Mobile APP development based on agility function

Mamta Pandey*, Ratnesh Litoriya, Prateek Pandey

Jaypee University of Engineering & Technology, Raghogarh, Guna (M.P.), India

mamta.pandey07@gmail.com

ABSTRACT. *Mobile applications increase at an epidemic pace, and their expansion activities are beneficial. Smartphone usage by peoples has proliferated over the last number of years, and it is expected that the utilization of mobile applications in different environments will continue to increase. In the case of mobile applications, existing process models are not found suitable. Hence, it is strongly required to customize the conventional software development life cycle approaches in such a way that could be suitable for handling frequently changing of the requirement of mobile based applications and to manage other dissimilarities of apps. This paper analyzes various view of desktop and mobile-based development and proposes a life cycle model which incorporates the aspects of extreme programming and feature-driven development (FDD) for mobile-based apps development successfully and also to calculate mobile apps agility factor (MAAF) for handling dynamically changing of customers requirement. A different aspect of agility feature for the proposed model calculates the degree of agility.*

RÉSUMÉ. *Les applications mobiles augmentent à un rythme épidémique et leurs activités d'expansion sont bénéfiques. L'utilisation du smartphone par les populations a proliféré au cours des dernières années et on s'attend à ce que l'utilisation d'applications mobiles dans différents environnements continue à augmenter. Le développement d'applications mobiles est une reproduction de l'ingénierie logicielle qui entre dans l'ère du marché de 2007 et qui est très populaire parmi les utilisateurs. Il existe différentes méthodologies disponibles pour les applications de bureau. Dans le cas d'applications mobiles, les modèles de processus existants ne sont pas adaptés. En raison de la mise au point d'une nouvelle technologie pour les applications mobiles, la complexité s'est accrue de jour en jour. Par conséquent, il est fortement nécessaire de modifier l'approche du cycle de vie du développement des applications pour postes de travail de manière à pouvoir gérer des modifications fréquentes des exigences des applications mobiles et pour gérer d'autres dissimilarités d'applications. Au cours des dernières années, le développement d'applications basées sur mobiles suit une approche de base non organisée. Néanmoins, il n'existe aucune approche systématique du cycle de vie disponible pour le développement d'applications. Pour le développement d'applications mobiles, il n'y a pas d'approche spécifique possible dans la littérature. Les chercheurs ont proposé certains modèles de développement d'applications, mais dans la plupart des cas, la littérature ne donnait pas une approche systématique du développement d'applications mobiles. Cet article analyse diverses perspectives de développement sur postes de travail et sur mobiles et propose un modèle de cycle de vie intégrant les aspects de la programmation extrême et du développement piloté par les fonctionnalités (FDD) pour le développement d'applications*

mobiles et permettant également de calculer le facteur d'agilité des applications mobiles (MAAF). pour gérer de manière dynamique l'évolution des besoins des clients. Le degré d'agilité calculé par un aspect différent de la caractéristique d'agilité pour le modèle proposé.

KEYWORDS: *app development, extreme programming, MAAF, life cycle model, requirement engineering.*

MOTS-CLÉS: *développement d'applications, programmation extrême, MAAF, modèle de cycle de vie, ingénierie des exigences.*

DOI:10.3166/ISI.23.6.19-44

1. Introduction

From a high-level point of view, there is no difference between developing conventional software and mobile software development. The conventional method used to determine and develop desktop applications will not work with mobile app development (Joorabchi *et al*., 2013). Fundamental steps of all types of software development are similar such as the gathering of requirements, design, programming testing, deployment, and maintenance. Mobile devices are different from desktop PCs, due to the different characteristic of mobile devices; desktop application development lifecycle approach can not directly apply for mobile app development without significant amendments. Determination of quality of software depends on the approach of software development followed by the developer. The priority of software development firms is getting satisfaction from customers and consumers. Unorganized and unsystematic approach to software developments will lead to various issues, unsatisfied consumers as well as the high cost of products. Android has bedelivered come the domina become nt platform for universally accessible information and applications of any kind and here is where the mobile based applications are coming into play. The need for mobile apps is increasing day by day; people would not live without using apps. Due to the popularity of apps among users, revenue of apps is also increasing. Therefore in large pleasure development of mobile applications is becoming a challenging task (Hammershoj *et al*., 2010). App developers are trying to improve mobile app development, but there is still a shortage of a well organized, scientific, and systematic approach for mobile-based apps (Nezerwa *et al*., 2015). Mobile-based applications differ from conventional applications from various points of view such as short development time, quick response, changing of requirement frequently, need of continuous maintenance, increasing of user demand day by day (Ei *et al*., 2016). The need for a systematic development approach, a particular deployment lifecycle methodology, quality assurance, equipment, and tools have been introduced by different software development framework (Habra *et al*., 2008). Since mobile apps are smaller than desktop apps, so a huge percentage of small software development firms have been included with approximately 5 to 20 employees working on a project (Galeano *et al*., 2016). The issue confronted by these firms is ad-hoc bases approach of mobile apps development, limited skilled developer lack of good adopted quality management and mobile-based app development industries cannot manage the cost, risk of applying a desktop apps development tools and technology. Over as long as eight a considerable

