RESEARCH PLAN PROPOSAL

ADOPTING AN AGILE APPROACH FOR THE DEVELOPMENT OF MOBILE APPLICATIONS

For registration to the degree of

Doctor of Philosophy

IN THE FACULTY OF COMPUTER SCIENCE

THE IIS UNIVERSITY, JAIPUR

Submitted by

Harleen Kaur Flora

ICG/2011/13086

Under the Supervision of

Dr. Swati V. Chande

Professor (Computer Science),

International School of Informatics and Management

Department of Computer Science

July, 2012

INTRODUCTION 3

AGILE SDLC 6

AGILE TECHNIQUES 6

AGILE TECHNIQUES USED IN SOFTWARE DEVELOPMENT 7

MOBILE APPLICATION DEVELOPMENT PROCESS 8

REVIEW OF LITERATURE 9

MOTIVATION/JUSTIFICATION AND RELEVANCE 16

OBJECTIVES 20

OVERVIEW OF RESEARCH PLAN AND RESEARCH TASKS 21

PLAN OF WORK AND METHODOLOGY 22

REFERENCES, BIBLIOGRAPHY, WEBLIOGRAPHY 24

Introduction

Software development is a very complex field with myriad variables impacting the system. Software systems therefore cannot be built with mathematical or physical certainty. The more prevalent software development methods however have been designed assuming software development to be a very precisely specified activity which can be carried out in a phased manner with each phase leading to the other, and without much need to revisit any of the completed phases. These techniques assume that most of the project requirements can be accurately gathered at the beginning of the project which is far from being true for almost all software systems. 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. The success of lean development method of the 1980s inducted a broad class of "iterative" software methods including the Unified Process, Evo, Spiral, and Agile methods [1].

In recent years, with the rising competence of software market, researchers are seeking more flexible methods that can be used to adjust to dynamic situations where software system requirements are changing over time. The agile method [2] aims at facilitating software development processes where changes are acceptable at any stage [3], customers are treated individually as collaborators [4] and unavoidable evolutionary changes can be identified [5]. Agile methodologies provide a structure for highly collaborative software development. Commonly associated with 'lean' engineering, agile software development closely follows the flow of business value, with a focus on activities that directly contribute to the project end goal of quality software. The Agile manifesto [6], published to define the approach now known as agile software development process and is a guiding force for Agile practitioners, was created by 17 influential figures, some of which formed the Agile Alliance.

The Agile Manifesto established a common set of overarching values and principles for all of the individual agile methodologies at the time. It details four core values for enabling high-performance, efficiency and outputs [6]:

- Individuals and their interactions
- Delivering working software
- Customer collaboration
- Responding to change

These core values are supported by 12 principles which underlie the Agile Manifesto [7]:

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

The mobile applications market is currently witnessing rapid growth, as mobile platforms continue to advance in performance, and as the users' need for a wide variety of mobile applications increases.

Mobile Application Development is the process by which applications are developed for small low-power handheld devices such as personal digital assistants, enterprise digital assistants or mobile phones. These applications are either pre-installed on phones during manufacture, or downloaded by customers from app stores and other mobile software distribution platforms. In many respects, developing mobile applications is similar to software engineering processes for other embedded applications. Issues common to software engineering and mobile application development include: integration with device hardware, traditional issues of security, performance, reliability and storage limitations. However, mobile applications present some additional requirements that are less commonly found in traditional software applications, namely [8]:

- **Potential interaction with other applications** – Factory-installed software vs. numerous applications from varied sources, with interactions among them.
- **Sensor handling** – "smartphones", include an accelerometer that responds to device movement, a touch screen that responds to numerous gestures, along with real and/or virtual keyboards, a global positioning system, a microphone usable by applications other than voice calls, one or more cameras, and multiple networking protocols.
- **Native and hybrid (mobile web) applications** – software installed directly on the device vs**.** applications that invoke services over the telephone network or the Internet via a web browser and affect data and displays on the device.

- **Families of hardware and software platforms** – Execution of code that is custom-built for the properties of that device vs. support of applications that were written for all of the varied devices supporting the operating system, and also for different versions of the operating system.
- **Security** – most embedded devices are “closed”, in the sense that there is no straightforward way to attack the embedded software and affect its operation, vs. mobile platforms that are open, allowing the installation of new “malware” applications that can affect the overall operation of the device.
- **User interfaces** – with a custom-built embedded application, the developer can control all aspects of the user experience, but a mobile application must share common elements of the user interface with other applications and must adhere to externally developed user interface guidelines.
- **Complexity of testing** – while native applications can be tested in a traditional manner or via a PC-based emulator, mobile web applications are particularly challenging to test. Not only do they have many of the same issues found in testing web applications, but they have the added issues associated with transmission through gateways and the telephone network.
- **Power consumption** – many aspects of an application affect its use based on the device’s power or the battery life of the device. Dedicated devices can be optimized for maximum battery life, but mobile applications may inadvertently make extensive use of battery-draining resources.

