

Use of Agile Methodology for Mobile Applications

Vaishnavi Patil¹, Sanjana Panicker², Maitreyi KV³

^{1, 2, 3} Student (B.E. in Computers), Fr. Conceicao Rodrigues Institute of Technology, Navi Mumbai, Maharashtra

Abstract—With the ever increasing popularity and demand for mobile applications, there has been a significant increase in the number of mobile application development projects. Highly volatile requirements of mobile applications require adaptive software development methods. The Agile approach is seen as a natural fit for mobile application and there is a need to explore various Agile methodologies for the development of mobile applications. This paper evaluates how adopting the Agile approach improves the development of mobile applications and if they can be used in order to provide more tailor-made process improvements within an organization. A case study on the music streaming application 'Spotify' shows how the agile methodology was implemented to deliver a commercially successful mobile based application. The findings of the study show the Agile methods have the potential to help deliver enhanced speed and quality for mobile application development.

Index Terms—Agile, Mobile Application

I. INTRODUCTION

The Agile methodology is a term which focuses on development which is tractable and it moves at a fast pace. This methodology is characterized by division of a large module into several smaller adaptive phases which are implemented iteratively for the development purposes. Agile methodology is a collection of values, principles and practices that is tailored for a particular software development project. It is an incremental model which assesses each phase and gives direction to the project, by giving direction the project shapes up to form a shippable final product. In the agile paradigm every aspect is carefully taken into account to improvise changes. Agile model is based on customer feedback, it takes into consideration customer feedback at each stage and then incorporates the user requirements as per their needs. All the agile model are modeled with a purpose. It is the backbone of agile modeling which when included with rapid feedback of customers produces reasonable effect on the project.

Fundamental agile models are based on parallelism where several artifacts are produced in parallel and applied to one another and thus moving forward at a steady pace. The stakeholders' participation is important as they are knowing what they want and it will contribute to faster development processes. These models work on 'inspect and adapt' approach which simultaneously implements the requirements it had found in the analysis phase ,also 'analysis-paralysis' problem which occurs during the analysis phase and which hampers the progress of the project. Agile methodology produces products that are up to the user standards by empowering them to continuously re plan and optimize the products so that

they stand to compete with the outside market.[1]

Agile development is itself not a methodology. It is an umbrella term which describes several agile methodologies. During the signing of Agile Manifesto in 2001, some methodologies included were Scrum, XP, Crystal, FDD, and DSDM. Since then, lean practices have majorly emerged as a valuable agile methodology and are included under the agile development umbrella. The most powerful technique amongst these is the Scrum which is rapidly being used in the mobile development applications. [2]

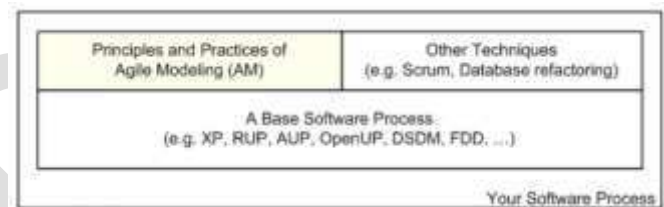


Figure 1: The classification of agile which finally builds the user software process.[3]

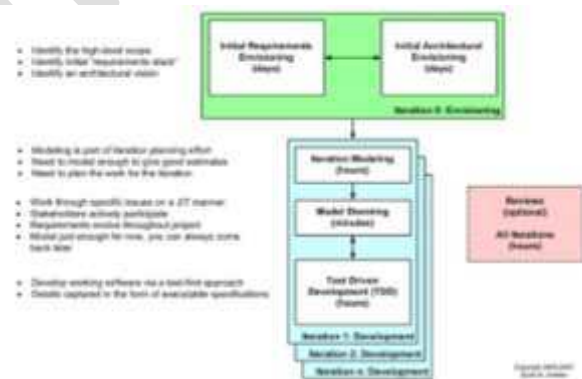


Figure 2: Architecture[4]

The modelling in agile follows the above architecture, where a higher level scope is taken whose requirements are listed so its architectural envisioning is complete after that an iterative model is followed which tests and implements these requirements made. Reviews are conducted simultaneously with the analysis based on customer feedback who response will be acknowledged in these iterations. [4]

