Student ID: 180911

Name: O’shara Mason

Course: SWEN Software Project Management

Instructor: Professor Thomas Xu Canhuo

Date: 7th October 2019

Question 3

The research article “A Comparative Study of the Different Software Development Life Cycle Models in Different Scenarios” written by Aproova Mishra and Deepty Dubey discusses the processes employed by to ensure a project is completed within the its predefined time constraints and to enforce industry standards. Such methods are collectively known as the software development life cycle. According to the study, all life cycles have their strengths and weakness, therefore, the choice of which model to use is left to the discretion of the developer. However, a specific model should only be selected based on how its functionalities relate to that specific project.

All software development life cycles are comprised of the following phases:

- Requirements acquisitioning
- Planning and design
- Implementation
- Testing
- Deployment maintenance

In larger projects, the phases of the life cycle are further broken down into the following:

- System study
- Feasibility study
- System analysis
- Design
- Coding
- Testing
- Implementation
- Maintenance

Mishra and Dubey described five software development life cycle models and the situations where they would be most applicable in. The Waterfall model establishes a clear and manageable method that ensures the developer has consistent control of the project. It best suited for small projects because each step in the life cycle is predefined and once the plan is defined it is very hard to change after the project has begun. It is a low cost, high risk model which is recommended for highly skilled developers. The Incremental model initiates with planning phase, displays the relationships between the intermediate phases in a cycle and concludes with the deployment phase. Another low-cost, high expertise model, however the risks are easy to manage because each risk is resolved during its cycle. For large projects which require highly skilled developers, Mishra and Dubey suggested the Spiral model, it has comprehensive risk analysis that results in a small amount of risks involved. This model is very expensive, requires a large amount of end user involvement and the development process is extensive. Rapid application development was the final model discussed; it is a very flexible model used in the fast development of “functional systems”.

On the other hand, the study "Generalizing Agile conducted by Bhalero, Puntambekar and Ingle indicates that traditional software development life cycle models for example, the models previously discussed, although useful in some cases, have numerous disadvantages. For this reason, agile models have become increasing popular in the software development industry. Bhalero et al. proposes using the Agile Software Development Life Cycle which was conceptualised and created to incorporate the general methods and standards of all pre-existing agile methods. Agile development is said to a combination of all of the best prices in the industry, therefore by using these methods, projects should display more concise documentation, greater flexibility and it improves both the internal and external quality of the end project.

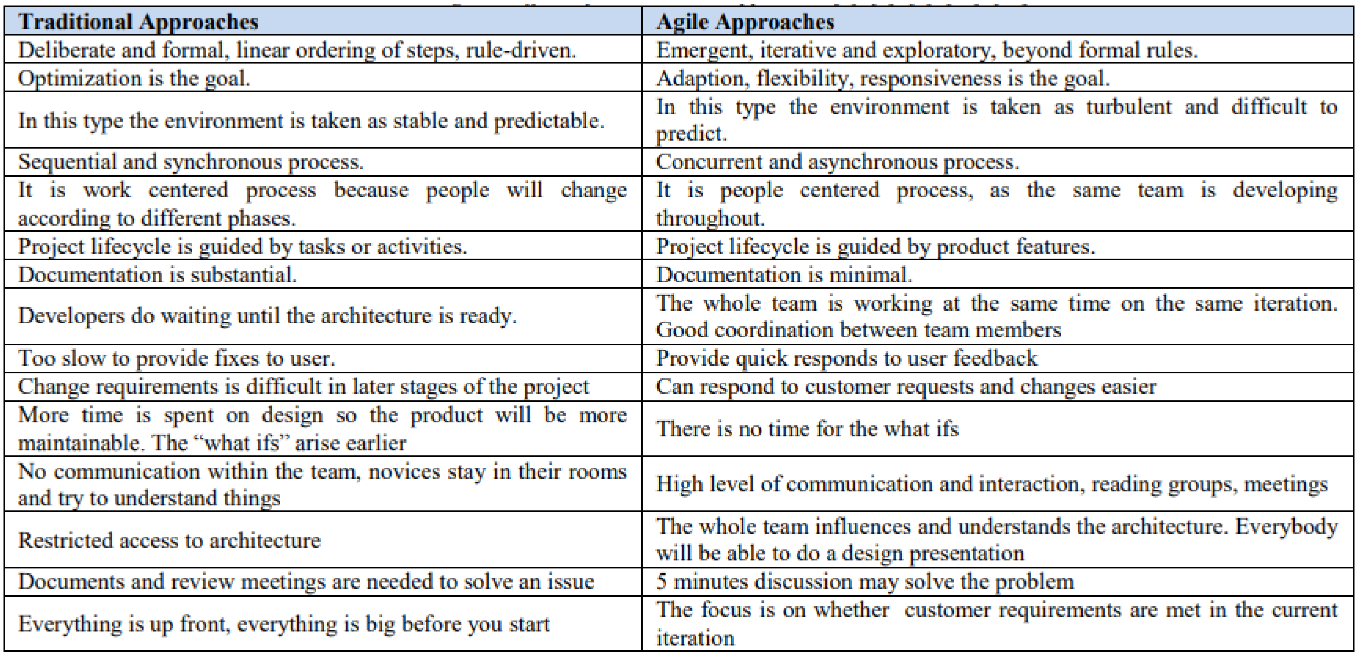
Similarly, www.agilealliance.org, explains Agile Software Development to be a collection of "frameworks and practices based on the values and principles expressed in the Manifesto for Agile Software Development and the 12 Principle behind it". Unlike the models discussed in the study by Mishra and Dubey, this model allows the various sections of the development team to incorporate numerous strategies in order to reach the project's goal.