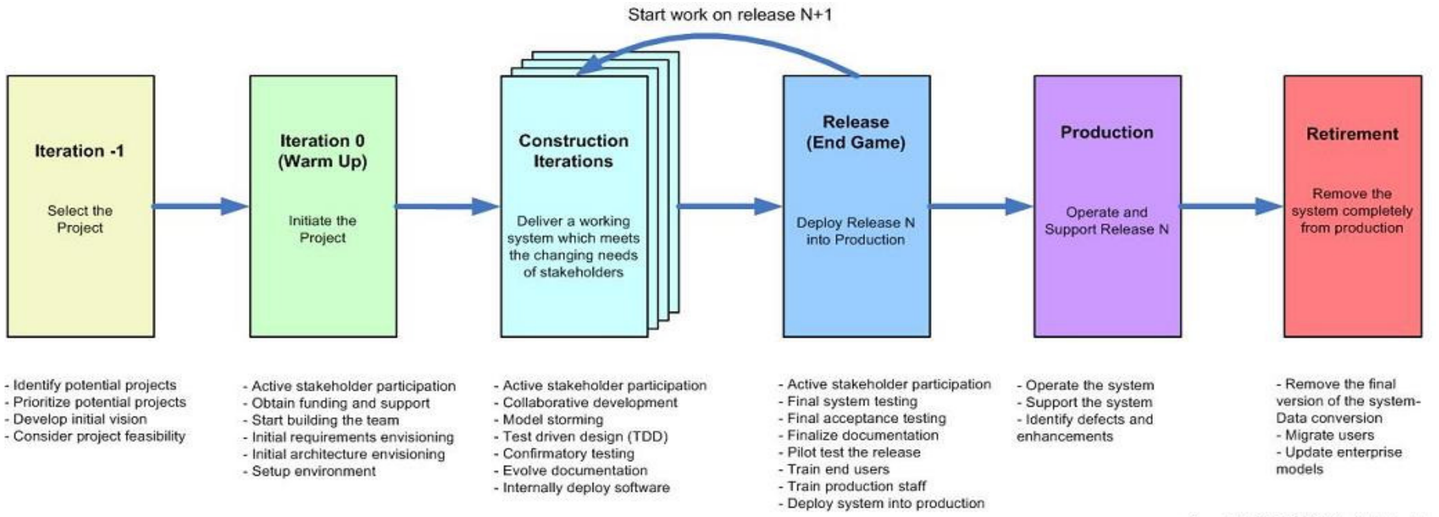
The rapid proliferation ubiquity of mobile and smart devices in the consumer market have forced the software engineering community to quickly adapt development approaches conscious of the requirements and novel capabilities of mobile applications. The combination of computing power, access to novel on-board sensors and ease of application transfer to market has made mobile devices the new computing platform for businessmen and independent developers. The rapid growth of this new computing platform has almost outpaced the software engineering processes tailored to mobile application development.

As is evident from the requirements of mobile applications, the development environment and the technologies that support mobile software are different as compared to traditional settings. Development teams therefore face the challenge of a dynamic environment, with frequent modifications in customer needs and expectations.

The changing need and expectations make the systems more complex. The more complex a project is and the more coordination it requires, the more formalisation in terms of processes is desirable. It is not possible to simply transfer the techniques of traditional software engineering one-to-one to mobile software engineering without significant modifications. The agile processes are considered to be very appropriate with software’s for fast-paced markets, where customer satisfaction is governed by early and frequent delivery, where there is scope for changes even late in the project, the delivery cycle is short (e.g. every couple of weeks), there is appropriate collaboration between businesses and developers, where working software is the primary measure of progress, where there continuous attention to technical excellence and good design and simplicity, which is definitely the case in mobile application development.

Agile SDLC

The below Agile System Development Life Cycle (high level) looks very much like a traditional SDLC, but when dived deeper, this isn't the case because the agile SDLC is highly collaborative, iterative, and incremental the roles which people take are much more robust than on traditional projects [27].


Start work on release N+1
Iteration -1
Select the Project
Iteration 0 (Warm Up)
Initiate the Project
Construction Iterations
Deliver a working system which meets the changing needs of stakeholders
Release (End Game)
Deploy Release N into Production
Production
Operate and Support Release N
Retirement
Remove the system completely from production
- Identify potential projects
- Prioritize potential projects
- Develop initial vision
- Consider project feasibility
- Active stakeholder participation
- Obtain funding and support
- Start building the team
- Initial requirements envisioning
- Initial architecture envisioning
- Setup environment
- Active stakeholder participation
- Collaborative development
- Model storming
- Test driven design (TDD)
- Confirmatory testing
- Evolve documentation
- Internally deploy software
- Active stakeholder participation
- Final system testing
- Final acceptance testing
- Finalize documentation
- Pilot test the release
- Train end users
- Train production staff
- Deploy system into production
- Operate the system
- Support the system
- Identify defects and enhancements
- Remove the final version of the system-Data conversion
- Migrate users
- Update enterprise models

Copyright 2006-2008 Scott W. Ambler

Fig. 1 Agile SDLC, www.ambysoft.com**, 2011) [27]**

Agile Techniques

Several Agile techniques have been proposed and used by researchers in difference domains. Some of the well-known agile software development methods include:

- **Extreme Programming (XP)** This methodology focuses on Engineering and is a system of practices that a community of software developers is evolving to address the problems of quickly delivering quality software, and then evolving it to meet changing business needs. The main characteristics of XP involves short iterations with small releases and rapid feedback, close customer participation, constant communication and coordination, continuous refactoring, continuous integration and testing, collective code ownership, and pair programming.

- **Scrum** is an empirical approach based on flexibility, adaptability and productivity in a volatile environment. Developers choose the specific software development techniques, methods, and practices for the implementation process and involve frequent management activities.
- **Agile Modeling (AM)** produces advanced models to support acute design needs and documentation purposes by keeping them as low as possible.
- **Crystal** includes a number of different methods from which to select the most suitable one and tailoring them for each individual project. Larger projects are likely to ask for more coordination and heavier methods than smaller ones.
- **Dynamic systems development method (DSDM):** Instead of fixing the amount of functionality in a product, and then adjusting time and resources to reach that functionality, it is preferred to fix time and resources, and then adjust the amount of functionality accordingly.
- **Adaptive Software Development (ASD)** is a method for the creation and development of software systems. ASD is part of rapid application development and focuses on rapid creation and evolution of software systems. It offers solutions for the development of large and complex systems by incremental and iterative development, with constant prototyping.
- **Feature-driven development (FDD)** focuses on the design and building phases, emphasizes quality aspects throughout the process and includes frequent and tangible deliveries, along with accurate monitoring of the progress of the project.
- **Internet-speed development (ISD)** addresses the problem of handling fast releases in short development cycles. The framework consists of time drivers, quality dependencies and process adjustments.