length of time number product particular architects brings get concentrate something like the improvement of mobile-based apps (Sambasivan *et al*., 2011). Mobile based apps developer applying conventional software approach on mobile-based apps development but due to poor suitability of conventional software engineering approaches, tools, and techniques for the development of mobile based applications. There are different varieties of development approaches have been recommended for software development such as waterfall, spiral, evolutionary and incremental, etc. In any case, these development methodologies are not suitable for creating mobile apps in small software firms (Stojanovic *et al*., 2003), due to the flexible nature of mobile apps, some agile methods proposed by different researchers for mobile app development (Putra *et al*., 2012; Solinski *et al*., 2016). Mobile apps come under the category of small enterprises, sometimes which demands replacement in desktop methodologies so the use of agile methodologies better option for app development (Heberto *et al*., 2015). Still, there is lacking applicability of existing methodology for mobile apps (Jiang *et al*., 2008; Clancy *et al.,* 2015). Now most of the software migrating to the mobile apps, conventional approaches has become unfit for portable devices software development. The existing methodologies are failing for facing the challenges due to frequent changes of requirement and maintenance (Summerville, 2000). Previous survey and researchers have been shown that the primary reasons for failing mobile apps unorganized methodology, poor designs, weak estimation effort and management (Kim *et al*., 2013; Litoriya *et al.*, 2013, 2014). Lack of organized development approach for portable device software development, firms and developers are unable to completely implement. It has been presented by previous study systematic and organized approach required for mobile apps development. The development and evolution of apps development lifecycle should be similar to desktop apps development. Sequences and selection of phases should be arranged in such pattern so that they can face challenges occurs in mobile apps (Nagappan *et al.*, 2016; Voas *et al.*, 2012). After giving a brief overview of the existing software development practices, this paper proposes a systematic and effective lifecycle process model for mobile-based apps development, also focus on the traditional (conventional) software development process and how can the phases of different methodologies be put together to facilitate small software industries develop a quality mobile based applications. Our model is a combination of extreme programming (XP) and feature-driven development (FDD). The requirement of mobile apps frequently changes so it is difficult to manage to the developer, here we proposed a novel concept mobile apps agility factor (MAAF), and using this factor we can calculate agility for managing dynamic change of requirement. We have also analyzed agility degree with the help of different agile feature (Rajput *et al*., 2014).

This paper is organized as takes after. Section 2-related work. Section 3-need for a lifecycle model for development of mobile applications. Section 4-proposed lifecycle model for mobile applications development. Section 5-calculating rank of different agile methods. Section 6-Conclusions and future work.

2. Related work

The mobile execution ecosystem introduces numerous constraints that are not existent in desktop PC's computing (Zhang *et al*., 2009). These limitations signify further cogitations to address through the various phases of the software development lifecycle (Gasimov *et al*., 2010). It foresees short lifecycles, short development cycles, limited hardware, frequently changing user demands, must be easily updateable, must download quickly, low price and large distribution channels like app stores. All mentioned those requirements need to have a link with the conventions that help to yield software able to accomplish in such a complicated.

