team delivering a set of potential deliverables in each iteration. The Kanban method uses visual Kanban cards to manage workflow, improve team focus, and reduce communication barriers. Extreme programming emphasizes communication, simplicity, feedback, and courage, thereby lowering the costs associated with requirement changes through active dialogue and feedback. Lean software development emphasizes eliminating waste and maximizing customer value.

In summary, agile software development methods have become an indispensable part of modern software development due to their rapid response to changes, frequent delivery, and emphasis on teamwork.

# 3. Application of Agile Software Development in Software Development

This chapter will introduce the concepts of team management, requirement management, development process, and risk management, and explain the current application status of agile software.

# 3.1. Team Management

The core of agile software development lies in efficient collaboration and rapid response within the team. In terms of team management, agile methods emphasize a "small but refined" team composition, where each team typically includes key roles such as developers, testers, product managers, and designers, ensuring smooth communication and quick decision-making within the team [4]. Agile teams emphasize the principle of "self-organization" and encourage team members to take on responsibilities proactively and choose tasks based on their respective professional skills and interests, thereby improving work enthusiasm and efficiency.

In order to maintain the continuous and efficient operation of the team, agile methods also advocate regular team review meetings, allowing team members to collectively reflect on their work performance over the past period, identify existing problems, and propose improvement measures. This mechanism of continuous self-optimization helps the team enhance its collaborative capabilities and development efficiency. In addition, agile teams also focus on face-to-face communication, ensuring that each member is aware of the current task progress, obstacles encountered, and next steps through regular stand up meetings (Daily Scrum), thereby enhancing team transparency and cohesion.

## 3.2. Requirement Management

In terms of requirements management, agile software development adopts a flexible and iterative approach. Compared to the traditional waterfall model, agile methods no longer rely on detailed pre planning, but encourage close collaboration between users and development teams to gradually clarify and refine requirements through short iterations (such as Sprint) [5]. During each iteration cycle, users can provide feedback, and the development team adjusts and optimizes product functionality based on this feedback.

To facilitate this process, agile methods introduce concepts such as User Story and Backlog. User stories describe the functions or problems that users hope to achieve or solve in concise and clear language, while a requirement pool is a dynamically updated list that includes all user stories to be implemented and prioritized. Through regular Refinement Meetings, the team and users work together to refine the stories in the requirements pool, clarify acceptance criteria, and lay the foundation for subsequent development work.

# 3.3. Development Process

The process of agile software development revolves around iteration and increment, with each iteration cycle (such as a two-week sprint) consisting of four stages: planning, development, testing, and review. In the planning phase, the team selects a set of user stories for development based on the priority in the demand pool. During the development phase, team members divide their work and collaborate in a self-organizing manner to jointly promote the implementation of functionality. During the testing phase, the testing team conducts detailed testing on the developed features to ensure product quality [6]. Finally, in the review phase, the team presents the iterative results to users, collects feedback, and adjusts subsequent plans based on the feedback.

In order to maintain continuity and traceability in development, agile methods also emphasize the importance of Continuous Integration and Continuous Deployment. By employing automated build, test, and deployment processes, teams can promptly identify and resolve issues, ensuring that the software remains in a deliverable state.

## 3.4. Risk Management

In terms of risk management, agile software development adopts a forward-looking and adaptive strategy. The team identifies potential risks and challenges, such as technical difficulties, resource shortages, or changes in requirements, through regular iterative reviews and retrospective meetings. In response to these risks, the team can develop corresponding coping strategies, such as adjusting development plans, seeking external resources, or introducing new technologies.

In addition, agile methods encourage teams to maintain an open attitude towards change and respond quickly to market changes and user feedback through flexible iteration cycles and iteration plans. This adaptability allows the team to quickly realign its direction and mitigate losses when risks arise. At the same time, agile methods also emphasize communication and collaboration between teams, reducing the risks of information asymmetry through information sharing and shared decision-making.