Some of these methods namely like XP, Scrum, ASD, Crystal have been used in mobile application development studies.

Agile Techniques used in Software Development

Following Agile techniques have been proposed by researchers in the development of software applications using combination of Agile and Non Agile techniques:

		Agile	Non Agile
1	Mobile D [13]	XP, Crystal	RUP
2	ME: Method Engineering Approach [15]	ASD	NPD
3	MASAM [16]	XP	RUP, SPEM
4	SLeSS [17]	Scrum	Lean Six Sigma
5	eXSCRUM [26]	XP, Scrum	
6	The Marriage of Lean, Scrum and Extreme	XP, Scrum	Lean

	Programming [28]		

Fig. 2 Software Application Development and various approaches

Mobile Application Development Process

While each mobile application has its own set of unique challenges, most projects follow the basic development process outlined below. This process ensures that your project is developed on time, in budget, within scope and to understand exactly what the app needs to do, what features need to be incorporated, and what our client's NEED versus what they WANT.

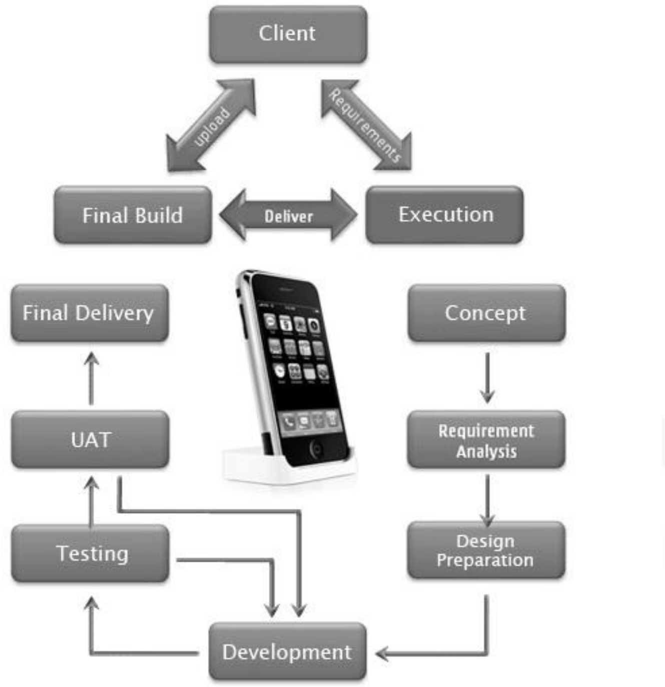


Fig. 3 Mobile Application Development Lifecycle

The development life cycle of a mobile application is defined in the steps below:

	Suitable Agile Technique
Concept / Scope	Scrum
Requirement Analysis	XP
Planning	Scrum/XP - Overlap
Design Preparation	Scrum
Development	XP
Testing	XP
Final Delivery	Scrum

Maintenance	XP

Review of Literature

The agile approach is seen as a natural fit for mobile application development [9][10][11], 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.

As per Nari Kannan [10], there are many reasons why Agile Software Development suites Mobile Application Development these include small teams, short deadlines, importance of usability, fast delivery and less complexity. The authors have suggested seven ways in which Agile development practices enhance the development of mobile apps. Agile development fits the experimentation and adaption nature of mobile apps, Agile increases reliability and leads to continued use of apps, Agile sprints extend naturally into mobile app update model, Agile enables mobile app development to be responsive to technology changes, Agile mobile development enables rapid accommodating of customer feedback, Agile mobile development enables thoughtful user experience, Agile mobile development enables phased roll out of feature sets.

According to Holler [9], for mobile development teams looking to introduce a lightweight development process or scale back more bureaucratic processes, agile software development offers tremendous opportunities and value. Progress in mobile computer technology and the rapid escalation of wireless networks in quality and quantity has brought in new applications and concerns in computer science and industry. The distinctive requirements and constraints associated with mobile systems have brought new challenges to software development for such environments, as it requires extensive improvements to conventional systems development techniques in order to fulfil the special needs of this field.

Abrahamsson [12] states reasons why Agile fits mobile software development. Below is the mapping of agile themes to traits observed in mobile software development:

Ideal Agile Characteristic	Rationale	Mobile software
High environment volatility	Due to high change of requirements, less need for up-front design & planning, need for incremental and iterative development approach.	High uncertainty, dynamic environment: Hundreds of new mobile phones published each year.
Small development teams	Small teams are able to react more rapidly, share information, less is documentation needed, etc.	Majority of mobile software is developed in micro or SME companies, or development teams.

Identifiable customer	To avoid business misunderstanding.	Potentially unlimited number of end-users. Business customer easier to identify, e.g. distributor.
Object-oriented development environment	Most tools that support Agile mode of development exist for object oriented development platforms.	E.g., Java and C++ used. Some problems in proper tooling e.g. for refactoring and test-first approach
Non-safety critical software	Failures do not cause loss of lives. More agility can be pursued.	Majority of existing mobile software is for entertainment purposes. Mobile terminals are not reliable.
Application level software	Large embedded systems require extensive communication & verification mechanisms	While mobile systems are complex and highly dependent, mobile applications can be stand-alone applications.
Small systems	Less upfront design needed	Size of mobile applications varies, but generally they are less than 10000 lines of code.
Short development cycles	For the purposes of rapid feedback	Development cycles vary. Generally mobile applications and services can be developed within 1-6 month time frame

Fig. 4 Mapping of agile themes to traits observed in mobile software development, 2005 [12]

Morris [11] has proposed a model for the mobile software engineering process which has three major phases:

Phase 1: Feasibility & economic efficiency analysis phase

- Requirements Engineering
- Design Drafting
- Early Prototyping
- User Acceptance Testing

Phase 2: Software Product Realisation

- Requirements Reviewing
- Design Detailing
- Defining Test Cases
- Programming

- Testing
- User Acceptance Testing

Phase 3: Distribution

- Marketing
- Preparing for Deployment
- Product Maintenance

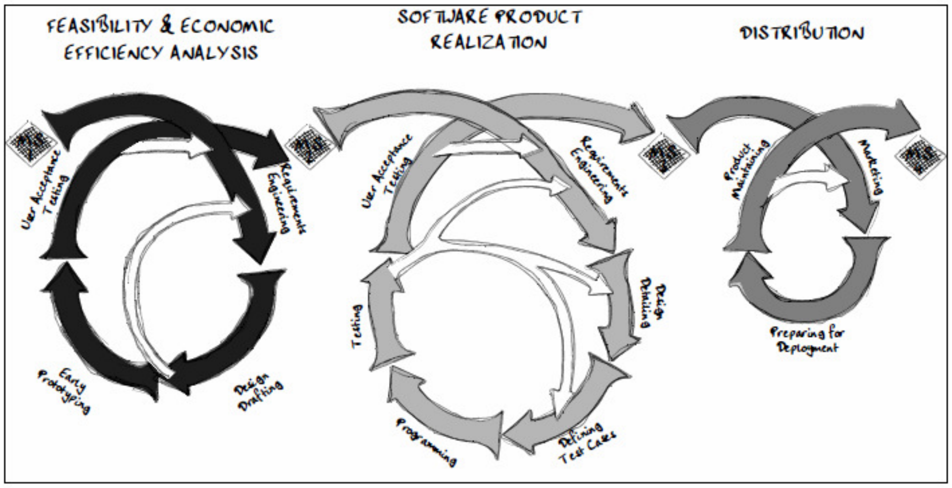


Fig. 5 A software engineering process model [11]

One of the pioneering studies in agile approach is by Abrahamsson et al. in 2004 [13], where they have assessed that agile development solution provides a good fit for mobile application development environment and proposed a new approach called Mobile D.

Briefly the methodology of Mobile-D comprises of five phases, each of which has a number of associated stages, tasks and practices. These include Explore, Initialize, Productionize, Stabilize and System Test and Fix. The Mobile-D process is used by a team of at most ten co-located developers, working towards a product delivery within ten weeks. There are nine main elements involved in the different practices throughout the development cycle: Phasing and Placing, Architecture Line, Mobile Test-Driven Development, Continuous Integration, Pair Programming, Metrics, Agile Software Process Improvement, Off-Site Customer, and User-Centred Focus. Mobile-D has already been applied in development projects, and advantages such as increased progress visibility, earlier discovery and repair of technical issues, low defect density in the final product, and a constant progress in development is observed.

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. The approach is based on Extreme Programming, Crystal methodologies and Rational Unified Process.

Spataru et al. [14] evaluated in 2010 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 and usage logging in the lifecycle. The researcher offered a number of improvements to the Mobile-D method, including a study in mobile application categories, related paradigms, end-user inclusion in the lifecycle, as well as performance testing of components and adoption of software product line principles. Besides this, the support tool enabling these improvements offers functionalities that include performance testing for Android components, usage logging and automatic test case generation. The intention of such contribution was to bring Mobile-D closer to an ideal mobile application development methodology, while providing useful features that can be outside the process, either in the form of practices or tools.

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.

Rahimian et al. [15], presents a different approach to the Mobile-D by proposing Hybrid Method Engineering that generates a method suitable for mobile application development. Methodology Engineering is a discipline concerned with creating methodologies suitable for different development scenarios, motivated by the belief that no single process fits all situations. Hybrid Methodology Design uses pre-established key requirements and conclusions as input, to iteratively generate the desired methodology.

In this study, the authors utilized as input a list of key methodology traits as well as conclusions from related work in the field. Each iteration of the method 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.

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 final iteration prototyping was added to mitigate likely technology-related risks. The final methodology phases proposed by the authors are presented as follows:

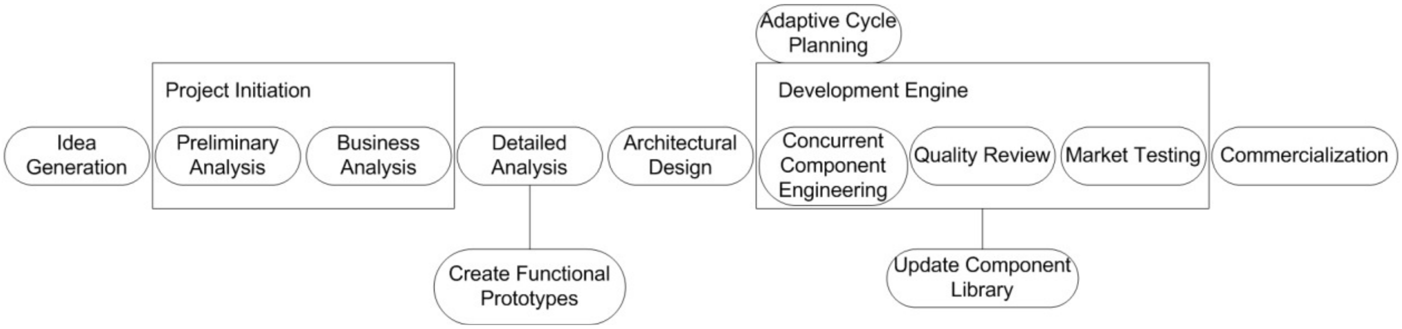


Fig. 6 Mobile development method proposed in (Rahimian & Ramsin, 2008) [5]

The proposed development framework takes into account most of the issues identified in related work in the field.