Table 1. Analysis of previous works for mobile apps development

Sr. No	Reference	Year	Method
1	Jeong *et al.*	2008	Agile method used for mobile app development.
2	Rahimain and Ramsin	2008	Agile methodology with hybrid approach.
3	Hussain and Fernelay	2008	Goal Question metric approach.
4	Kmthan	2008	Develop high quality mobile apps by using systematic integration of pattern.
5	Sa and Carrio	2009	Authors have used three case study for introducing different challenges occur During apps development.
6	Dentass *et al.*	2009	Various requirement for mobile apps testing
7	Scharff C and Verma R	2010	Mobile-based project develop by using Scrum
8	Liu *et al.*	2010	Mobile app testing by adaptive random bases.
9	Cunha *et al.*	2011	Authors have proposed a customized software development approach for app development by using Scrum and Lean Six sigma integration method.
10	La *et al.*	2011	An efficiency-centric design methodology for mobile application architectures.
11	Kim and Park	2011	The 4-tier design pattern for the development of an Android application.
12	Nosseir *et al.*	2012	Spiral methodology used for mobile app development.
13	Amalfitano *et al.*	2013	Considering context events in event-based testing of mobile applications.
14	Flora *et al.*	2014	Different method of agile software development introduced by authors.
15	Vallon *et al.*	2015	Agile and Lean process model for mobile apps development validated by Australian Project.
16	Prasad and Hamsini	2016	Agile development methodology and testing for mobile apps.
17	Amasri *et al.*	2016	Hybrid Agile framework model for mobile apps development.

Software development lifecycle model is a pictorial presentation of phases, in which process seems like broken in small parts or sub-phases (Scacchi *et al*., 2011; Jalote *et al*., 2013). It presents a detail of a process from some unique point of view.

By using such kind of approach, formal and accurate description of software activities can be developed. As we know, traditional software development systems become complex and significant. Numerous software development lifecycle practices have been proposed to supervise the mechanism (Kim *et al*., 2013). Waterfall model is a trendy model among various models; there are different phases such as requirement engineering, design, implementation, verification and maintenance, and performing. However, this model has some limitations as it assumes requirement should be apparent at the starting of the project also we cannot move to the next stage before completion of the previous phase. As we know in mobile software requirements, so waterfall model is not suitable for developing mobile software (Pressmen, 2014). Sometimes the spiral model is also known as the Boehm model, shortcomings of waterfall model overcome by spiral model (Boehm, 1986). The spiral model is a sequential model; all phases are similar to waterfall model except risk analysis and anchor point. The cost of this model is high and need deeply skilled people for working on this model - features of the system defined and implemented on the bases of lower priority (Jawadekar, 2004; Garg *et al*., 2017). In the mobile software, changing of requirement frequently whereas in desktop apps not changing often. These phenomena deal with the agile approach.

In other words, we can say changing of requirement effectively deals with the agile approach. In waterfall model, it is difficult to manage. Activities are carried out in small phase in agile approach development based on regular feedback of customer, continuous maintenance, early delivery, adaptive planning and collaboration (Singh, 2015). Due to increasing software complexity and user involvement, adoption of agile methodologies increasing day by day. There are several methodologies in agile software development like extreme programming, Scrum, Kenben, Lean, Feature-driven development (FDD), Crystal, and Dynamic System Development Method (DSDM). Agile methodologies are suitable for small software firms due to higher involvement of users and quick feedback, but being iterative nature of agile approach, it cannot be applied without modification for apps development.

Our proposed mobile app development lifecycle model's aim to provide a systematic approach to app development. We consider the different phases of software development lifecycle model apply a novel approach to manage frequently requirement uncertainty of apps by using mobile apps agility factor (MAAF). Our method calculated MAAF and based on the result it will decide which software development life cycle model is suitable for app development according to the degree of agility.

2.1. Analysis of mobile-based apps

The mobile devices are different due to the ability to manage all types of information from audio to video and video to audio and transform a much practical way to correspondence data and information to meet business needs. There are millions of apps available in different categories in different digital distribution channels. The number of apps grown more than 2.5 millions dramatically (see Figure 1) and the mobile has become the dominant platform for accessing all kinds of

applications which are widely distributed across the digital distribution platform, such as Google play store, Blackberry plays store, iTunes play store and window play store, etc. The number of varieties of apps is growing day by day very rapidly. Our life has been changed by mobile and work at every level, and this trend has just started. The growth of downloads of mobile apps has shown in Figure 1 (Perez, 2016).

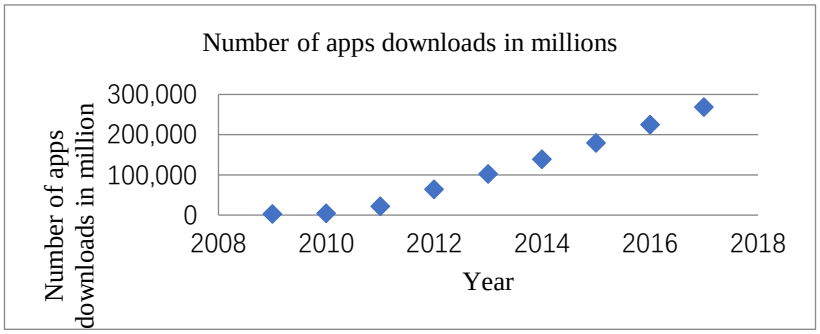

Figure 1. The growth of downloads of mobile apps

Now the app store model is taking over desktop computers, as we have seen in fig: 1 more than 250000 apps have been downloaded. A number of the mobile app has surpassed desktop apps to become the famous and dominating platform regarding entire time consumed (Chaffey, 2016). Various investigations have attempted to define mobile based applications. Robert and Michele defined mobile based applications as "complicated system" based on a diversity of mobile devices, languages, interfaces (Minali *et al*., 2016).

