International Journal of Research in Computer Science
eISSN 2249-8265 Volume 3 Issue 4 (2013) pp. 9-18
www.ijorcs.org, A Unit of White Globe Publications
doi: 10.7815/ ijorcs.34.2013.068

A REVIEW AND ANAYSIS ON MOBILE APPLICATION DEVELOPMENT PROCESSES USING AGILE METHODLOGIES

Harleen K. Flora[1], Dr. Swati V. Chande[2]

[1]*Ph.D Scholar, Department of Computer Science, The IIS University, Jaipur, INDIA*
Email: harleenflora@gmail.com

[2]*Professor, Department of Computer Sciences, International School of Informatics and Management, Jaipur, INDIA*
Email: swatichande@rediffmail.com

Abstract: *Over a last decade, mobile telecommunication industry has observed a rapid growth, proved to be highly competitive, uncertain and dynamic environment. Besides its advancement, it has also raised number of questions and gained concern both in industry and research. The development process of mobile application differs from traditional softwares as the users expect same features similar to their desktop computer applications with additional mobile specific functionalities. Advanced mobile applications require assimilation with existing enterprise computing systems such as databases, legacy applications and Web services. In addition, the lifecycle of a mobile application moves much faster than that of a traditional Web application and therefore the lifecycle management associated therein must be adjusted accordingly. The Security and application testing are more stimulating and interesting in mobile application than in Web applications since the technology in mobile devices progresses rapidly and developers must stay in touch with the latest developments, news and trends in their area of work. With the rising competence of software market, researchers are seeking more flexible methods that can adjust to dynamic situations where software system requirements are changing over time, producing valuable software in short duration and within low budget. The intrinsic uncertainty and complexity in any software project therefore requires an iterative developmental plan to cope with uncertainty and a large number of unknown variables. Agile Methodologies were thus introduced to meet the new requirements of the software development companies. The agile methodologies aim at facilitating software development processes where changes are acceptable at any stage and provide a structure for highly collaborative software development. Therefore, the present paper aims in reviewing and analysing different prevalent methodologies utilizing agile techniques that are currently in use for the development of mobile applications. This paper provides a detailed review and analysis on the use of agile methodologies in the proposed processes associated with mobile application skills and highlights its benefit and constraints. In addition, based on this analysis, future research needs are identified and discussed.*

Keywords: *Agile Methodologies, Mobile Software Development.*

I. INTRODUCTION

Software development methodologies have evolved since the 1970s. Agile methodologies came into existence to accommodate changing business requirements and for better management of the software development lifecycle. It provides practices that facilitate communication between the developer and the customer, and undergo develop-deliver-feedback cycles, to have more specific view of the requirements, and be ready for any change at any time. Agile development methodology helps companies build the right product and empowers teams to continuously redesign their release to optimize its value throughout development, allowing them to be as competitive as possible in the marketplace. Therefore, the main aim of the methodology is to deliver what is needed at appropriate time during the development cycle. Agile methodologies are among the best software development approaches to apply at times, when customer's requirements are not exact, or when the deadlines and budgets are tight. However, besides the benefits associated in employing agile technologies in mobile software development there are issues that raise concern in reporting gains in quality, productivity and business satisfaction by different groups. It has been reported that agile methods had been successful in delivering in majority of cases whereas there are conflicting reports that claim that the methodology is still too young to require extensive academic proof of their success. This paper is an attempt to review the published literature on application of the agile approaches for the development of mobile software as the researchers believe that agile innovations offer a solution to mobile specific applications that requires high quality development processes.