Jeong et al. [16] studied the development process of mobile application software based on Agile Methodology and proposed MASAM (Mobile Application Software Agile Methodology) that provides the process for developing the application SW operated on mobile platform. This methodology is based on Extreme Programming, Agile Unified Process, RUP and The Software and Systems Process Engineering Meta-model. It provides a series of principles from which different development processes can be defined according to the context of an agile software development company. In its structure and its detailed implementation, MASAM showed a strong tie with the previously released Mobile-D methodology and only introduce slight variations, for example a project management and follow up tool harnessed on the Eclipse Process Framework. Like Mobile-D, MASAM follows a software life cycle based on the Agile approach, underlining the importance of interaction among participants, communication with the customer, extreme development practices and continuous deliveries. In addition, MASAM also proposes a simple development cycle conformed by 4 phases, each one comprising a significant segment of the development process. The first stage comprises of a Preparation Phase that defines a summary and a first notion of the product, and assigns roles and responsibilities. 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.

Cunha et al. [17] in 2011 proposed SLeSS, an integration approach of Scrum and Lean Six Sigma used in real projects of developing embedded software customizations for mobile phones. This approach enables the achievement of performance and quality targets, progressively improving the development process and the outcome of projects.

Scharff et al. [18] studied in 2010 the use of Scrum in developing mobile applications and defined an innovative model of working with Scrum in a class setting at Pace University that involved certified Scrum Master of a the industry and a real product owner. The developed mobile application targeted the growing mobile market in Africa and the overall experience of adopting Scrum and its appropriateness in mobile development was studied in detail.

Hussain et al. [19] studied the process of integrating Extreme Programming XP (Agile methodology) with UCD (user centered design) and indicated how an Agile development technique facilitates to be user-oriented and at the same time preserves the social values of the development team, emphasizing iterative user-interface development involving usability engineers and end-users. The authors proposed that the success of a software development project is associated not only with tools and technologies, but also depends on how much the development process helps to be user-centered and developer-oriented. In their study, the authors examined that XP was a lightweight process that puts very little administrative overhead on the developers. Therefore, extending XP with additional practices was much easier than for other, more restrictive methodologies. The integration of usability engineering methods worked especially well because of the many overlapping principles (e.g. iterative development, end-user incorporation) of XP and UCD. The UI design process according to UCD was largely beneficial as it provided feedback which was used for the system's functional requirements. The assessment of each feature from the users' perspectives influenced the whole development process of the application and addresses the problems which arise when the system requirements are gathered only by discussions with stakeholders.

Norshuhada Shiratuddin and Sarif [20] proposed md-Matrix, a mobile application development tool which was designed and developed to assist mobile application developers especially the novice, to choose the methodology that suits the requirements of their mobile development projects. The authors studied that graduates entering the mobile development world deemed to understand the characteristics of mobile devices and applications affecting decisions about software design and be able to select and use appropriate standards, APIs and tool kits to build mobile applications. In view of that, the authors proposed an electronic decision matrix based on Pugh method to be used as one of the learning tool in mobile development course, especially for new developers, to choose the right methodology that suits the requirements of their mobile development projects instead of jumping straight to the development process blindly. The authors proposed that having the right methodology will help to spearhead the delivery process of the product and avoid unnecessary mistake.

Md-Matrix is an electronic version of a constructed decision matrix based on the Pugh method. In Pugh's method, decision matrix is prepared in a tabular format with the horizontal axis occupied by list of options and the vertical axis occupied by list of criteria or vice versa. This decision making tool is mainly aimed at assisting software developers especially the novice to choose the most appropriate development methodology for the development project of mobile applications.

Mushtaq et. al. [25] studied that a fine integration of management and engineering practices is desperately required to build quality product to make it valuable for customers and proposed a novel framework hybrid model to achieve this integration. The proposed hybrid model is actually an express version of Scrum model. It possesses features of engineering practices that are necessary to develop quality software as per customer requirements and company objectives. This model is enriched with the good features of both XP and Scrum by removing their limitations. Scrum does not provide much more about how to engineer a product while XP lacks in management practices. XP process model is mainly focused on engineering practices and Scrum focuses on management practices. The proposed model is a project management paradigm and it has the ability to produce quality software product that is aligned with customer requirements and company objectives. Scrum does not provide any direction about how to engineer a software product. The project team has to adopt suitable agile process model for the engineering of software.

Qureshi et. al [26] studied the phases of XP and Scrum models in order to identify their potentials and drawbacks and found out that XP model has certain drawbacks, such as not suitable for maintenance projects and poor performance for medium and large-scale development projects and Scrum model has certain limitations, such as lack in engineering practices. Majority of the software development companies are reluctant to switch from traditional methodologies to agile methodologies for development of industrial projects. A fine integration, of software management of the Scrum model and engineering practices of XP model called eXScrum is proposed to accumulate the strengths and remove the limitations of both models. The proposed model is validated by conducting a controlled case study which shows that the proposed integrated eXScrum model enriches the potentials of both XP and Scrum models and eliminates their drawbacks.

Motivation/Justification and Relevance

The review of the literature on Application of the agile approach to mobile application development brings forth the following observations which indicate substantial scope for further research in the domain.

- Although the work of Abrahamsson [12] on mobile software development seems to be very promising, the description that they provide of their Mobile-D approach is cursory and incomplete. This being a pioneering study, further improvements on it, have been suggested by other authors and the model could further be improved using hybrid agile techniques.

- As with all methodologies, the improvements brought to Mobile-D as proposed by Spataru *et al.* [14] 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.

- Rahimian and Ramsin [15] proposed hybrid method engineering development framework that 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 future work includes performing further iterations to obtain lower-level tasks in the process. In its current state, the methodology is more at theoretical level with Mobile-D.