Question 7

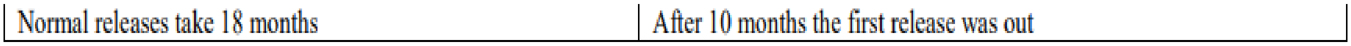
According to the research paper, “Review of Agile Software Development Methodologies” by Kaushal Pathak and Anju Saha, “agility, for a software development organisation, is the power of software to choose and react expeditiously and fittingly to various changes… An agile process is one that readily embraces and supports this degree of flexibility”. Before the introduction of agile methods to the software development industries, Traditional Software Development Methods such as the Incremental, Waterfall and Spiral methods were regularly use. The study, “Obstacles in Moving to Agile Software Development Methods; At a Glance” by Javdani et al. indicated that the introduction of Agile was initially met was scepticism. However, its popularity has gradually increased, it is not only used in project development but the methods have been integrated into the functionality of companies. “These methods have gained higher edge on traditional development by accommodating frequently changing requirements in high tight schedules” (Bhalerao et al.). This study also suggests that implementation of Agile methods have faced opposition and the reason persons experience difficulty with it is due to inexperience and lack of training in Agile software development. Even though such challenges have occurred, agile methods have been proven to reduce development challenges and time.

The 12 principles of Agile software development were created to ensure customer satisfaction by delivering software which fulfils industry principles and standards. Agile methods encourage simple, flexible, interactive approaches which can be applied to all types of software projects. Agile development in terms of companies and organisations allows all team members to be creative and efficient. In order to have the high standard product, the development team must first understand the basis of the principles then choose the best suited agile method for that specific project. Incorrectly selecting a method results in higher costs, longer life cycles and possible failure (Javdani et al), feature which the Agile movement was put into effect in order to reduce.

The table below by Pathak and Saha compares Agile and traditional approaches:



Traditional Approaches	Agile Approaches
Deliberate and formal, linear ordering of steps, rule-driven.	Emergent, iterative and exploratory, beyond formal rules.
Optimization is the goal.	Adaption, flexibility, responsiveness is the goal.
In this type the environment is taken as stable and predictable.	In this type the environment is taken as turbulent and difficult to predict.
Sequential and synchronous process.	Concurrent and asynchronous process.
It is work centered process because people will change according to different phases.	It is people centered process, as the same team is developing throughout.
Project lifecycle is guided by tasks or activities.	Project lifecycle is guided by product features.
Documentation is substantial.	Documentation is minimal.
Developers do waiting until the architecture is ready.	The whole team is working at the same time on the same iteration. Good coordination between team members
Too slow to provide fixes to user.	Provide quick responds to user feedback
Change requirements is difficult in later stages of the project	Can respond to customer requests and changes easier
More time is spent on design so the product will be more maintainable. The "what ifs" arise earlier	There is no time for the what ifs
No communication within the team, novices stay in their rooms and try to understand things	High level of communication and interaction, reading groups, meetings
Restricted access to architecture	The whole team influences and understands the architecture. Everybody will be able to do a design presentation
Documents and review meetings are needed to solve an issue	5 minutes discussion may solve the problem
Everything is up front, everything is big before you start	The focus is on whether customer requirements are met in the current iteration



Normal releases take 18 months	After 10 months the first release was out

Numerous studies in favour of the Agile movement recommend transitioning from the Traditional software development methods to Agile software development methods because of the numerous benefits. However, these studies recommend researching and learning about the types of agile methods before converting from the traditional ones.

References

Bhalerao, Shilpa & Puntambekar, Devendra & Ingle, Maya. (2009). Generalizing Agile Software Development Life Cycle. International Journal on Computer Science and Engineering. 1.

Javdani Gandomani, Taghi & Zulzalil, Hazura & abdul ghani, Abdul azim & Md Sultan, Abu Bakar & Ziaei, Mina. (2013). Obstacles in moving to agile software development methods; At a Glance. Journal of Computer Science. 9620625. 620-625. 10.3844/jcssp.2013.620.625.

Mishra, Apoorva & Dubey, Deepty. (2013). A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios. International Journal of Advance Research in Computer Science and Management Studies. 1. 64-69.

Pathak, K., & Saha, A. (2013). Review of Agile Software Development Methodologies. International Journal of Advanced Research in Computer Science and Software Engineering, 3(2), 270–276.