II. AGILE SOFTWARE DEVELOPMENT

The Agile Manifesto, also called the Manifesto for Agile Software Development, is a formal affirmation of four key values and 12 principles to guide a reiteration to software development. The Manifesto was published by a team of software developers that discussed various means for lightweight development methods as compared to Extreme Programming, SCRUM, DSDM, Adaptive software development, Crystal, Feature-Driven Development, and Pragmatic Programming [1]. In easier terms, The Agile manifesto distinct a development process that provided a guiding force for agile practitioners. Agile methodology focuses more on the human aspects of software engineering than the perspective processes, thereby employing human interaction over tools and processes. The Agile Manifesto established a common set of values and principles for all of the individual agile methodologies. It details four core values for enabling high-performance, efficiency and outputs:

1. Individuals and their interactions
2. Delivering working software
3. Customer collaboration
4. Responding to change.

These core values are further supported by 12 principles which underlie the Agile Manifesto that includes

1. Customer Satisfaction through early and frequent delivery
2. Scope for changes even at a later stage in the project
3. Short delivery cycle (e.g., every couple of weeks)
4. Collaboration between businessmen and developers
5. Motivation among individuals
6. Face to face communication
7. Working software-Primary measure of progress
8. Promoting sustainable development pace
9. Continuous focus on technical excellence and good design
10. Simplicity
11. Self-Organization to obtain best results
12. Self-improvement.

III. MOBILE APPLICATION DEVELOPMENT

Mobile Application Development is the process by which applications are developed for small low-power handheld devices which are either pre-installed on phones during manufacture, or downloaded by customers from app stores and other mobile software distribution platforms. The mobile application market is witnessing a rapid growth with the increasing popularity and demand from users for a wide variety of mobile applications. There has been significant increase in number of projects for mobile application development services as mobile platforms continue to advance in performance and people skilled in developing applications for mobile phones and tablets are highly in demand. It has been speculated that developing mobile applications is similar to software engineering for other embedded applications in many aspects [2]. The issues that commonly encountered include integration with device hardware, as well as traditional issues of security, performance, reliability, and storage limitations. However, mobile applications present some additional requirements that are less commonly found with traditional software applications, including: interface with other applications, handling device movement, mobile application types, cross platform compatibility, varying hardware complexities, security risks, privacy, user interfaces, testing complexity, power consumption, 24/7 “always on”, personal mobile computer, development process, application size, and mobile phone screen size.

The rapid growth of mobile computing platform has surpassed the software engineering processes tailored to mobile application development. The mobile software development teams face the challenge of the dynamic environment of different development and technical requirements, with frequent modifications in customer needs and expectations. The changing need and expectations make the systems more complex. During this continuous changing environment, the agile processes are considered to be most suitable for fast-paced markets, where customer satisfaction is governed by early and frequent delivery, scope for changes even late in the project, shortened delivery cycle, collaboration between businesses and developers,, need for continuous technical excellence, good design, simplicity and where working software is the primary measure of progress.

IV. IS AGILE – A NATURAL FIT FOR MOBILE APPLICATION DEVELOPMENT?

The mobile telecommunications industry comprises a highly competitive, dynamic and uncertain environment. The agile approach is seen as a natural fit for mobile application development. and studies carried out for the application of the agile development approach to mobile application development indicates the need for software development processes tailored to suite the mobile application requirements [3]. It has been recommended that agile practices are the best choice which assures different phases of software development life cycle and to solve the mobile application development issues more efficiently [4]. It is believed that agile innovations may offer a variety of

solutions for mobile application and assist service developers in need of high quality development processes [2].

Abrahamsson et al. [5] have demonstrated the traits which reasons why agile technologies fits best in mobile software development. The various issues includes, high environment volatility, small development teams, identifiable customer, object-oriented development environment, non-safety critical software, application level software, small systems and short development cycles.

Kannan [6] has also highlighted the suitability of agile software development in mobile application development because of small teams, short deadlines, importance of usability, fast delivery and less complexity. The authors have suggested seven methods in which Agile development practices enhance the development of mobile apps that includes experimentation and adaption nature of mobile apps; reliability that leads to continued use of apps; extension of Agile sprints into mobile app model, responsiveness to technology changes; rapidly accommodating customer feedback; a more thoughtful user experience; and phased roll out of feature sets.