- The development method MASAM proposed by Jeong *et al.* [16] is included in Mobile-D, with a minor original contribution in the methodological area. The use of a specific process framework means an important aid for design and project management, but the locale conditions of the tool make it unlikely to be widely extended. The MASAM methodology claims to be supportive for small companies 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 to appreciate its results.

With the increasing popularity and demand for mobile applications, people skilled in developing applications for mobile phones and tablets are highly in demand [22]. There has been significant increase in number of projects for mobile application development services; it’s a growing market which every business may require in the near future [23]. It is believed that agile innovations many offer a solution for mobile application and service developers who are in need of high quality development processes [8].

As can be summarized from the review, following approaches have been proposed for mobile application development:

	Mobile Development Methodology	**Agile**	**Non Agile**
1	Mobile D: An Agile Approach for Mobile Application Development (2005)	XP, Crystal	RUP
2	ME: Designing an Agile Methodology for Mobile Software Development: A Hybrid Method Engineering Approach (2007)	ASD	NPD
3	MASAM: Development Process of Mobile Application SW Based on Agile Methodology (2008)	XP	RUP, SPEM
4	SLeSS: A Scrum and Lean Six Sigma Integration Approach for the Development of Software Customization for Mobile Phones (2011)	Scrum	Lean Six Sigma

Fig. 7 Mobile Application Development and various Agile and Non Agile Techniques

Beside the use of above proposed methods, other agile approaches can also be investigated and integrated with the development of mobile applications. For example, individual mobile developers are currently following a Scrum like process during the mobile application development [8]. In addition, other researchers have also investigated and suggested that Scrum, as a project methodology have natural advantages in development of mobile applications based on having a disciplined and limited scope, high customer/end-user interaction, and condensed time to market cycles [18,24]. Besides Scrum, Feature Driven Development (FDD) is a requirement methodology that provides excellent practical advice if the group typically works on features. Another Agile approach is Dynamic Systems Development Method (DSDM) which has fundamental practices and emphasizes heavy prototyping [21].

Conventional software development methods have gradually been replaced by lightweight agile software development methods since the mid-1990s. This phenomenon is mainly due to the conventional methods' shortcomings, including a slow adaptation to rapidly changing business requirements, and a tendency to be over budget and behind schedule. The mobile telecommunications industry has shown to be comprised of a highly competitive, uncertain and dynamic environment. The potential number of different mobile applications is virtually unlimited. With the increasing popularity and demand for mobile applications, there has been significant increase in number of projects for mobile application development services. Agile innovations offer a solution for mobile application and service developers who are in need of high quality development processes. There is necessity to explore various agile methodologies for the development of mobile applications

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. Most of the work performed in this field has been focused on low-level (implementation-oriented) aspects of software development, while high-level (methodology-oriented) issues still remain to be properly addressed.

		PROS	CONS
XP	- XP is an Agile methodology where there is a high level of involvement between the customer and the development team. - The customer drives the development output by prioritizing the most important functions as user stories. - The development team delivers the user stories iteratively through continuous programming, testing, planning and close collaboration. - Working software is delivered very frequently. - It is very specific to software development and includes many engineering practices e.g, pair programming, continuous integration, test driven development. - XP focuses on Engineering and is based upon four values and 12 specific software development practices.	- Fast, aggressive delivery model; - High collaboration, - Minimal documentation up front.	- High discipline methodology, requires dedicated people from outside the IT department; - Requires that everyone on the team have intimate knowledge of XP to be successful; - Little documentation for handover, not suitable for maintenance projects and poor performance for medium and large-scale development projects.
Scrum	- SCRUM takes a more broad brush approach to building software than XP. - It is a framework for managing and controlling iterative work at the project level. In SCRUM, a 'product owner', works with both business and IT teams to identify and prioritize system wide functionality in the form of a 'product backlog'. - Various team members sign up to deliver shippable increments of working software in what is known as a sprint, which generally last up to 30 days. - Scrum is more focused on managerial skills of both	- Excellent control over scope creep; - Encourages teamwork and transparency; - Good visibility to management on development deficiencies ad therefore and therefore is adaptive; - Scales very well to	- Focused on course grained features; - Sometimes leads to quick-and-dirty' programming; - Little documentation for handover. - lacked in engineering practices

	managers and developers. It deals with project management principles and allows the team freedom to choose its specific development practices.	multiple teams and geographic locations.	

A hybrid collection of best agile practices from various methods such as XP, Scrum, DSDM, FDD, AUP, Crystal and Kanban as well as practices from traditional waterfall development methods will reduce the time-to-market, effort, cost of application development and deployment, and improve overall quality of mobile development project. The results of this research will contribute towards greater understanding of agile software development issues and should be useful to mobile application development companies that want to adopt agile methodologies as a generic development culture without worrying about specific agile methodologies.

After multilateral analysis and investigation of above mentioned agile approaches, it is assumed that these can be successfully implemented to enhance and evaluate the performance of mobile application development to improve the mobile application development process. Therefore the proposed Ph.D. work on the topic “Agile techniques for Mobile Software Engineering Development of Mobile Applications" will ascertain a significant and distinctive contribution towards the performance of the existing process of mobile application. Hopefully it would be a good quality contribution to the knowledge domain of mobile software engineering.

Objectives

The objectives of the proposed research are:

- To study mobile application development process using agile methodologies, such as XP, Scrum, Crystal an FDD. Since each agile approach is focussed on different aspects, having a comparison of all these agile methods would also prove valuable.

- To conduct a survey for gaining a better understanding of development practices for mobile applications and identify the problems and challenges faced by mobile professionals involved in making decisions related to application development.

- To investigate and implement the best fitting Agile approach for each phase of mobile software engineering process that accommodates the strengths while suppressing the weaknesses of each selected approach for strengthening the mobile development practices and addressing the identified issues and challenges.