Different studies compare to mobile apps with desktop apps (Hart *et al*., 2016). We summarise these differences as follows:

1) Multi-Disciplinary Development Team- Mobile-based applications require the collaboration of larger teams of people with different expertise compare to desktop applications. An ideal team would comprise of mobile interface designer, content writer, usability engineers, tester and human-computer interaction professional.
2) Divergent Requirements- Due to agile nature and short development lifecycle period of mobile apps, changing of requirement frequently. In very short duration customer demand new requirement. Such as navigational, architectural, network, performance, etc.
3) Technology Visibility- As compare to the technology in desktop software applications, the technology of mobile apps much more visible to the users due to the distributed nature of the mobile, the immediately modifying view of the underlying technologies and the lightweight component-based structure of most mobile based systems (Inukollu *et al*., 2014). This signifies that the link between business architecture and the technical design of the system may be much tighter than for desktop software applications and errors are often not tolerated.

4) Variety of Users- Mobile apps are used by many remote users with a different background. Therefore, user interface design in mobile based applications needs to consider much broader concerns, i.e., cultivation, inventive than a desktop application.
5) Short Development Lifecycle- Because of the changing business environment rapidly changing mobile technologies and users requirements, mobile-based apps usually have shorter development cycles prescribed than desktop software applications.
6) Quick Start-up Time- One of the important characteristics of the mobile application is quick start-up time because of mobile devices used by users frequently and for short time durations. Whereas desktop software applications take long duration times, users are willing to agonize longer start-up times are required. For a mobile-based application, 20 seconds are enough for check or update some short piece of info.
7) Different Computer Architecture- Mobile gadgets have architecturally differed from the laptop, and desktop computers in those utmost mobile gadgets do not have a hard drive and often use the available RAM for both program execution and file storage.
8) Design- Application those works on desktop PC's without any interruption, they cannot work similarly on mobile devices. A well designed with good interface app work for a long time.
9) Operating System Design Issues- Operating system used in mobile devices is not multitasking. On portable gadgets, only one app can be work at a time if different app activated then it becomes problematic.
10) Screen Size and Orientation Issues- The mobile device has significantly less area to design the interaction that users can experience with the app. Mobile devices are also used in different situations than traditional computing devices are. The limited screen size is to carefully plan the user's interaction with the app. The screen should focus on one or very limited coherent, set of tasks that the user can or would want to do. Although the screen size is a nontrivial design issue, the fact that by default a screens orientation can change as the user turns the device issues.
11) Battery Issues- If every time app is used the user's device quickly becomes a brick. An app that quickly drains power will not get used will get bad reviews and eventually will not get downloaded at all. A global positioning system (GPS), camera, communication, and other sensors are all significant power draws.
12) Miscellaneous Issues- There is various difference identified between desktop applications and mobile applications by researchers who have mentioned in Table 2.

The complexity of mobile-based applications in industries has grown up immensely but currently most mobile apps developers often use unorganized hacker style approach (Eom *et al*., 2013). Some issues have been identified related to mobile apps which are listed in Table 3 (Perez, 2013).

Table 2. Difference between conventional and mobile-based application

Sr. No	Desktop Application	Mobile Application
1	Desktop apps have small user range.	There are various users of mobile apps.
2	User requirements are specific.	User requirements changes with time.
3	Growth and changes are ignorable.	Growth and changes are not ignorable, and it changes very fast.
4	Development budgets varying wide range according to the size of the company.	Development budgets are small.
5	Development time is longer.	Development time is small.
6	Less emphasis on the user interface.	More emphasis on the user interface.