Holler [3] suggested that agile software development offers tremendous opportunities and value, for mobile development teams working into introducing a lightweight development process or scale back bureaucratic processes. The author has emphasized about the progress in mobile computer technology and the rapid escalation of wireless networks in quality and quantity that has brought in new applications and concerns in this dynamic environment. He has also underlined the promptness with which the industry needs to adapt and change itself from conventional systems development techniques fulfilling the special needs of this field.

In addition, agile methodologies have also been criticized in its ineffectiveness when used in large organizations and certain types of projects where it has been enlisted as an area that needs further research. It has been suggested that agile methods seem best for developmental and non-sequential projects and many organizations believe that the methodology is too extreme. Barlow et al. [7] suggested that these organizations should prefer adopting a hybrid approach that mixes elements of agile and plan-driven approaches to fulfill their needs. Moreover, they suggested that agile methods seem more suitable for developmental and non-sequential projects and many organizations believed that agile methodologies were too extreme. Boehm and Turner [8] and Beck [9] 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.

The technology has also been assessed by numerous observers as being a “management fad” and claims of a measurable business improvement via measurement of metrics defined by itself (e.g. velocity) [10, 11, 12]. The technology has its limitations in distributed development efforts, using an Agile Software Process with Offshore Development, and mission-critical systems where failure is not an option at any cost (e.g. software for air traffic control) (Fowler, 2010) [13].

The technology has also been criticized for various other reasons that includes lack of structure and necessary documentation, works with senior-level developers, incorporates insufficient software design, requires too much cultural change to adopt, can lead to more difficult contractual negotiations, feature driven, non-functional quality attributes are hard to be placed as user stories. [14].

In the scope of mobile software development, Corral et al. presented a survey that shows a lack of evidence that shows a clear link between the proposed Agile methodologies and their utilization in a real-world setting [15], instantiating a trend mentioned in a previous analysis by Janes et al. [16], that reflects an identified decline on considering Agile practices as a silver bullet.

V. REVIEW OF MOBILE APPLICATION DEVELOPMENT PROCESSES - USING AN AGILE APPROACH

Following Agile methodologies have been proposed that uses combination of agile and non-agile techniques for the development of mobile applications by various scientists in last decade.

Table 1: Mobile Application Development Processes using Agile Methodologies

Mobile Process	Mobile Development Process Description	Year	Techniques
Mobile D	An Agile Approach for Mobile Application Development	2004	XP, Crystal, RUP
RaPiD 7	Rapid Production of Documentation - 7 steps	2005	AM
Hybrid Methodology Design	Designing an Agile Methodology for Mobile Software Development - A Hybrid Method Engineering Approach	2007	ASD, NPD

MASAM	Development Process of Mobile Application SW Based on Agile Methodology	2008	XP, RUP, SPEM
SLeSS	A Scrum and Lean Six Sigma Integration Approach for the Development of Software Customization for Mobile Phones	2011	Scrum, Lean Six Sigma

A. Mobile-D

One of the pioneering studies in agile approach is by Abrahamsson et al. [17], where it was assessed that agile development solution provides a good fit for mobile application development environment and proposed a new approach called Mobile D. The research provides an overview on to the mobile application development that makes it challenging and how these special characteristics and limitations affect mobile software development process. The study also introduces a software development approach drawn from the field of agile software engineering, which is designed to meet the specific demands of extremely volatile mobile environment. A new architecture concept called Architecture Line has also been introduced that aims producing an application framework, which guides the development of future mobile applications.

Mobile-D approach is based on Rational Unified Process RUP (life-cycle coverage), eXtreme Programming XP (development practices) and Crystal methodologies (scalability). Briefly the methodology of Mobile-D comprises of five phases, each of which has a number of associated stages, tasks and practices: Explore, Initialize, Productionize, Stabilize, System Test and Fix. The software development project that follows Mobile-D approach is divided into five iterations. These phases are set-up, core, core2, stabilize and wrap-up. Each of these phases consists of three different types of development days: Planning, Working and Release Day. The Mobile-D is organized into a framework that conjoins the main processes (plan, design, implement, test, release) with the support processes (project management, software configuration management, software process improvement).

