ISSN No. 0976-5697

Volume 8, No. 5, May-June 2017

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

Agile Development Method for Mobile applications: A Study

Mudasir M Kirmani
Assistant Professor, Computer Science
Sher-e-Kashmir University of Agricultural Sciences & Technology of Kashmir
Srinagar, J&K, India

Abstract: It is highly important to be knowledgeable about the software development methodologies available in order to make the right choice when it comes to developing a mobile application. For developing mobile applications, existing software development process models are incorporated but looking at remarkable growth of mobile devices, the complexity involved in the development of mobile application needs a dedicated life cycle process model, instead of using traditional life cycle process models. Although many existing life cycle models are adapted for mobile application development but there are certain constraints that needs to be addresses specifically. This paper presents the results of literature review performed in order to identify newly created methodologies targeting the development of mobile applications. The unique requirements and constraints associated with mobile systems have brought new challenges to software development for such environments as its demands extensive improvements to traditional systems development methodologies in order to fulfill special needs of this field. Agile software development methods have caught attention of software engineers and researchers worldwide. The comparative analysis is performed using the methods life-cycle coverage and empirical evidences. This paper provides detailed review and analysis on the use of agile methodologies with mobile application skills.

Keywords: agile, mobile, application

INTRODUCTION

Software process model describes a sequence of activities carried out for developing software. A generic framework for software engineering defines five frame work activities i.e., communication, Planning, Modeling, Construction and Deployment. All software process models can put up the generic framework activities but each applies a different emphasis to these activities and leads to a process flow that appeals each frame work activity in a different manner. There exists various process models like waterfall, spiral, RAD, Agile etc.

Agile means "quality of being agile", activity. In development work, accordingly development groups concentrate only on the functions needed immediately, delivering them fast, collecting feedback and reacting rapidly to business and technology changes. The aim of this paper is to organize, analyze and make sense out of the dispersed field of agile software development methods. Based on the results of the analysis, practitioners are in a better position to understand the various properties of each method and make their judgment in a amore informed way. For these purpose an analytic framework is constructed which guides the analysis of the different existing methods.

Finally it is important to notice that the development of mobile applications has some similarities with the traditional development e.g., after performing an extensive literature review, Hosbond and Neilsen (2005)**[1].** concluded that scope of mobile systems development is an extension of the scope and the body of knowledge on traditional systems development. However they also noted that in the existing literature knowledge about traditional systems development is largely neglected. Due to significant differences in the environment and in platform specifications, mobile application development requires a suitable development methodology. By taking into account the main features of mobile application development scenario, a matching development paradigm can be identified.

RESEARCH OUTLINE

This paper discusses different models out of many development model's that exists and the comparison includes the advantages and gaps of different models which can help to select specific model at specific situation depending on customer request and including business requirements. It also describes about agile methodology, its various principles and steps. It also gives a comparison of agile and waterfall models and also describes the benefits of agile over traditional methodology.

Related Research:

Agile methods represent relatively new approach to software development; the ideas behind these methods originate from the principles of Lean Manufacturing (in the 1049's) and the Agile Manufacturing, which emphasized the adaptability to enterprises to a dynamic environment**[2]**. Boehm and Turner **[3]**, the author identify fundamental concepts to agile development ; simple design principles, a large number of releases in a short time frame, extensive use of refactoring, pair programming, test driven development and seeing change as an advantage.

Another definition of agile methods is provided in (Abrahamsson,et al.,2002)**[4]** an agile development method is incremental (multiple releases): a little more is added in each iteration until final product is finished), cooperative (good cooperation between developer and client), straight forward (easy to understand and modify) and adaptive (allowing for frequent changes).

Dyba and Dingsoyr,2009**[5]**; partition studies into four categories: introduction and adaptation, human and social factors, perception of agile methods and comparative studies.

In (Abrahamsson,2005)**[6]**, the author performs a direct comparison between Agile method characteristics and mobile application feature ; focusing on environment volatility, amount of documentation produced, amount of planning involved, size of development team, scale of the application in development, customer identification and object orientation.

Barlow et al., suggested that agile methods seem more suitable for developmental and non-sequential projects.

Boehm and Turner and Beck suggested that agile development was found to be less reliable and suitable for certain types of environment and teams that include small number of experts.

Boehm and Turner**[7]** recognize five main factors that affect agility: Operate culture, Team size, Criticality of the software, Competence of the developers and Stability of the requirements

Boehm argues that a software development method works best when it is applied to situations with specific traits [**8**]. He calls these situations the "home grounds" of the software development method.

Corral et al.**[9]**, presented a survey shows lack of evidence that shows a clear link between the proposed agile methodologies and their utilization in a real world setting.