- To propose an agile hybrid approach that integrates agile approaches, such as XP, Scrum, Crystal, FDD to meet the needs of volatile agile mobile application development for the assistance of mobile developers and managers. A flexible and extensible process map is proposed to be developed that will be capable of modelling and selecting suitable agile techniques for mobile application development projects which can be further tailored and implemented by mobile industries to their best convenience.

- To develop a new mobile application using the proposed agile hybrid approach. The proposed development of a mobile phone application can be an Education Application coded for the latest mobile phone.

- The proposed agile hybrid approach will be empirically tested and validated on latest mobile phone in real organization to appreciate its result. The results will be further evaluated by the process-based testing that will be measured in terms of duration, cost, resource, speed, complexity, accuracy, security, functionality, transportability, adaptability and flexibility before and after the application of proposed agile methodology.

Overview of Research Plan and Research Tasks

	Task	**Task Description**	**Start Date**	**End Date**
1	Agile Techniques	Study Agile methodologies recommended by various researchers for mobile application development such as Extreme Programming, Scrum, Crystal, Feature Driven Development, Adaptive Software Development, Dynamic System Development Method, Agile Unified Process, Kanban and Lean Development	July, 2012	Aug, 2012
2	Mobile Application Development	Study mobile application development process	Sept, 2012	Oct, 2012
3	Questionnaire	Prepare questionnaires for survey and interview	Nov, 2012	Nov, 2012
4	Survey	Conduct a survey for gaining a better understanding of development practices for mobile applications and identify the problems and challenges faced by mobile professionals involved in making decisions related to application development.	Dec, 2012	Feb, 2013
5	Survey Analysis	Conduct data collection, analysis and interpretation	Mar, 2013	Apr, 2013
6	List Issues and Challenges	After survey analysis and review of literature - define the desired performance, describe actual performance, describe the performance gaps and find out the root cause and problem mobile application development industry is currently facing.	May, 2013	May, 2013
7	Select elements for designing a model	Investigate the best fitting Agile approach for each phase of mobile software engineering process that accommodates the strengths while suppressing the weaknesses of each selected approach.	June, 2013	July, 2013
8	Design a hybrid agile model	Propose a hybrid agile development approach that integrates agile approaches to meet the needs of volatile agile mobile application development for the assistance of mobile developers and managers. After collection of data from various sources and to add value to mobile software development process, broadly in terms of cost, time and effort, a flexible and extensible process map will be constructed that will be capable of modelling and selecting suitable agile techniques for mobile application development projects	Aug, 2013	Oct, 2013
9	Develop/Test a mobile app	Develop a mobile application in Java programming language to test the proposed method.	Nov, 2013	Jan, 2014
10	Case Study	Shortlist the mobile application development companies where the proposed method can be empirically tested and validated.	Jan, 2014	Jan, 2014
11	Perform experiment	The proposed hybrid model will be empirically tested and validated on latest mobile phone in real organization to appreciate its result. The results will be further evaluated by the process-based testing that will be measured in terms of duration, cost, resource, speed, complexity, accuracy, security,	Feb, 2014	May, 2014

		functionality, transportability, adaptability and flexibility before and after application of proposed agile methodology.		
12	Evaluate experimental results	Evaluate the experimental result before and after the application of proposed model. This can be further tailored and implemented by mobile industries to their best convenience.	June, 2014	July, 2014
13	Thesis	Thesis Outline Thesis draft write-up Thesis reviewing and editing	July, 2014	

Plan of Work and Methodology

Following is step wise plan to accomplish the proposed goal(s):

1. **Data Collection Methodology**

   This consists of analysing the current practices followed by mobile application development companies by reviewing the literature and by taking responses of mobile application developers, managers etc. By developing and using an online questionnaire by collecting the problems and challenges faced by mobile software development companies. Interviews with mobile developers and mobile project managers will also be conducted to gather information regarding the topic.

2. **Tools**

   The endnotes made during the review of the literature, on-line questionnaire and interviews, etc. will be collected as separate variables and data collected through survey and responses will be analysed using statistical software packages such as SAS, SPSS and Stata. Preliminary analysis of data can be done also by using a spreadsheet program such as Microsoft Excel for entering, coding, sorting data and comparison of results.

   The use of software like DIA, which is free open source software alternative to Microsoft Visio can also help to create a process map that is required for the successful completion of the proposed research.

   There are also some free tools that can be used for mobile testing objectives such as 'FoneMonkey 5' which is the only tool for iOS that records all actions with the iPhone or iPad (while in use) and plays them back as a test script at any time and enables interactive creation, editing and playback of automation scripts that exercise an application's user interface; 'W3C mobileOK Checker' and performs various tests to determine level of mobile-friendliness, etc.

3. Methodology envisaged for a theoretical work

After collection of data from various sources, it will be analysed using software's like SPSS, MS Excel etc. A model shall be developed using a hybrid/improvised Agile technique and to add value to mobile software development process, broadly in terms of cost, time and effort, a process map will be constructed by selecting the most suitable Agile approach for each phase of mobile software engineering process. This can be further tailored and implemented by mobile industries to their best convenience.

4. Experimental Techniques

The new hybrid agile approach is proposed be developed to help solving the problem of mobile application developers and mobile project managers:

- By integrating mobile software engineering process with most suitable agile approach, a flexible and extensible process map could be developed. This process map will be capable of modelling and selecting suitable agile techniques for mobile application development projects.
- The proposed methodology will be empirically tested in mobile software development organization. The process-based testing will be measured in terms of duration, cost, resource, speed, complexity, accuracy, security, functionality, transportability, adaptability and flexibility before and after application of proposed agile methodology.
- A case study will be conducted to show the application of the proposed methodology in mobile development projects.

5. Errors in Experiments and Accuracy of Results

It will depend on contemporary situation.