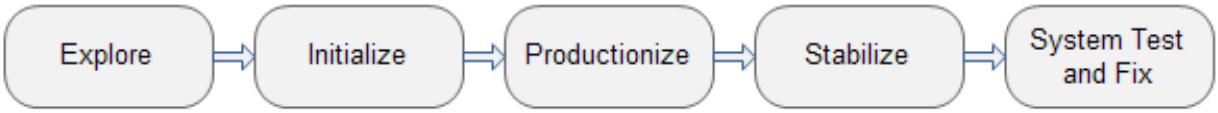

Figure 1: Phases of Mobile-D Software Development Process [17]

The 9 principal elements of Mobile-D are Off-Site Customer (Requirements), Phasing and placing in Planning Day (Planning), Agile Modeling (Modeling), Architecture Line (Architecture), Time, size and defect (Metrics), RaPiD7-method (Documentation), Agile Software Process Improvement (Improvement), User-Centred Focus End-users), Mobile Test-First development (Testing).

The Mobile-D approach has been empirically tested in development projects and has been successfully assessed against the CMMI level 2 certification. It is recommended to use Mobile-D by a small co-located team of at most ten co-located developers, working in a short development cycle towards a product delivery within 8 to 10 weeks of calendar time. Although this methodology being a pioneering study in the field seems very promising and plays an important role in theory, it is important to mention that this approach is cursory and not completely defined in order to be literally used in practice. Also, further improvements on it, have been suggested by other authors and the model could further be improved using hybrid agile techniques.

Spataru [18], evaluated the suitability of Agile methods for mobile application development projects, bringing a set of improvements to an established Agile via Mobile-D methodology, and providing tool support to enable these improvements, facilitating performance testing, usage logging and automatic test case generation. The researcher found importance of post-release that was not provided by Mobile-D and addressed the issue by extending the methodology in terms of lifecycle coverage, and by adding a newly evolved phase that deals with continuously integrating end-user feedback on the delivered product into future releases. The improvements brought to Mobile-D as proposed by Spataru should undergo experimental validation in a real organization. Such an experiment would require extensive work to be performed in collaboration with development teams working with different methodologies, and analysing if the envisaged benefits of the improvements described are met when applied on real projects.

B. RaPiD7

Working software has been given a focus over comprehensive documentation; however the required documents should be identified and documented. Dooms et al. has proposed a method called ‘RaPiD7’ (rapid production of documentation, 7 steps) that improves the documentation work without scarifying the quantity and quality of documentation [9]. RaPiD7 describes how human interaction is planned in software projects and how documents are to be created in facilitated workshops. The survey data RaPiD7 is

presented to speed up the planning work, improve the quality of the results and to enable more efficient information sharing. Moreover, metrics from a case study show how RaPiD7 has reduced significantly the number of fatal findings in inspections and provides a method that amends the planning activities in software engineering and the experiences show evidence of the improvements.

RaPiD7 provides a three-layer structure: Project, Case and Workshop layers.

Table 2: Three layer structure of RaPiD7 [19]

Layers	Description
Project Layer	Describes how human interaction and joint decision-making is planned for software projects.
Case Layer	Describes how the selected cases such as documents are to be created in consecutive workshops.
Workshop Layer	Describes how the actual work is carried out in form of facilitated workshop, using seven steps of method.

The workshops are planned in detail (1. Preparation Phase), initiated properly (2. Kick-off Phase) and different idea gathering (3. Idea gathering Phase), problem solving (4. Analysis Phase) and decision-making (6. Decisions making Phase) techniques are used in the workshops. The decisions are written down on a desired level (5. Detailed design Phase), and the workshop results are verified and the next steps are agreed on (7. Closing Phase).