Abrahamson identifies agile methods as a potential solution for mobile software development [**10**]. Based on the home ground for agile methods, he performs a comparative analysis to prove the suitability of agile methods for the development of mobile software.

Literature of various software process models for mobile application development:

Classical models of system life cycle:

Software development methodologies play a vital part of developing the software. There are many methodologies which are used by the professional software development companies nowadays. There are certain advantages and disadvantages associated with each of them. The basic purpose of these methodologies is to provide smooth software development according to the project requirements.

Software development methodology is a framework that is used to structure, plan, and control the process of developing an information system. This kind of development methodologies are only concerned with the software development process, so it does not involve any technical aspect of, but only concern with proper planning for the software development

Model title	Advantages	Gaps
Waterfall Model	-The documentation_and_structure design represents an advantage when new members join the team. -A simple model that is easy to use by all members. -Each stage has an expected outcome, as the model is rigid(it imposes total control over stages). -Stages are implemented individually. -It is recommended to be used for small projects, in which requirements are very clearly formulated.	-Problems that emerge in one stage are completely solved in another stage. -it doesn't allow to partition the project according to stages. -When new requirements are formulated, they will be implemented in another version. -Consequently it imposes supplementary costs for their implementation. -It is hard to give a correct estimate of the time and cost allotted for each stage. -The finite product is obtained quite late.
Prototype Model	-Users are directly involved in development and they can better understand how the application functions by means of the prototype. -Errors can be detected in time. -The user's feedback is fast, which leads to better solutions. -Less time and lower costs.	-The model leads to increase in the system's complexity, and it goes beyond the conditions established at first. -the project analysis is insufficient. -Developers may become attached to a prototype, out of subjective reasons, running the risk to transform the prototype into a final product even though the basic architecture is not correct. -It takes an excessive amount of time to implement a prototype.
Incremental Model	-The highest priorities are the first to be delivered. -Deliveries are done once in a few weeks. -The feedback on the delivered increments offers the specifications for subsequent increments. -The model ensures a low risk of total failure of the project. -Higher priority increments undergo an ampler testing process.	-It may be necessary to create temporary solutions so as to deliver an increment in time. -In some some situations, a significant part of the code can be erased. -Under the circumstances, planning is difficult.
Iterative Model	In iterative model we can only create a high-level design of the application before we actually begin to build the product and define the design solution for the entire product. Later on we can design and built a skeleton version of that, and then evolved the design based on what had been built.	-Each phase of an iteration is rigid with no overlaps. -Costly system architecture or design issues may arise because not all requirements are gathered up front for the entire lifecycle

	-In iterative model we are building and improving the product step by step. Hence we can track the defects at early stages. This avoids the downward flow of the defects. -In iterative model we can get the reliable user feedback. When presenting sketches and blueprints of the product to users for their feedback, we are effectively asking them to imagine how the product will work. -In iterative model less time is spent on documenting and more time is given for designing.	
V Model	-It can be used in object-oriented programming as it favors the turning of higher structures into prototypes and their reuse. -It provides the strong control over the system, which allows it to be used in the case of complex systems as well. -It encourages one to approach the system accordingly to its constitutive parts.	-The major report that has been brought to this model is connected to the fact that the validation stage is launched late, which makes the system suffer from the point of view of efficiency.
Spiral Model	-It allows one to assess risks at several moments. -The model is characterized by high flexibility(both in fund allocation and in defining activities).	-The allocated time and the costs involved are hard to estimate from the beginning. -It is costly model to use. -It doesn't work well for smaller projects.

Agile Models:

The development process of mobile application differs from traditional software's as the user expect same features similar to their desktop computer applications with additional mobile specific functionalities. Apart from the classical models used in software product development, we ought to mention group of methods termed agile methods. They are based on incremental and iterative development and they are mostly applied in cases that require project specifications and solution be the product of the collaboration between teams organized individually but which aims towards the same common purpose. Agile methods were developed starting from twelve principles included in "Agile Manifesto".**[11]** which are going to list below:

-Satisfying the customer requirements, through quick delivery of usable software solutions.
-Possibility to change specifications, no matter how late the product would be in its implementation stage.
-Software versions delivered very frequently.
-The main measure of progress is usable software.
-Development is at steady pace, it can maintain a steady rhythm.
-Close collaboration between developer and customer.
-Face-to Face conversation is the best way to communicate.
-Projects are made by motivated individuals of high credibility.
-Teams are organized individually.
-Adaptation to changing circumstances.
-Constant attention to technical excellence and quality design.
-Simplicity.