Table 3. Different types of issues occur in mobile apps

Sr. No	% age of issues	Descriptions
1	63	Insufficient device coverage
2	62	Mobile apps crashed/froze/an error
3	58	User interface issue
4	52	Performance
5	50	Functionality
6	48	Lack of reliable automation
7	47	Need more time for testing
8	45	Device compatibility
9	29	Missing functionality
10	23	Operating system compatibility
11	14	Security
12	4	Other

Quality and standard of apps are critical in the absence of systematic development approach and its impact on the universe ranging from human being lives to public sector. It is very different and more severe from desktop software application (Harleen *et al.*, 2013). To address these concerns and challenges, many researchers suggest the need for an enhance certain and efficient approach for mobile based application development.

There are different categories of mobile apps available in the market, which have shown in Table 4 (Chaffey, 2016).

Table 4. Classification of mobile applications

Category	Domain	Key Characteristic	Implementations
Transactional	Communicational	Consultation according to need	Ebuddy, truphone.
	Financial	Money transformation	Paytm, etc
	Tracking & Tracking	Track the information or location of any product.	Mobile tracing, tracing of child
Content dissemination	Informational	Providing information according to the request of users	Weather, Google map, railway inquiry
	Broadcasting	Information delivery for marketing purposes	Cellfire
	Transformational	Transformation of data via the internet	Music play MP3, music player
Social networking	Networking	Facilitate the forming and maintenance of social relationships, and organizing social activities	Imo video chat, twitter, facebook
	Game	Collective-based leisure activities undertaken using a mobile device	Wazeer, Anthology
Personal productivity		Support work-related activities to improve personal productivity	MSOffice, Pdf converter
Tool		For personal leisure activities that do not involve social exchanges with others	Shazam – discover music, jango radio.

Increasing growth shows that mobile-based software has become challenging and complicated. The use and evolution of portable electronic gadgets have come up with new objectives for developers, consumers, and industries.

2.2. Evaluating existing methodologies for mobile based application

Mobile-based applications increasing day by day, previously many analyzers have identified that mobile apps are different from desktop apps. Wasserman defined that mobile based applications as “complex systems, based on a variety of hardware and software components, protocols, interfaces and standards, user experience, process, tools, architecture, and portability” (Anthony, 2010). Joshi defined mobile based application are different from conventional software such as families of software and hardware, security, user interface, the complexity of testing, power consumption (Nezerwa *et al*., 2015). Vijaya Lakshmi and Suganthi defined as the requirements for an app are usually derived from strategic business goals or market opportunities. During the development of an app, developers have limited contact with potential users. The number of download and revenues measures success is generated from apps (Kathuria *et al.,* 2015]. Mobile applications are interactive software which has

compels graphical user interfaces (GUIs). For example characteristics of mobile-based applications are network intensive and it belongs to the architectural issue of software development.

2.2.1. Waterfall model

Waterfall model proposed by Royce in 1970 (Roger, 2014). There is a series of phases in this model and provides a sequential approach to software development. Different aspects are connected progressively. Waterfall model is suitable for desktop software development because the requirement does not change frequently. This type of software development model is used for the projects which are no uncertain requirements. In waterfall model phases do not overlap.

The most prominent critique revolves around the fact that very often, customers don't really know what they want up front, and instead what they wish to emerge out of repeated two-way interaction over the course of the software. Another issue with the waterfall model is that moving on next phase in not possible without completion of the previous stage.

Once the first phase finished, the software system will be moved on the second phase. This approach is not flexible when developing complicated software with dynamic requirements as the traits of the model makes it complicates to modify the extended stages once they are concluded. Mobile applications are complicated software and requirement also frequently change, so the waterfall model is not suitable for mobile application development.

2.2.2. Spiral model

The concept of the spiral model is based on redo activity again and again. The schedules of the whole structure are not designed early. Features are defined and implemented on the priority bases then analyze the user's perception with this expertise. Features with lower priority determined and performed in a smaller piece (Boehm, 1988) Entire the necessary activities related to software development contained by each repetition. This model more emphasis on risk analysis into the life cycle. The spiral model has different four phases are: Planning, risk analysis; engineering and evolution. Risk analysis requires high expertise. The cost of the spiral model is top comparatively other models. It is not well suitable for small projects. The size of apps (in terms of LOC) is smaller than a desktop software application. The smallest app is Searchlight (Main, 2013) has less than 300 LOC and apps are cheaper than a desktop software application. The desktop software application development dominated by IT professional they have good knowledge of programming, database designing, and management of the project is mandatory. In contrast, mobile apps encompass a much variety of developers, an amateur with no suitable programming skills. Anyone can develop mobile apps without the necessity for programming knowledge (Brereton, 1998). So take care of things those mentioned above spiral model cannot be directly applied for the development of mobile based applications.