Figure 2: Seven steps of RaPiD7 Workshop Layer [19]

RaPiD7 supports all software development projects, whether related or unrelated to mobile application development, but the case study experiments was performed in major mobile companies such as Nokia and Philips. This method has been experimentally tested at Philips Digital Systems Laboratory and it was developed within Nokia in the 2002-2003 timeframe. The motivation is to make documents that match reality and create them with as little effort as possible. RaPiD7 approach embraces two very agile practices: Whole Team & Do the Simplest Thing That Will Work. RaPiD7 improves the traditional approach for specification work by offering a way to plan the human interaction in the early phases of software projects and by providing means to make decisions and to document authoring jointly with a built-in quality assurance.

C. Hybrid Methodology Design

Rahimian and Ramsin [20] presents a different approach to the Mobile-D. They proposed a hybrid Agile and risk-based methodology that generates a method suitable for mobile application development designed from Methodology Engineering techniques. This is a discipline concerned with creating methodologies suitable for different development scenarios, motivated by the belief that no single process fits all situations. It is based on a combination between agile methodologies, Adaptive Software Development (ASD) and New Product Development (NPD).

The ideal mobile software development characteristics that the hybrid engineering methodology is based on are: agility, market consciousness, software product line support, architecture-based development, support for reusability, inclusion of review and learning sessions, early specification of physical architecture. The Hybrid Methodology Design process has been developed as a top-down, iterative-incremental process comprised of the following tasks: prioritization of requirements, selection of the design approaches to be used in the current iteration, application of the selected design approaches, revision, refinement and restructuring of the methodology built so far, defining the abstraction level for the next iteration, and finally the revision and refinement of the requirements, prioritizing them for the next iteration.

Table 3: Phases of Hybrid Engineering Methodology [20]

Hybrid Engineering Methodology Phases	
Idea Generation	
Project Initiation	Preliminary Analysis
	Business Analysis

Analysis	Detailed Analysis
	Creation of Functional Prototype
Design	Architectural Design
	Detailed Design
Implementation (Development Engine)	Adaptive Cycle Planning
	Concurrent Component Engineering
	Updates to Component Library
Test	Quality Review
	Market Testing
Commercialization	

Briefly, the proposed mobile development methodology was created in four iterations, starting from a generic software development lifecycle. In the first iteration, the methodology was detailed by adding practices commonly found in agile methods. Taking into account market considerations, the second iteration included activities from New Product Development, a process concerned with introducing a new product or service to the market. In the third iteration, Adaptive Software Development (ASD) ideas were integrated into the methodology, while in the fourth & final iteration, prototyping was added to mitigate likely technology-related risks.

The hybrid method engineering development framework takes into account most of the issues identified in related work in the field. However, the methodology is still at a high-level, and no specific tasks for the identified stages have been provided. The published material on Hybrid Engineering Methodology does not include any case study or shows that the methodology has been empirically tested on developing an actual mobile software product.

D. MASAM

Jeong et al. [21] proposed the Mobile Application Software Agile Methodology (MASAM) that provides the process for developing the mobile applications swiftly using an agile approach. It is based on Extreme Programming (XP), Agile Unified Process, RUP and the Software and Systems Process Engineering Meta-model (SPEM). It is GUI based architecture-centered that uses Agile approaches for rapid development and utilizes domain knowledge. MASAM shows a strong tie with the Mobile-D methodology and introduces slight variations, such as a project management and follow up tool coupled with Eclipse Process Framework.

MASAM is described according to Software and Systems Process Engineering Meta-model (SPEM) and defines following three kinds of process assets:

Table 4: Process Assets of MASAM [11]

Kind	Description	Name
Role	It defines a set of related skills, competencies and responsibilities of an individual(s).	Planner, Manager, UI designer, Developer, Development team, Initial development team, Tester, User
Task	It is an assignable unit of work assigned to a specific role. The granularity of a task is generally a few hours to a few days and usually affects one or only a small number of work products.	Product Summary, Initial Planning, User Definition, Initial Analysis, Select Resource, Select Process, Establish Environment, Write Story Card, UI Design, Define Architecture, Planning, Iteration plan, Face-to-face Meeting, Incremental Design, TDD, Refactoring, Release Plan, Feedback, Pattern Manage, Pair Programming, Integration, Acceptance Test, User Test Figure
Work Product	It is a general term for task inputs and outputs. There are three types of work product.	Product Summary, Project Planner, UI Sample, UI Model, UI pattern, Architecture Pattern, Application Pattern, Story Card, Task Card, Architecture Model, Component Model, Test Case.

