

Title:

Comparison of Agile and Waterfall Software Development Methodologies

Module:

Introduction to Software Engineering

Student Name:

Mariam Al-Obaidly

Student ID:

12702766

Introduction

Software development methodologies are structured approaches used to organize and manage the software development process to deliver high-quality software that meets user requirements within the planned time and budget. These methodologies provide a framework that guides development teams through different stages of software creation and project management (Stephens, 2015). This is important in software project management because it helps teams organize development activities, manage resources effectively, and reduce project risks. This report compares two of the most widely used software development methodologies, Waterfall and Agile, based on their flexibility, suitability for different project sizes, and approaches to risk management in order to evaluate which methodology is more appropriate for different project environments.

Agile Methodology

The Agile methodology is a flexible approach used in software development that focuses on continuous improvement and iterative development. Instead of completing the entire project in a single sequence, the work is divided into small cycles called sprints. During each sprint, specific features are designed,

developed, and tested within a short time. This method encourages regular communication and feedback between developers and users, allowing teams to respond quickly to changes in requirements. Therefore, Agile is often suitable for projects where requirements may change during development, such as mobile applications or startup projects. However, frequent changes in requirements can sometimes make planning and documentation more challenging (Beck et al., 2001).

Waterfall Methodology

The Waterfall model is a traditional software development methodology that follows a linear and sequential process, where each stage must be completed before moving to the next. The process usually includes requirements analysis, system design, implementation, testing, and maintenance. This structured process allows development teams to plan their work more effectively and maintain a predictable project workflow. It is often suitable for projects where requirements are clearly defined and unlikely to change. However, because the process is sequential, it can be difficult to adapt to changes during development, and some problems may only appear in the later stages of the project (Royce, 1970).

Comparison of Agile and Waterfall

Agile and Waterfall differ mainly in flexibility, project size suitability, and risk management. The Waterfall model follows a fixed and sequential process, which makes it less flexible when requirements change during development (Royce, 1970). For example, large banking systems or government platforms often use the Waterfall model because their requirements are clearly defined and must follow strict regulations. In contrast, Agile allows teams to adapt to changes more easily because development is carried out in short iterations and continuous feedback

(Beck et al., 2001). For instance, many mobile applications and startup products adopt Agile practices because they require frequent updates and user feedback. Regarding risk management and quality assurance, Waterfall usually performs testing in later stages of development, while Agile incorporates continuous testing throughout the development process, helping teams identify and resolve issues earlier.

Conclusion

In conclusion, both Waterfall and Agile are important software development methodologies, but each one is suitable for different types of projects. Waterfall works better for projects with stable and clearly defined requirements, while Agile is more suitable for projects that require flexibility and continuous updates. Therefore, choosing the appropriate methodology depends on the nature of the project and its requirements.

References:

Beck, K. et al. (2001) *Manifesto for Agile Software Development*. Available at: <https://agilemanifesto.org> (Accessed: 13 March 2026).

Royce, W. (1970) *Managing the Development of Large Software Systems*. Proceedings of IEEE WESCON, pp. 1–9.

Stephens, R. (2015) *Beginning Software Engineering*. 2nd edn. Indianapolis: Wiley.