II. CHARACTERISTICS OF AGILE METHODOLOGY

- People Oriented- Agile considers people – customers, developers, stakeholders, and end users – as the most

important factor for software methodologies. [5]

- Adaptive – The participants of an agile process are not rigid and are open to change. [5]
- Balancing Flexibility and Planning –A better planning strategy is to make detailed plans for the next few weeks, very rough plans for the next few months, and extremely crude plans beyond that. [5]
- Decentralized Approach –Agile software development spreads out the decision making to the developers. [5]
- Simplicity –The reason for simplicity is so that it will be easy to change the design if needed on a later date. [5]
- Collaboration –The customer of the software works closely with the development team, providing frequent feedback on their efforts. [5]
- Small Self-organizing teams –Responsibilities are communicated to the team as a whole, and the team determines the best way to fulfil them. [5]

III. AGILE VERSUS WATERFALL DEVELOPMENT METHODOLOGIES

Agile methodologies break down programming, development and project management into smaller modules. Documentation and quality testing is integrated at every step. In waterfall methodology the requirements are predefined ahead of time. Testing and documentation are the last steps of development. This makes agile development methodologies efficient and advantageous for mobile apps development. [6]

Why use Agile?

The traditional practices do not work due to :

- 1.Device diversity
- 2.Network connectivity
3. Other mobile-specific considerations.

In the Waterfall Model, the requirements are defined in the start. Users find it hard to describe what they need the mobile app to do. This causes failures in collecting the requirement for the app. This failure occurs in the first stage of the life cycle (i.e.Collecting Requirement) and propagates to the rest of the stages.

Historically many models have worked successfully in the development of Apps, but most of them do not suit the needs of Mobile Apps :

1. It is not viable to port the workflow of a Desktop App into Mobile Apps. Desktop Apps are mature whereas Mobile apps are relatively new category for most users. Desktop Apps do not have constraints faced by mobile apps like the size of the screen and nature of the platform. [7]

2. The screen size of mobile phones affect the session lengths. Mobile Apps are used for shorter lengths. This makes them significantly different from Desktop usage. [8]
3. Richer user interface, better responsiveness and intuitive user experience is crucial for Mobile Apps. Poor user experience is a huge complaint that Mobile Apps are facing. To overcome this the focus needs to concentrate on designing the optimal UI, and also combine this with a workflow that represents how users actually work. [9]
4. Testing the product greatly differs in Mobile Apps. In Mobile Apps, testing is dependent on the OS in the device as they differ in behavior. It is also dependent on the device and wireless network to which the device connected. Therefore, a combination of device types and OSs must be tested. A minimal of two-tier approach of testing must be conducted. First on device simulators. But since simulators don't produce a real user perspective, it has to be followed by testing on a subset of the most popular or latest devices. This can be enriched by in-the-wild [10] testing.
5. After deployed of a Mobile App, it is necessary to understand how it is actually used. An in-app analytics are recorded with which developers can learn how thw mobile app behaves and the customers liking to the app. The developers and stakeholders can get to know what users are doing inside the applications analyzing how they use the app,how often they use it along with popular features. They can modify the app to suit the needs of the cutsomer. [8]

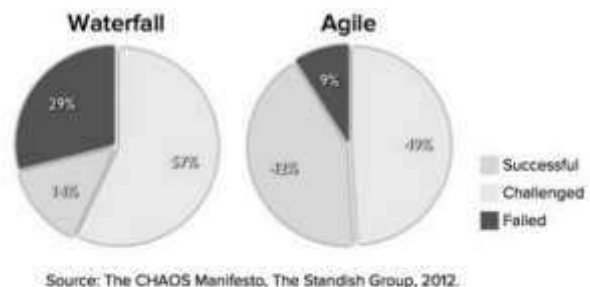


Figure 3: Comparison between Agile and Waterfall Models[11]