MASAM proposes a mobile application development cycle comprised of four phases. The Preparation Phase defines a summary and a first notion of the product, and assigns roles and responsibilities. The Embodiment Phase focuses on understanding user's needs and defining the architecture of the software product. The last stage formed the Product Developing Phase, that benefits from traditional agile principles to furnish an iterative Extreme Programming development sequence. The implementation of the software product is carried out through Test-Driven Development, Pair Programming, Refactoring and Continuous Integration, with a close relationship with iterative testing activities. Finally, a Commercialization Phase concentrates on product launching and product selling activities.

Table 5: Phases of MASAM process [21]

Phase	Activity	Task
Preparation Phase	Grasping Product	Product summary
		Pre-planning
	Product Concept Sharing	User Definition
		Initial product analysis
	Project Set-up	Development process coordination
		Project resource coordination
		Pre study
Embodiment Phase	User Need Understanding	Story card workshop
		UI design
	Architecting	Non-functional requirement analysis
		Architecture definition
		Pattern management
Development Phase	Implementation & Preparation	Environment setup
		Development Planning
	Release Cycle	Release Planning
		Iteration Cycle
		Release
Commercialization Phase	System Test	Acceptance Test
		User Test
	Product Selling	Launching Test
		Product Launching

It is recommended to use the MASAM methodology for small companies that are focused on the development of mobile software applications. However the, authors have not presented a case study of an actual implementation of this methodology in a real-world environment.

E. SLeSS

Cunha et al. [22] proposed SLeSS; an agile approach integrates Scrum and Lean Six Sigma (LSS) that focuses on project management and process improvement respectively. The approach uses two types of product backlogs, Customization Product Backlog (for customizing development projects) and LSS Product Backlog (for process improvements).

The use of SLeSS assists in the easy adaptation to requirement changes in the later stages of the project and with less overall impact than the traditional approach, helps in meeting deadlines, reduces overtime hours, and delivers more rapidly versions and shortening the development cycle. Besides this, the use of the approach enables the achievement of performance and quality targets of real software development project, increases productivity, improves process quality, helps in cost reduction, progressively improves the development process, management process and the outcome of projects with fewer defects and failures.

Scrum is an agile methodology for project management and software development that adopts an empirical approach rather than prescriptive one and therefore it may be used in complex projects. On the other hand, Lean Six Sigma (LSS) is a methodology for defining and improving products, processes and services with a focus on reduction of defects and failures, on variation and waste elimination, prioritizing, in a planned and objective way, the achievement of quality and financial results. A six sigma level represents a process with 3.4 defects per million opportunities (DPMO).

1. Scrum in SLeSS

Scrum is widely used in software development, and it has been observed and documented in the scope of mobile software development [23]. The execution of SLeSS assumes an incremental approach by first implementing the Scrum alone and once the Scrum is well settled in the organization, LSS should be implemented as a quality framework. Initially, training team was chosen in the Deployment phase, aiming that the customization development would not suffer time and quality deviations which followed simulation of a real customization development in the team that was

already using Scrum. Later, Scrum was introduced to relative experienced team, with well-defined scope, activities and known risks. The recommendations for Scrum in SLeSS are:

- *Sprint Size:* It should be one or two weeks.
- *Team Size:* There should be four to nine people in a team.
- *Sprint Backlogs:* It contains customized activities and process improvement. The development team and client identify the problems or issues in Sprint which are prioritized and resolved by the team members on regular basis.
- *Lean Six Sigma (LSS):* It is important for Scrum Master and Product Owner to have proper understanding of LSS techniques, the development and management processes.