In summary, agile software development has demonstrated unique advantages in team management, requirements management, development processes, and risk management. Through flexible iteration, rapid response, and continuous improvement, agile methods not only improve the efficiency and quality of software development but also bolster team cohesion and innovation capabilities.

## 4. Challenges and Countermeasures of Agile Software Development Methods

#### 4.1. Challenge Analysis

Agile software development methods face multiple challenges. Firstly, frequent requirement changes may lead to development delays and resource waste [7]. Secondly, the typically limited emphasis on documentation in agile development increases the likelihood of errors, thereby placing additional pressure on the quality assurance team [8]. In addition, as new features are rapidly iterated, testing teams often face time constraints, making it difficult to comprehensively assess whether these features meet the established requirements. Meanwhile, communication and collaboration among team members are also a major challenge, as misunderstandings and conflicts may affect project progress.

#### 4.2. Response strategies and future outlook

To address the above challenges, the team should actively communicate with clients, clarify requirements, and ensure timely adjustments and adaptation to changes during the development process. Establishing a robust requirements management mechanism is crucial, encompassing requirement prioritization, change control, and documentation. In terms of quality assurance, it is essential to establish a comprehensive testing process, including unit testing, integration testing, and user acceptance testing. In addition, teams should leverage collaboration tools and communication platforms, such as version control systems and project management tools, hold regular team meetings, and improve collaboration efficiency.

Famous companies such as Amazon, Google Maps, and Spotify have adopted agile development methods to continuously optimize and improve their products through iterative development, rapid deployment, and user feedback. These enterprises have successfully launched numerous innovative products and services, improved user experience, and achieved commercial success by utilizing the rapid response to changes in agile development.

With the development of new technologies such as artificial intelligence, cloud computing, and the Internet of Things, the agile development industry will usher in more opportunities. The acceleration of enterprise digital transformation will further stimulate growth in the agile development market. Additionally, increased government support for the software industry and innovation has created a favorable policy environment for agile development. In the future, agile development will place greater emphasis on technological innovation and pattern upgrading to meet the enterprise's pursuit of software development efficiency and quality. At the same time, companies in the industry need to strengthen cooperation, jointly respond to market challenges and opportunities, and promote the sustainable and healthy development of the agile development industry.

## 5. Conclusion

This study discusses the overview and characteristics of agile software development methods, emphasizing their core concepts such as user centricity, iterative incremental development, and flexible response to changes. Through case analysis, it was found that agile methods can significantly enhance the efficiency and quality of software development, foster team collaboration, and effectively address complex and evolving project requirements. However, agile software development also faces many challenges in practical applications, such as frequent changes in requirements and obstacles to team collaboration. Corresponding countermeasures have been proposed to address these challenges. Although this article has achieved certain results in applied research in four fields, there are still shortcomings, particularly the relatively small research sample. Future research should further expand the sample size to cover more industries and project types, in order to obtain more comprehensive and accurate research conclusions, and provide more solid theoretical and practical support for the promotion and optimization of agile software development methods. Through continuous exploration and innovation, agile software development methods are expected to play a more important role in the field of software development.

#### References

- [1] Bozic, D., & Panto, M. (2010). Agile development methods in the power system software industry. IEEE Transactions on Power Systems, 25(4), 1922-1930.
- [2] Beck, K. (2001). Extreme programming explained: Embrace change. Addison-Wesley Longman Publishing Co., Inc.
- [3] Highsmith, J. (2009). Agile project management: Creating innovative products. Pearson Education.
- [4] Kniberg, H. (2010). Scrum and XP from the Trenches. "C4Media".
- [5] Cohn, M. (2004). User Stories Applied: For Agile Software Development. Addison-Wesley Professional.
- [6] Highsmith, J. (2002). Agile Software Development Ecosystems. Addison-Wesley Longman Publishing Co., Inc.
- [7] Highsmith, J. (2001). Agile software development ecosystems. Computer, 34(1),26-33.
- [8] Ambler, S. W., & Shull, F. (2012). The role of documentation in agile software development: A case study of four agile teams. Journal of Systems and Software, 85(3), 584-594.