The rapid pace of change in the market has to be combated by adopting rapid development and deployment practices. These processes must be constantly iterated to deliver to the expectations of the user. It is of vital importance to be agile and quickly iterate to improve mobile apps. The need for ongoing analytics to monitor how the app is used, new OSs and new devices [12] flooding the market call for Agile methodology.

| Desktop | Mobile |
|--|---|
| Deployment takes 18 months; the application may be maintained for about five years prior to retirement. | Development and operations are complicated due to the number of mobile devices present since a wide range of hardware API access, screen sizes, resolutions, and performance changes rapidly. These changes in mobile devices present challenges in development. Moreover, these conditions are only to increase. |
| Rarely revised more than once after deployment. They are relatively stable. Not too challenging to manage. | Frequently revised to meet customer needs. Need to support frequent mobile app deployments and updates. |

Table 1: Desktop versus mobile application

IV. INSTAGRAM GOING AGILE

Waterfall model is used by many large companies to develop a new product. The evolution of Instagram was completely different.

They worked on the product and tested it with real users. From the experience of the user and their feedback, they made modifications to Instagram. They used agile methodologies that helped them to make swift modifications, cut down or build features based on the user inputs. This enabled them to give the customers what the customers actually wanted as opposed to what they thought the user would like. Within two months of their launch, they had a million downloads. [13]

The crux of Agile lies in building and testing it with the customers and getting the product to the market as quickly as possible.

V. CASE STUDY ON AGILE MOBILE APPLICATION: SPOTIFY

One of the best examples of a mobile application development team using Agile is Spotify, a music streaming application. Even though during its inception, Spotify started out by applying the SCRUM methodology, one of the iterative and incremental agile software development methodologies, it found the SCRUM framework very restricting, rigorous, complex and prevent flexibility especially in the crucial stages of expansion of the company when it started growing and splitting into various sub-teams and the complexity of the product increased. Spotify has been successful in implementing agile even though they have scaled to over 30 teams across 3 cities.

This made the company abandon the SCRUM framework and adopt a more flexible and experimental approach implementing the basic Agile development method. Even Mr. Alistair Cockburn, one of the founding fathers of Agile development was so impressed with agile implementation at Spotify that he said that he was looking for someone to implement this matrix format since 1992. It put its own spin to

the Agile rules and created its version of the Agile methodology based on those values that Spotify hold very dear.

So how did it make this change?

Here are the major changes it brought above and implementations that it carried out are described below via different aspects:

A) Change in roles

One of the first major changes that it implemented was concerned with the terminologies.

1. Scrum master became Agile coach
2. Teams became squads

B) Overall Implementation and Organization

These squads consisted of about eight members and they were given complete autonomy to be a cross-functional and self-organizing unit. They are a self-organizing team, which has all the required skills and tools needed to design, develop, test and release the product to production. They have complete autonomy in deciding their own way of working which could differ from team to team. This autonomy gives them the freedom to decide WHAT to build, HOW it has to build and how the squad would work together to build the end product. The key principle is "Be autonomous, but don't sub optimize".

The goal they had was to achieve "Loosely coupled, yet tightly aligned squads".

The key principle used here is alignment and the product manager decides which problem has to be solved and the squads collaborate with each other to find the best solution.

Different squads might use completely different solutions to similar problems but they are able to do so because of Spotify's culture to value cross-pollination more than standardization which results ultimately in a healthy balance between both consistency and flexibility.

Every squad also has a long-term mission which will emphasize the contribution of the squad to Spotify. [14]



Figure 4: Example of different Squad Responsibilities [14]

Squads may learn Lean Startup principles such as Minimum Viable Product [MVP] and validated learning.

The architecture used at Spotify is designed to keep the systems independent and decoupled of its interfaces and protocols and each squad focuses mainly on its own system or systems. There exists an internal open source model which promotes “Sharing, not owning”.