2. Lean Six Sigma in SLeSS

Once Scrum is well settled in the organization, Lean Six Sigma (LSS) is applied as a quality framework that complements Scrum as a development methodology. The model is represented by the 5-phase DMAIC phases (Define, Measure, Analyze, Improve and Control) as follows:

Table 6: DMAIC: 5 Phases of SLeSS approach [12]

	Phases	Backlog Items
1.	Definition Phase	LSS Project Contract
		Initial Analysis
2.	Measurement Phase	SIPOC Diagram (Supplier, Inputs, Process, Outputs, Customers)
		Process Map
		Cause and Effect Diagram
		Cause and Effect Matrix
		Impact Effort Matrix
		Initial Capability
		Measurement and Inspection Systems
3.	Analysis Phase	FTA (Fault Tree Analysis)
		Analysis of critical inputs of the processes
4.	Improvement Phase	Action Plan
		SIPOC
		Process Map
		Piloted Solution
		Final capability of the processes
5.	Control Phase	Control Plan

The SLeSS approach has been used in real embedded software customization development projects for mobile phones. The approach was experimented in P&D&I laboratory, with a mobile phone manufacturer as a client with team of 7-12 developers, in duration from 4-6 months, with average size of 529 LOC (Line of Code) developed in ANSI C programming language.

The practice of SLeSS facilitates in obtaining higher outcomes, such as better adaptation to changes in requirements, the fulfillment of the deadlines, the decrease in number of unplanned overtime and delivering more versions rapidly that too with fewer defects or failures. It demonstrates increase in productivity, improvement in process quality and reduction in cost. Besides this, the approach has allowed the improvement in development and management processes, initiating their statistical control and aligning them to meeting the customer goals and expectations.

VI. CONCLUSION

There has been significant increase in 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 softwares 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 of the identified business opportunity, a very few scientific publications can be found, which address the specific problems that the development organizations are facing while developing software for mobile devices. Although the proposed mobile application development models as discussed and reviewed in this paper are encouraging but still theoretical. These findings should undergo

experimental validation in a real organization to appreciate its results.

Beside the use of these proposed methods, there is a requisite to explore other agile approaches that can also be investigated and integrated with the development of mobile applications. This can be done by means of industrial surveys, interviews with mobile software developers, mobile project managers, concrete discussions with Agile experts and planning other experimental studies. It is expected that this review will assist in planning future work on investigating mobile application development process using hybrid collection of best agile approach which could further undergo experimental validation. Later, the conclusions from those studies could be recommended to mobile software development companies and researchers that would suitably adopt agile methodologies as a generic development culture without worrying about specific agile methodologies. In addition, it is anticipated that these studies 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.

VII. REFERENCES

[1] "Manifesto for Agile Software Development". Agile Alliance, 2001. [Online]. http://agilemanifesto.org/.

[2] A.I. Wasserman, "Software Engineering Issues or Mobile Application Development," Proceedings of the FSE/SDP workshop on Future of Software Engineering Research, FoSER '10, 397-400. doi: 10.1145/1882362.1882443

[3] R. Holler, "Mobile Application Development: A Natural Fit with Agile Methodologies". VersionOne, LLC. White Paper, 2011. [Online]. www.versionone.com/pdf/mobiledevelopment.pdf.

[4] P. Abrahamsson, J. Warsta, "New directions on Agile methods: A comparative analysis", International Conference on Software Engineering, pp. 244 – 254, 2003. doi: 10.1109/ICSE.2003.1201204

[5] P. Abrahamsson, "Keynote: Mobile software development – the business opportunity of today", Proceedings of the International Conference on Software Development, Reykjavik, Iceland, pp. 20-23, 2005.

[6] Kannan N. (2011). "Mobile development: Why using an agile methodology makes sense", September 2011. [Online]. http://searchsoftwarequality.techtarget.com/tip/Mobile-development-Why-using-an-Agile-methodology-makes-sense.

[7] Jordan B. Barlow, Justin Scott Giboney, Mark Jeffery Keith, David W. Wilson, Ryan M. Schuetzler, Paul Benjamin Lowry, Anthony Vance. "Overview and Guidance on Agile Development in Large Organizations", Communications of the Association for Information Systems Vol. 29, Article 2, pp.25–44, 2011.

