Week02 – Short Paper Assignment – AGILE

SWEN 603 9041

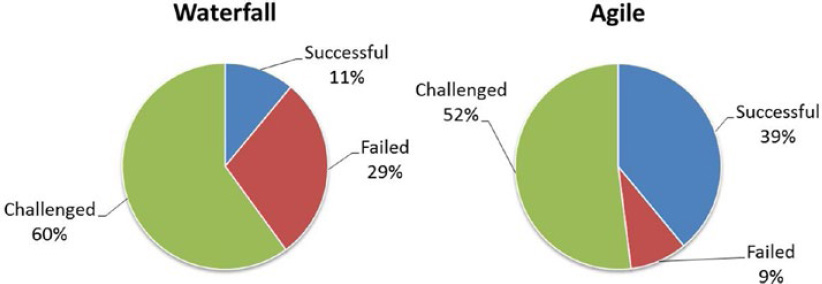
Debashis Jena

University of Maryland Global Campus

# Introduction

Before the agile methodology came into the picture, the traditional waterfall model was the one which was the most popular model and widely used process to build software. However, as we observed the waterfall model has many shortcomings. The biggest drawback being the length of the projects. With requirement gathering, documentation, planning, design, and other formal phases, a midsized project may take a few months to a year to get the first release of the application. Additionally, if during the acceptance or testing phase of the project, the design is found to be wrong, then the success rate of the project becomes alarmingly low. The cost of the entire project may get doubled. To address each of these issues, the iterative or incremental method came into the picture.

The iterative methods can be defined as an approach to building applications in which the complete lifecycle consists of multiple iterations in sequence. The goal of each iteration is to release a working, integrated and stable application that may be partially complete. Each of the subsequent iterations, take feedback from the previous one and implement on top of that.

****

The complete system undergoes reviews and validations in each iteration, thus preventing unfortunate surprises in the system in the later phases. RAD and Waterfall models are two of the variations of the iterative development process, with two different methodologies and objectives.

# Agile Methodology

This method is guided by the agile manifesto, which mainly states the following.

* *Individuals and interactions over processes and tools* – In contrast to the waterfall model, the Agile method values interactions more than the traditional processes. Agile encourages more face to face interaction than remote because it is the most efficient and effective way of communication to convey information to or within the development team. This is demonstrated in agile prices like scrum and extreme programming.
* *Working software over comprehensive documentation* – In the case of the waterfall model, sufficient documentation of each activity, process, and subsystem is created and maintained throughout the lifecycle. However, in agile the documentation is relatively less valued. The more important objective of the entire process is, the working software. In the case of the waterfall model, requirement documents are created in the requirement analysis phase which itself takes 20% to 30% of the entire time. Whereas in the agile method, the system requirements are split into smaller stories which can be accomplished within one sprint.
* *Customer collaboration over contract negotiation* – Another major time-consuming part of the waterfall model is contract negotiation, review and signoff/approval. Usually, the review and signoff process depends on the availability of the business people. Sometimes, this may cause a major delay to start the project. However, in agile the business users become part of the agile team. Continuous and frequent face to face meetings occurs between the development team and the customer. No formal signoff is required for starting the project. This makes the process kickoff instantly rather than waiting for approval from the customer.
* *Responding to change over following a plan* – Agile method is built upon the principle of accommodating change in the requirements promptly. In the case of the waterfall model, the requirements freeze after the signoff process. There is hardly any scope or time for accommodating a change. Therefore the requirement documents are reviewed carefully, which consumes time. Also, if something is missed during the analysis phase or if there is a business process change, then to accommodate it, the requirement change undergoes the same formal process starting from analysis to testing. However, in agile, the requirements are written as short stories and typically they are implemented just-in-time. So if there is a change to the business process or rule, it can be accommodated easily. That makes the method ever ready for responding to a change.

The project that follows agile, follows these guidelines strictly. As discussed above, this method is more result-oriented than the process.

**Specific Agile Methods –**

There are many agile methods, however, the below methods are the most popular ones and widely implemented.

* Scrum – The Scrum method is used over 50% of the usage of all the agile method implementors. This involves daily standup meetings (aka scrum meetings), usually 2 to 4 weeks iterations, lean and small team, demo to the external customer at the end of each iteration. Short and independent stories are created and then arranged as per the priority. Typically the stories move from one phase to another (also known as swimlanes in JIRA software).

A screenshot of a cell phone

Description automatically generated

* XP (Extreme programming) – This is commonly based on collaboration, fast and early most viable product creation, and efficient development practices. It values communication, simplicity, feedback, and courage.

A screenshot of a cell phone

Description automatically generated

XP process focuses more on software creation activity than anything else. Therefore, to gain the highest productivity the whole team works together in a common room, may involve in some pair programming, constant refactoring of the code and driven by predefined test scenarios.

* Kanban – As compared with the scrum method, the Kanban is more of a continuous flow than being a fixed length activity. Software implementation and delivery happen continuously, not at the end of the iteration. This method responds to changes more frequently than the scrum method.

# Drawbacks of Agile Methods

So far we have discussed the advantages of agile methodology over the traditional waterfall model. However, this method comes with some disadvantages too. Here are some of them described in brief.

1. Limited documentation – Since the working software is valued more than comprehensive documentation, sometimes the documentation is highly neglected. Lack of documentation becomes an issue later for any future reference. Sometimes the information regarding a specific implementation is completely omitted and causes confusion.
2. Inadequate resource planning – Typically the projects that run using an agile method, do not have a finite end. Therefore, it becomes impossible to estimate the project cost and time required. This issue becomes more prominent as the project grows bigger and complicated.
3. Fragmented output – The development teams that follow the agile method are usually small teams. So when they work on a big project, multiple teams are working on different components. With insufficient documentation and sometimes distributed teams, the output may become fragmented and may miss links with each other.

**Application of Agile Methodology**

Below are some of the candidates, based on which a team may adhere to the agile methodology for the software development process.

1. For a brand new software system which is automating some of the repetitive and manual efforts. These kinds of applications are created instantly with the basic needs of the users and gradually the new features are added based on the need of the end-users.
2. A system that is replacing an old system, which can gradually transition to the new system and does not necessarily require a big bang migration. In such cases, agile can be used to create the new system and slowly features can be moved from the legacy system to the new one.
3. Typically the production support and maintenance teams follow the Kanban version of the agile method. As the tickets are logged, the team members pick the items promptly and work on them one by one.

# References

1. Bibik, Ilya. (2018). *How to kill the scrum monster: quick start to agile scrum methodology and the scrum master role*. Retrieved from *http://library.books24x7.com.ezproxy.umuc.edu/toc.aspx?bookid=142618*
2. Larman, C. (2003). Agile and iterative development: a manager's guide. Retrieved from *https://learning.oreilly.com/library/view/agile-and-iterative/0131111558/ch03.html#*
3. No Author. (n.d.). What are the disadvantages of agile? Retrieved *from https://leankit.com/learn/agile/what-are-the-disadvantages-of-agile/*