6. Validation of Results

The collected data will be validated against any biases or misinterpretations.

Place of Work and Facilities Required

The research proposal demands responses to the questionnaire from Developers and Managers of few mobile development companies which can be taken through contact in person and may be supplemented by online questionnaires. The workplace would be the computer laboratory of SBBS Post Graduate College for designing of the questionnaire for field work and analysis after taking responses from about 100 persons. The proposed improvement through the proposed Agile technique will be empirically tested in 3 to 5 mobile companies.

References, Bibliography, Webliography

[1] Victor Szalvay, “An Introduction to Agile Software Development”, Danube Technologies, 2004. http://www.danube.com/docs/Intro_to_Agile.pdf.

[2] A. Cockburn, “Agile Software Development”, Addison-Wesley, 2001, R. C. Martin, Agile Software Development, Principles, Patterns, and Practices, Prentice Hall. 2002.

[3] L. Williams, and A. Cockburn, “Agile Software Development: It’s about Feedback and Change”, IEEE Computer, Vol. 36, No. 6, pp. 39-43. 2003.

[4] The Agile Alliance. http://www.agilealliance.org/.

[5] J. S. David, W. E. McCarthy, and B. S. Sommer, “Agility – the Key to Survival of the Fittest in the Software Market”, Communication of the ACM, Vol. 46, No. 5, pp.65-69. 2003.

[6] Beck, Kent, "Manifesto for Agile Software Development". Agile Alliance. 2001. http://agilemanifesto.org/.

[7] Agile Alliance. “Principles behind Agile Manifesto”. www.agilemanifesto.org/principles.html. 2006.

[8] Wasserman, A.I., “Software Engineering Issues or Mobile Application Development,” Proc. Workshop on Future of Software Engineering Research. 2010.

[9] Robert Holler. “Mobile Application Development: A Natural Fit with Agile Methodologies”. VersionOne, LLC. 2011. White Paper: www.versionone.com/pdf/mobiledevelopment.pdf.

[10] Nari Kannan. “Mobile development: Why using an Agile methodology makes sense”. http://searchsoftwarequality.techtarget.com/tip/Mobile-development-Why-using-an-Agile-methodology-makes-sense. September 2011.

[11] Ben Morris. Book on a Software Engineering Model. “A Software Engineering Model for Mobile App Development”, http://static.bada.com/contents/blog/20110616/20-introduction_to_bada_app03.pdf.

[12] Abrahamsson, P. “Keynote: Mobile software development – the business opportunity of today”, Proceedings of the International Conference on Software Development, Reykjavik, Iceland. 2005.

[13] P. Abrahamsson, A. Hanhineva, H. Hulkko, T. Ihme, J. Jäälinoja, M. Korkala, J. Koskela, P. Kyllönen, and O. Salo, "Mobile-D: An Agile Approach for Mobile Application Development," presented at OOPSLA 2004, Vancouver, Canada. 2004.

[14] Andrei Cristian Spataru. “Agile Development Methods for Mobile Applications”. Master of Science Thesis submitted to Computer Science School of Informatics, University of Edinburgh. 2010.

[15] 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, 2008. RCIS 2008. , (pp. 337-342). Marrakech. 2008.

[16] Yang-jae Jeong, Ji-hyeon Lee, Gyu-sang Shin. “Development Process of Mobile Application SW Based on Agile Methodology”, presented at Electronics and Telecommunications Research Institute, Seoul. 2008.

[17] Thiago Ferraz V. da Cunha, Valéria Lelli Leitão Dantas, Rossana M. C. Andrade: “SLeSS: A Scrum and Lean Six Sigma Integration Approach for the Development of Software Customization for Mobile Phones”. SBES. 2011.

[18] Scharff, C and Verma, R. “Scrum to Support Mobile Application Development Projects in a Just-in-time Learning Context”, Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), Pages 25-31. 2010.

[19] Zahid Hussain, Martin Lechner, Harald Milchrahm, Sara Shahzad, Wolfgang Slany, Martin Umgeher, and Peter Wolkerstorfer. “Integrating Extreme Programming and User-Centered Design”. In PPIG 2008, The 20th Annual Psychology of Programming Interest Group Conference, Lancaster University, UK. 10th - 12th September 2008, 2008.

[20] Norshuhada Shiratuddin, Siti Mahfuzah Sarif: “The md-Matrix: a learning tool in the mobile application development course”, IJMC 7(4): 494-514. 2009.

[21] White Paper. Three Beacons. “Agile Methods Beyond Scrum!”. www.threebeacons.com/Agile-methods-beyond-scrum.aspx. April 2012.

[22] “Mobile apps developers in hot demand”. http://www.news.com.au/technology. April 2012.

[23] Webwire. “Increasing interest in mobile app development”. http://www.webwire.com. January 2012.

[24] Neek Alyani, Sara Shirzad. “Learning to Innovate in Distributed Mobile Application Development” Learning Episodes from Tehran and London. FedCSIS. 2011.

[25] Zaigham Mushtaq, M. Rizwan Jameel Qureshi. "Novel Hybrid Model: Integrating Scrum and XP", IJITCS, vol.4, no.6, pp.39-44, 2012.

[26] M. Rizwan Jameel Qureshi. “Empirical Evaluation of the Proposed eXSCRUM Model: Results of a Case Study”. IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 2. 2011.

[27] Ambysoft. “The Agile System Development Life Cycle (SDLC)”. http://www.ambysoft.com/essays/agileLifecycle.html#Cycle0. 2011.

[28] Geoffrey Bourne. “The Marriage of Lean, Scrum and Extreme Programming (XP): How to Align Agile Across an Organization”. http://www.agilejournal.com/articles/columns/column-articles/2878-the-marriage-of-lean-scrum-and-extreme-programming-xp-how-to-align-agile-across-an-organization. 2010.