[8] B. Boehm, R. Turner, "Balancing Agility and Discipline: A Guide for the Perplexed", Boston, MA: Addison-Wesley. pp. 55–57, ISBN 0-321-18612-5, 2004.

[9] K. Beck, "Extreme Programming Explained: Embrace Change", Boston, MA: Addison-Wesley. ISBN 0-321-27865-8, 1999.

[10] Kupe Kupersmith, "Agile is a Fad", [Online] http://www.batimes.com/kupe-kupersmith/agile-is-a-fad.html

[11] Christopher R. Goldsbury, The Agile Management Fad". [Online] http://anagilestory.com/2012/08/21/the-agile-management-fad/

[12] Luke Halliwell, "The Agile Disease", [Online] http://lukehalliwell.wordpress.com/2008/11/16/the-agile-disease/

[13] Martin Fowler, "Using an Agile Software Process with Offshore Development". [Online] http://martinfowler.com/articles/agileOffshore.html July 2006.

[14] What are pitfalls for agile development methodologies?, [Online] http://stackoverflow.com/questions/359790/what-are-pitfalls-for-agile-development-methodologies.

[15] Luis Corral, Alberto Sillitti, Giancarlo Succi, "Software Development Processes for Mobile Systems: Is Agile Really Taking Over the Business?" 1st International Workshop on Mobile-Enabled Systems (MOBS 2013) in connection with ICSE 2013, pp. 19-24. IEEE. 2013.

[16] Andrea A. Janes, Giancarlo Succi, "The dark side of Agile software development", Proceedings of Onward! 2012. pp. 215-228, 2012. doi: 10.1145/2384592.2384612

[17] Pekka Abrahamsson, Tuomas Ihme, Kari Kolehmainen, Pekka Kyllönen, Outi Salo, "Mobile-D: An Agile Approach for Mobile Application Development", OOPSLA 2004.

[18] Spataru A. C., "Agile Development Methods for Mobile Applications", Master of Science Thesis submitted to Computer Science School of Informatics, University of Edinburgh, 2010.

[19] Ko Dooms, Roope Kylmäkoski, "Comprehensive documentation made agile – experiments with RaPiD7 in Philips", In Proceedings of the 6th International Conference on Product Focused Software Process Improvement - PROFES 2005, pp 224-233. doi: 10.1007/11497455_19

[20] Rahimian V., Ramsin R., "Designing an Agile Methodology for Mobile Software Development: A Hybrid Method Engineering Approach", Second International Conference on Research Challenges in Information Science, RCIS 2008, pp. 337-342. doi: 10.1109/RCIS.2008.4632123

[21] Yang-Jae Jeong, Ji-Hyeon Lee, Gyu-Sang Shin, "Development Process of Mobile Application SW Based on Agile Methodology", 10th International Conference on Advanced Communication Technology, ICACT 2008, vol.1, pp. 362-366. doi: 10.1109/ICACT.2008.4493779

[22] T.F.V. da Cunha, Valeria L. L. Dantas, Rossana M. C. Andrade, "SLeSS: A Scrum and Lean Six Sigma Integration Approach for the Development of Software Customization for Mobile Phones", 25th Brazilian Symposium on Software Engineering, pp. 283-292, 2011. doi: 10.1109/SBES.2011.38

[23] Christelle Scharff, Ravi Verma, “Scrum to support mobile application development projects in a just-in-time learning context”, Proceedings of the 2010 Workshop on Cooperative and Human Aspects of Software Engineering - CHASE 2010, pp. 25-31. doi: 10.1145/1833310.1833315

How to cite

Harleen K. Flora, Dr. Swati V. Chande, "A Review and Analysis on Mobile Application Development Processes using Agile Methodologies ". *International Journal of Research in Computer Science*, 3 (4): pp. 9-18, July 2013. doi: 10.7815/ijorcs.34.2013.068