2.2.3. Agile software development

Under the agile methodology, there are various software development life cycle models. It is not a single process of software development such as Extreme programming (XP), Scrum, Crystal, Adaptive Software development (ASD), Lean Software Development and Feature Driven Development (FDD). Agile development approaches are logically fit for mobile app development. The concept of agile methodology is suitable for clarifying and getting to what customers like. The convention of the agile development method is that at legitimate intermission, the team emulates on how to become more efficient from feedback, and then accustom its behavior accordingly. Any particular way of agile development is not fitting for mobile app development. Among all the agile software development methods, Feature Driven Development (FDD) is one of the software development methods. It proves to be adequate on complex apps. The more complex apps are more effectual if FDD is applied. FDD is managing the mobile development project more directly and effectively. It holds well no matter the size.

There is a different variety of technologies available in the market for mobile app development such as HTML, CSS, and JavaScript, etc. FDD accommodates some excellent planning and reporting aspects that confirm to be useful in the app development process. In mobile apps development various types of risks such as frequently change of requirement, new technologies and underestimated complexity of non-functional requirements. This model helps in reducing risk using shorter iterations of designing, understanding of the requirements and the system where the apps to be developed.

3. Need for a lifecycle model for development of mobile-based applications

The existing unorganized approach of mobile apps development is the reason for mobile apps crises, failure or un-satisfaction of user. Some researchers have declared that the developer for mobile app development follows website or web apps development methodologies and also identifying success factors to building suitable lifecycle model for mobile gadgets (Kim *et al.*, 2013). Developers try to apply directly existing approaches without any modification for developing mobile based software which affects the success rate of apps (Li *et al*., 2016). Earlier, we have analyzed there are some valuable confirmation that mobile apps have different and special characteristics which are not present in desktop apps.

Is existing software methodologies are applicable for mobile-based app development? Answer of this question may be in two ways. First, current software development methodologies are suitable for mobile-based apps development. Developers are applying these approaches, but there is need of some changes in selection and sequence of phases. Another, various developer and researchers have objected that there is a need of lifecycle model for mobile-based apps which incorporates fundamental characteristics of traditional software development approaches. Mobile apps were first proposed in 2007 as a contemporary developing technology (Ihara, 1993). In the absence of proper selection of methodology, apps

user get irritated, frustrate again and again (Herman *et al*., 2016). The mobile apps came in the market era of 2007. There is various controversy on differences between mobile-based apps and desktop apps (Bavota *et al*., 2015). Mobile apps have different functionality and characteristics than desktop apps. Some articles said that mobile apps are not a perfect reproduction of desktop apps but the concept of development and phases acquired from desktop app development methodologies (Corral, 2013). Hence in this article, the mainstream has taken by us for app development. Apps development approach can be amplified pattern of desktop apps development in such a way that can meet various expectations of users, business need and exclusive functionality of apps.

4. Proposed lifecycle model for mobile applications development

The basic concept of proposed lifecycle method has taken from desktop apps development approach. Only selections of phases and sequences have changed.

Proposed model for apps development presented by us in Figure 8 which incorporates a degree of agility and MAAF.

The life cycle model contains various phases like requirement gathering, planning, feature list, plan by feature, graphics and user interface designing, coding, testing, deployment and acceptance, Testing based on platform, maintenance and collection of user feedback from users.

Case Study- Param ERP solutions Ltd is a small Indian IT company delivering governed and developing web and mobile-based services in software utilization. The firm used in case study have established in 2005 to create distinct web-based projects which are existed on its server along with the internet for their consumers, also started development of mobile-based apps in 2008, various apps developed by this firm available in Google play store. “Since the subsisting reporting suites were produced by the unorganized (ad-hoc) approach and "fix it for now" for any issue, the reporting suites are much more complex for maintenance and development. the method applied for the development of app “ooVoo Video Call, Text & Voice." Especially a particular concentration was on the requirement gathering, and testing based (Flora *et al*., 2014). Scrutinizing the practical nature of instruction relating to firm business and we are discussing some significant steps, but we are assured that the information commenced here is still sufficient to administer the complete summary of the case study.