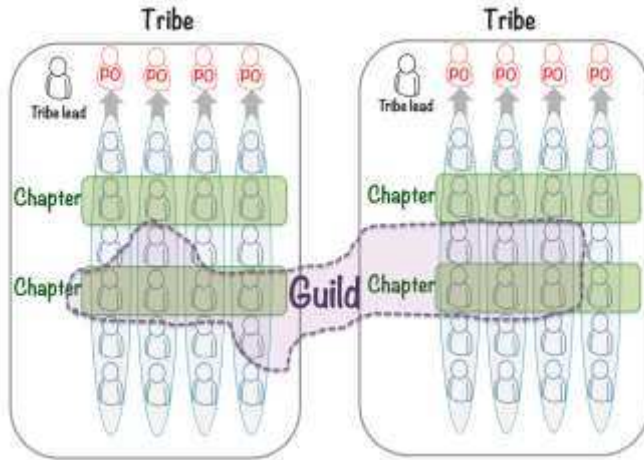


Figure 5: Organizational structure at Spotify [15]

C) Tribes and Chapters

Squads are grouped into tribes, where the tribe is a light weight matrix. - Each employee is a part of a squad as well as of a chapter. A chapter includes employees who have similar skills and work within the same general competency area, within the same tribe.

While the squad is the primary dimension which focuses on product delivery and quality, the chapter is a competency area such as Quality Assurance, Agile Coaching or Web Development. As a squad member, the chapter lead is the employee's formal line manager who coaches and mentors him which enables the employee to shift between squads under the same manager.

D) Guild

A guild is a light weight community of interest that is more wide-reaching and it consists of groups of employees that want to share knowledge, tools, code and practices. Every guild has a guild coordinator and mainly communicate informally.

Spotify makes small but frequent releases and works on a continuous delivery infrastructure.

An example of this is that when it started out, all the teams had to co-ordinate with each other to bring about one release which made the process unnecessarily long and tiresome for everybody.

As a solution to this, Spotify decoupled all the components of the architecture to enable teams to bring about releases

independent of each other using Chromium Embedded framework as a tool.

The exact system they use is based on the concept of release trains and feature toggles.

This lead to a classification of squads into three types:

1. Client App Squads
2. Feature squads
3. Infrastructure Squads

It follows a self-service model that encourages enabling rather than serving. [15]

E) Mutual Respect, Sharing and Co Ordination

The employees at Spotify have a satisfaction rate of 94% and this is no mere miracle. The employees maintain a culture of constantly motivating their peers and praising them for their accomplishments. The overall culture is very helpful and employees often lend a helping hand to one another and brainstorming ideas is highly encouraged.

Community over Individuality

Community is favored over individuality here.

Squad Dependencies

Since there are multiple squads, the possibility of dependencies is also high. While dependencies need not be bad, it is necessary to avoid those which will have a negative impact, such as block or slow down a squad.

Squads are thus asked to name which are the squads that it depends on, and which ones are dependent on them. After due deliberation, reorganization, reprioritization, organizational and technical changes are carried out.

Fail-Friendly Environment

Another feature is that it encourages failure and promotes its employees making mistakes as often as required so that while overcoming the stigma off failure by constantly sharing them with. Colleagues, the product has ultimately benefits as it would definitely be a stronger one.

VI. CONCLUSION

Although there are many benefits of using agile methodologies, these methodologies cannot be completely used in all projects. Agile methodologies are adaptive with changes and today some projects are doing based on heavy methodologies yet, choosing a methodology is dependent on type, scale of project and another factor. There has been a significant increase in number of mobile application development projects as the demand for mobile applications are growing and the potential number of different mobile applications is virtually unlimited. Furthermore, the published research work related to mobile software confirms Agile practices to be a natural fit for the development of mobile

applications. An appropriate Agile method could be selected for a given project and can be tailored to a specific requirement based upon project's complexity and team size. Despite the identified business opportunity, very few scientific publications can be found, which address the specific problems that the development organizations are facing while developing software for mobile devices. The mobile software development companies and researchers would suitably adopt agile methodologies as a generic development culture without worrying about specific agile methodologies. In addition, it is anticipated that agile can be successfully implemented to enhance and evaluate the overall quality and performance of mobile software development process that brings forth many observations which indicate substantial scope for further research in the domain.