Agile started in the mid -1990's. This methodology can describe as incremental and iterative development including flexibility throughout the systems development life cycle, minimal groundwork, customer relationship and regular delivery.

Agile methodologies aim at facilitating software development processes where changes are acceptable at any stage and provide a structure for highly collaborative software development. Agile methodologies are among the best software development approaches to apply at times, when customer requirements are not exact or when the deadlines and budgets are tight. Rather than creating tasks and schedules, all time is "time-boxed" into phases called "sprints". Each sprint has defined duration (usually in weeks) with a running list of deliverables, which are determined by customer. If all planned work for the sprint cannot be completed, work is reprioritized and the information is used for future sprint planning.

Drawback: Agile methodologies have been criticized in their ineffectiveness when used in large organizations.

The process of Agile Software Development:

Agile SDLC contains sic basic phases as shown below:

1-Pre-project preparation: The basic of the project and the market features well defined in this phase.

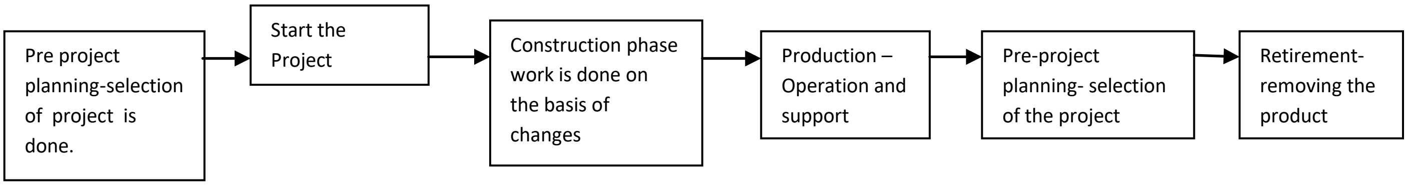
2-Start: Requirement displaying done through active participation of stakeholders in order to identify the initial requirement basic or high level requirements for the system. The main objective is to understand the problem.

3-Construction: In this phase excellent working software is delivered incrementally, which meets the variable needs of the customer. At the end of each progress iteration there is a fractional, working system which is shown to the customer and testing can be done on that like system integration testing.

4-Production: It handles all operation and provides support.

5-Selection: The goal of construction phase is to keep systems useful and active after the product has been deployed to the user

6-Departure: It is also known as system deactivating phase.


Pre project planning-selection of project is done.
Start the Project
Construction phase work is done on the basis of changes
Production – Operation and support
Pre-project planning- selection of the project
Retirement- removing the product

Agile software methodology have the potential to provide higher customer satisfaction, low bug rate, shorter development cycles and quicker adaptation to rapidly changing business requirements.

Comparison Between Traditional and Agile:

The comparison between classical and agile model shows periodically important towards SDLC models.

Traditional	Agile
- Measured and formal, direct ordering of steps, based on rules.	- Developing iterative and investigative, outside formal rules.
- Everything is upfront and large before starting.	- The focus is on whether customer requirements are met in the current iteration.
- Documents and Evaluation are needed to solve an issue.	- A short discussion can solve the problem.
- No communication within the team.	- High level of communication and iteration, understanding groups, meetings.
- Extra time is spent on design so the product will be more maintainable.	- There is no time for what is unknown.
- Modification necessities is difficult in later stages of the project.	- Can respond to customer requests and changes easier.
- Successive and synchronous process.	- Parallel and asynchronous process.
- In this the environment is taken as steady and expected.	- In this the environment is taken as temperature and difficult to predict.

Why Agile is better?

Agile practices enhances and speeds up the development of mobile app projects .

-Agile works well with highly volatile requirements of mobile apps.

-Agile development encourages stakeholder's involvement in mobile projects. This provides excellent visibility for key stakeholder's, enables rapid accommodation of stakeholder feedback and helps in rapidly rolling out suggested new features and ensuring that expectations are effectively managed.

-Agile increases reliability and leads to continued use of mobile apps:. If a mobile app crashes a few times, the mobile user will easily switch an alternative app as they have many choices available.

-Agile development empowers user experience for mobile apps: Agile development enables thoughtful user experiences and allows developers to experiment different options in subsequent sprints and adjust the design and features of apps to make the user experience is quick, smooth and seamless.

-Agile fits incomplete requirement nature of mobile projects: Initial requirements of mobile apps are generally incomplete, unclear, in sufficient, uncertain and change considerably during the development process. Developers will commonly put out an app with a limited set of features in the first release and update it in later versions. This nature of mobile app development fits with the iterative nature of agile.

-Agile development fits the experimentation and adaptation of mobile apps: The process of refining and improving a mobile app is accomplished with the help of customer feedback.