4.1. Requirement gathering process

The first step of this model is requirement gathering and understanding the requirements. Requirements of the user are starting points and denote what the expectations of the mobile user from mobile software are. These requirements must be collected and maintained by use cases and user stories. Early prototypes can be used for this because users can get more information about users to need in the future. Even more advanced is the use of user interfaces (UIs) should be created relatively. The user interface in mobile devices is a big issue. Gathering requirements is not a

trivial task, in conventional software development there are various steps for elicitation of requirement and the purpose of developing conventional software for 2, 3 or limited users but the purpose of developing mobile software unlimited number of users. So it is very clear that requirements are extracted from app reviews. It is also a very challenging task. It depends on the quality of user reviews. App reviews should be conclusive or informative so that developer can decide for the further improvement accordingly - requirements for mobile apps quite distinct from conventional software. We analyzed requirement gathering from google play store platform targeted user reviews and cover different categories of mobile apps. The essential aim of this phase is to appear with different and advanced concepts or enhancements to the already existing apps. In case that consumer himself comes out with the notion, the notion is further evaluated. There are various apps of similar functionality available in the digital distribution channel so that we can take an idea from those apps.

4.2. Mobile apps agility factor (MAAF)

The uncertainty of requirement is the main concern of mobile apps. The risk of changing of requirement diversities is obvious hazards in mobile apps management. There is a need for flexibility so that frequently changing of requirement can be managed. It is possible only in agility. Here we have proposed agility factor known as mobile apps agility factor (MAAF), Rate of the inflexibility of requirement changing measured by MAAF.

The concept of MAAF has taken from entropy and motivated by thermodynamics. It is the best technique for measuring the disorder of any system (Okediran, 2014; Iacob, 2013).

Suppose any random variable Y with n outcomes {y_1..... y_n}, measurement of uncertainty can be expressed by H(Y) and is defined as

$$\mathrm{H(Y)} = \sum_{i=1}^{n} p(y_i) I(y_i) = -\sum_{i=1}^{n} p(y_i) log bp(y_i) \quad (1)$$

Where $p(y_i)$ is the probability mass function of the result, b is the base of the logarithm. The value of b is usually 2 or the constant 'e' or 10. When b=2 the unit of it is bit. When b=e, the unit of it is nat. When b=10, the unit of it is dit.

Characteristics of information entropy:

· Symmetrical characteristic: Despite the fact the progression of action in the probability system is distant, the entropy value of probability system persists unchanged, which means entropy rate of probability system has nothing to do with the progression of action.

· Non-negative characteristics: Entropy is a positive value because the distribution probability of random variable is $0<p<1$.

· Extremism characteristic: When the action in the system is an inflexible probability distribution the entropy value is maximum.

Some tuples have defined by us R=(M, Q, C) for the frequently changing of requirement, M indicates the requirement of mobile apps, Q denotes the anticipation of frequently requirement changes of Mobile apps, C means the count of impact of changes. N requirements have assumed here for mobile apps. We set them as M_1, M_2, M_3,…, M_n. Count of the impact of functional requirement frequently changes are C_1, C_2, C_3,…, C_n. The probability of requirement is Q_1, Q_2, Q_3,…, Q_n.

Count of impacts of the non-functional requirement is as below:

The value of the degree of requirements count of impact on the mobile app is $(MAAF)_1$, $(MAAF)_2$,…,$(MAAF)_n$

$$(MAAF)_i=C_{1i}*Q_{1i}+C_{2i*}Q_{2i} \text{ (i= 1, 2, …, n)} \quad (2)$$

We commenced normalization to deal with $(MAAF)_i$ then we obtain:

$$\rho_i = \frac{(\text{MAAF})_\text{i}}{\sum_{i=1}^{n}(MAAF)i} \text{ (i=1, 2, …, n)} \quad (3)$$

From the concept of theory of information entropy, we recognize:

$$H = -\sum_{i=1}^{n} \rho_i \log_b \rho_i \text{ (i=1, 2, …, n)} \quad (4)$$

According to the behavior of entropy, the major inflexible ρ_i is the higher the value of entropy (H). When $\rho_1=\rho_2=\rho_n=1/n$, we can obtain the maximum value $H=\log_b n$.

However, the value of Equation (2) may be higher than 1. To simplify the assessment, we commenced normalization to deal with H, we get:

$$H^*=H/H_{max}=H/\log_b n \quad (5)$$

$0 \leq H^* \leq 1$, the more inflexible ρ_i is, the greater value entropy H is, the minor risk is so that we can obtain:

$$R=1-H^* \quad (6)$$

$0 \leq R \leq 1$, Entropy-based model for amplification of quickly changing of requirement can be obtained from the interpretation of risk and information theory. This is given below:

$$R=1+\frac{\sum_{i=1}^{n}\frac{C_{1i}\times Q_{1i}\times C_{2i}\times Q_{2i}}{\sum_{i=1}^{n} C_{1i}\times Q_{1i}\times C_{2i}\times Q_{2i}}\log_b \frac{C_{1i}\times Q_{1i}\times C_{2i}\times Q_{2i}}{\sum_{i=1}^{n} C_{1i}\times Q_{1i}\times C_{2i}\times Q_{2i}}}{\log_b n} \quad (7)$$

4.2.1. Data collection

Our criteria for the selection of an app store for data collection is the Google Play store because most of the people using this platform. The Google Play store is a digital distribution outlet run by Google. There exist greater than 2.5 million apps including all categories available in this store, as we have mentioned in Figure 1.