ACKNOWLEDGMENT

Every project big or small is successful largely due to the effort of a number of wonderful people who have always given their valuable advice or lent a helping hand. I sincerely appreciate the inspiration; support and guidance of all those people who have been instrumental in making this project a success. We are grateful to Mrs. Shweta Tripathi, our Software Engineering teacher for encouraging us to broaden our horizons and read more and learn more by way of assignments which help us discover topics like these. We would also like to express our gratitude to Mrs. Kiruthika, the coordinator for the Computer Department for her encouragement for the same. We also would like to thank Mr. Henrik Kniberg, Spotify for his wonderful videos on YouTube which were of immense help and Mr. Brendon Marsh, Spotify for his insightful answers on Quora.

REFERENCES

- [1]. Agilemethodology.org. (2016). *The Agile Movement*. [online] Available at: <http://agilemethodology.org/> [Accessed 20 Oct. 2016].
- [2]. Msdn.microsoft.com. (2016). *Agile Principles and Values*, by Jeff Sutherland. [online] Available at: [https://msdn.microsoft.com/en-us/library/dd997578\(v=vs.120\).aspx](https://msdn.microsoft.com/en-us/library/dd997578(v=vs.120).aspx) [Accessed 20 Oct. 2016].
- [3]. Anon, (2016). [online] Available at: <http://www.agilemodeling.com/essays/introductionToAM.html> [Accessed 20 Oct. 2016].
- [4]. Agilemodeling.com. (2016). *Agile Architecture: Strategies for Scaling Agile Development*. [online] Available at: <http://agilemodeling.com/essays/agileArchitecture.htm> [Accessed 20 Oct. 2016].
- [5]. Characteristics of agile methodology <http://www.unf.edu/~broggio/cen6940/ComparisonAgileTradition al.pdf>
- [6]. S. Shahid, "Agile Methodology For Mobile Application Development", *LinkedIn*, 2016. [Online]. Available: <https://www.linkedin.com/pulse/agile-methodology-mobile-application-development-salman-shahid>. [Accessed: 20- Oct- 2016].
- [7]. "Experts: Rethink Mobile App Development or Fail -- Visual Studio Magazine", *Visual Studio Magazine*, 2014. [Online]. Available: <https://visualstudiomagazine.com/articles/2014/09/10/experts-mobile-app-success.aspx>. [Accessed: 20- Oct- 2016].
- [8]. "Gartner Says Traditional Development Practices Will Fail for Mobile Apps", *Gartner.com*, 2016. [Online]. Available: <http://www.gartner.com/newsroom/id/2823619>. [Accessed: 20- Oct- 2016].
- [9]. "New Mobile App Dev Puts Pressure On IT Team - CXOtoday.com", *Cxotoday.com*, 2016. [Online]. Available: <http://www.cxotoday.com/story/mobile-apps-needs-new-development-methods/>. [Accessed: 20- Oct- 2016].
- [10]. H. Alvarez, "4 Ways to Test Your Product In the Wild | UserTesting Blog", *UserTesting Blog*, 2015. [Online]. Available: <https://www.usertesting.com/blog/2015/02/26/test-in-the-wild/>. [Accessed: 20- Oct- 2016].
- [11]. The Sandish Report, *Waterfall Vs. Agile*. 2012.
- [12]. "Worldwide Smartphone Growth Forecast to Slow to 3.1% in 2016 as Focus Shifts to Device Lifecycles, According to IDC", *www.idc.com*, 2016. [Online]. Available: <http://www.idc.com/getdoc.jsp?containerId=prUS41425416>. [Accessed: 20- Oct- 2016].
- [13]. "Instagram, the Titanic, and Agile Product Development", *Blog.bcaresearch.com*, 2016. [Online]. Available: <http://blog.bcaresearch.com/instagram-the-titanic-and-agile-product-development>. [Accessed: 20- Oct- 2016].
- [14]. H. Kniberg and A. Ivarsson, *Scaling Agile @ Spotify with Tribes, Squads, Chapters & Guilds*, 1st ed. Spotify, 2012, p. Oct 2012.at-the-spotify-engineering-culture/
- [15]. Ulf Eriksson, How Spotify does agile – A look at the Spotify engineering culture (31 March, 2015)<http://reqtest.com/blog/how-spotify-does-agile-a-look-at-the-spotify-engineering-culture/>