-Agile helps in identifying the risk in mobile projects at early stage: Small incremental releases made visible to the product owner and product team through its development help to identify any issues early in the project as they arise, making it much easier to respond to change.

-Agile is the best suitable for quick delivery and short development lifecycle of mobile apps: For initial release, a minimum viable product with prioritized features is built and delivered as fast as possible, followed by additional features in later versions.

Below table provides comparison between the agile methodology and traditional approaches. In this context the comparison focuses on five factors: product size, information sensitivity, enterprise dynamics, employees and environment/culture.

Analyzed Factor	**Agile Models**	**Traditional Methods**
Dimension	Can be used with small-size products and teams. They are based on tactic knowledge, there are limitations at the level of scalability.	They make it possible to manage large teams and ample projects. They are difficult to use with small projects.
Critical Projects	They have not been tested in the case of critical products. Difficulties can occur due to the simplicity of design and the	They have sufficiently evolved to successfully face critical products. Yet, they are hard to use with small-scale

	lack of documentation.	critical projects.
Dynamism	Due to their simple design and multiple versions, they are frequently used in dynamic environments.	Their use in dynamic environment is very expensive but they are considered an excellent choice in stable environment.
Employees	They require the massive presence of a large number of experts, who are expensive and hard to find. It is risky to use employees who are not familiar with agile methods.	They require many experts only in the project definition stage; subsequently, the project can use few employees, on condition that the field should be stable.
Environment/Culture Enterprise	They thrive in environment that offer comfort and freedom to their employees.	They are feasible in environments in which employees have comfort and freedom and roles are clearly defined through practices and procedures.

Source: adapted from **[12]**.

CONCLUSION

Our investigation and related research indicates that agile methods have a good level of suitability for the development of mobile applications. Agile processes were considered to be very appropriate for fast-paced markets, where customer's satisfaction is governed by early and frequent delivery, where there is scope for changes even late in the project, the delivery cycle is short, there is appropriate collaboration between businesses and developers and where there is continuous attention to technical excellence and good design and simplicity. 86% of survey participants believed that agile methods and practices are appropriate for the development of mobile applications. The findings of the study show the adopting and adapting these Agile methods are suitable and proposed to help deliver enhanced speed and quality for mobile application development. The result shows that the agile software development methods, cover certain/different phases of the software development life cycle and most of them don't offer adequate support for project universal solution and empirical evidence is still very limited. Based on the results new directions are needed and to place emphasis on methodological quality not method quantity. As perfect model should combine the responsibility and accountability of waterfall model; with quickness and adaptableness of the agile model. Thus that can create a more efficient software improvement model.

REFERENCES

1. J.H. Hosbond and P.A.Nielsen, "Mobile Systems Development" A literature review" in proceedings of IFIP 8.2 Annual conference,2005.
2. Salo, O. (2006) Enabling Software Process Improvement in Agile Software Development Teams and Organizations. Helsinki:VTT.
3. Boehm, B., & Turner , R.(2003). Balancing Agility and Discipline : A Guide for the Perplexed. Addison-Wisely.
4. Abrahamsson, P., Salo, O., Ronkainen, J., &Warsta, J.(2002). Agile Software Development Methods: Review and Analysis: VTT Electronics.
5. Dyba, T., & Dingsoyr, T.(2009). What Do We Know about Agile Software Develeopment? IEEE software, 26, 69.
6. Abrahamsson, p.(2005). Mobile Software Develoepment-the business opportunity of today. Proceedings of the International conference on Software Develeopment,(pp.20-23). Reykjavik.
7. B.Boehm and R. Turner Balancing agility and discipline: A guide for the perplexed, Adisson-Wesley, 2003.
8. B.Boehm, " Get ready for agile methods , with care: IEEE Computer, Vol. 35, No. 1,2002,pp.64-69.
9. Corral, L.(2012). Using Software Quality Standards to Assure the quality of the Mobile Software Product, In Proc. of the 3rd Annual Conference on systems, Programming and Applications: Software for Hummanity (SPLASH'12), pages 37-40, Tucson,AZ,USA.ACM.
10. Abrahamsson, P.(2007). Agile Software Development of Mobile Information Systems. In Advanced Information Systems(pp. 1-4). Berlin: Springer.
11. Agile Alliance. Manifesto for Agile Software Development. [Online] Retrieved 16th March 2009. Available at: http://www.agilemanifesto.org.
12. B.Boehm , R. Turner, " Balancing Agility and Discipline : Evaluating and Integrating Agile and Plan-Driven Methods,"Proceedings of the 26th International Conference on Software Engineering, Washington, DC,USA,2004.