Traditional vs. Agile Methodologies

Jacob Colburn

University of Arizona Global Campus

CST499: Capstone for Computer Software Technology

Instructor: Prof. Joseph Rangitsch

27JUL24

**Introduction**

Software development has changed drastically over the years. The traditional approach to development, such as the waterfall and spiral models, is still used but cannot keep up with the newly constantly changing requirements of dynamic modern projects. In response to this, agile methodologies like extreme programming (XP), scrum, and kanban have been used more frequently. These new methods allow for more flexibility and adaptability on the frameworks to better work with these challenges (Tsui, F., Karam, O., & Bernal, B., 2018). This paper will explore how these newly emerging methodologies address the current challenges in software development more effectively than the older traditional approaches and identify the limitations of traditional methodologies.

**Agile in Software Development**

Agile methodologies are more adaptive and responsive than the traditional approach, making them better suited for ever-changing requirements in a fast-paced environment. They are more flexible as well, which ensures adaption and improvement can be made throughout the development process. Collaboration with customers and stakeholders is another big key point for agile, ensuring the product aligns with business and customer expectations by incorporating their feedback into development (Rigby, D., Sutherland, J., Takeuchi, H., 2016). Unlike traditional methodologies, agile is incremental and aims to deliver a functional product increment on a regular basis. Daily meetings are held, and pair programming performed to improve team communication, reduce copied work, and foster a feeling of ownership among the team. Finally risk management is addressed in agile by breaking down the project into smaller more manageable parts to better identify potential issues saving on costs in the future (2016).

**Limitations of Traditional Methodology**

Traditional methods like waterfall and spiral methods unlike agile, struggle with the complex nature of modern software development since they are a more rigid and linear approach. The traditional methods are a more sequential approach, and once a step is completed such as requirements gathering, it is hard to go back and make changes (Stoica, M., 2013). This approach can often result in bigger problems later if the requirements were not fully understood at the time of requirements gathering. Testing in traditional approaches is performed later in the development cycle leading to critical problems being missed early on and contributing to a more time consuming and costly repair process. Traditional approaches do not hold much weight on collaboration through the process which could result in a product that does not meet the expectations of customers and shareholders (2013. Finally unlike agile, traditional methods are more resource heavy requiring specialized experts to complete risk assessments and requirements events leading to higher cost and complexity for the project.

**Conclusion**

In conclusion, while the traditional approaches to software development have been proven to be a solid foundation for frameworks that manage software projects, their limitations in their flexibility, collaboration, and risk managements make them less effective in more complex dynamic projects. While there are specific situations where this linear approach would be preferred, in these bigger complex projects agile is the recommended approach. Agile approaches emphasize adaptability, flexibility, collaboration, communication, and incremental deliveries offer a better approach toward the new challenges of software development. Fully understanding these benefits help developers better navigate the complex projects being built today by delivering high quality products that better align with the business and stakeholders' goals.

**References**

Stoica, M., (2013). “Software Development: Agile vs. Traditional”. DOI:[10.12948/issn14531305/17.4.2013.06](http://dx.doi.org/10.12948/issn14531305/17.4.2013.06)

Rigby, D., Sutherland, J., Takeuchi, H., (2016). “Embracing Agile”. Harvard Business Review. [**https://hbr.org/2016/05/embracing-agile**](https://hbr.org/2016/05/embracing-agile)

Tsui, F., Karam, O., & Bernal, B. (2018). *Essentials of software engineering* (4th ed.). Jones & Bartlett Learning. [Course Text](https://platform.virdocs.com/read/2348054/11/#/4/2[ch06]/6/4[ch06-sect1-002]/10/4/2)