Reviews of mobile apps are essential archives of opinions coming directly from precisely from the app end users. Such opinions span different issues, and according to Shannon (1948), 23.3% of them speak for a feature request, i.e., ideas through which user either advise advanced features for an app or express desires for the re-design of previously lying features of an app. Here we addressed the process of data collection. The strategy of data collection has shown in Figure 2.

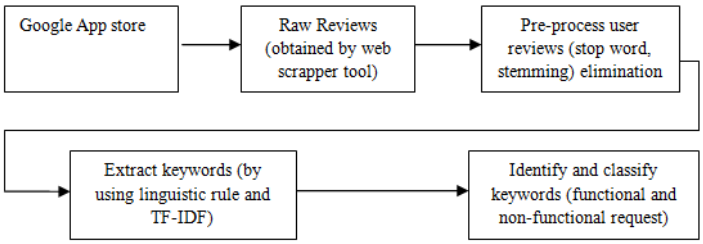

Figure 2. Processing phase of mobile apps requirements identification and classification

Based on the above analysis and with the help of MARK (Vu *et al*., 2015; Song, 2018). we identified functional and non-functional requirements of apps which are listed below:

The following keywords have identified for expressing functional requirements of mobile apps (Add, allow, complaint, could, hope, if only, improvement, instead of, lack, look forward to, maybe, missing, must, needs, please, prefer, request, should, suggest, waiting for, want, will, wish, would, not responding).

The following keywords have identified for expressing non-functional requirements of mobile apps (Battery, uninstall, crash, reinstall, freeze, drain, heat, expire, log, session, fail, consume, hog, hang, CPU, storage, overheat, memory).

We investigated the keywords related to the requirement on the bases of function points. According to the type of effect of user reviews, we divide requirements into four categories.

Severe Requirement-This type of requirement usually affects the whole app such as “apps are not responding" also change on speed and reliability of the apps. These requirements specify the substance effect of apps. They are related to the principal of programming, designing, and data structure and memory management of software engineering. It is a very significant task to affirm sever requirement at the beginning.

Moderated Requirement-These requirements are various issues and require in the characteristics segment. Such as - issues related to user interface, designing, etc.

General Requirement-It is a group of fundamental operation to conclude the work. These fundamental requirements are a general section that one of the various

issues and needs abstract. They are crucial for other subordinates. Network or connection related issues come under this category.

Least Important-In other words, we can say minor requirement, these requirements not need to satisfy too early in the app development phase. Security, maintenance related issues come under this category. Categorized requirements have listed in table 5, and the probability of those requirements has shown in Table 6.

Table 5. All types of requirements occur in app reviews

Sr.No	Type	No. of functional requirement change	No. of non-functional requirement change	Count
1	Severe requirement	58419	4986	831
2	Moderated requirement	12351	4462	472
3	General requirement	4986	3266	268
4	Least important	1563	600	69

Table 6. Probability of all types of problem change

Sr.No	Type	The probability of functional requirement change	The probability of non-functional requirement change
1	Severe requirement	0.01	0.16
2	Moderated requirement	0.03	0.1
3	General requirement	0.05	0.08
4	Least important	0.04	0.1

According to Equation (2), we can calculate A_1=1381.95, A_2=816.73, A_3=510.58, A_4=122.52.

According to Equation (3), we can compute $\rho1$=0.683, $\rho2$= 0.229, $\rho3$=0.081, $\rho4$=0.005.

According to Equation (4), we can obtain H=1.154.

According to Equation (5), we can calculate H_{max}, H_{max}=log2 1640=10.679 and H*=0.108.

According to Equation (7), we can calculate R=0.891.

Planning is steep level planning facet such as abstract types of components, relationships and functions are identified and artifices accredited to them, and then a list of phases and their preferences are determined for each stage. The value of MAAF will decide which model is suitable for mobile app development. The ranking of six agile methods (Scrum, XP, Crystal, FDD, DSDM, and ASD) on the bases of practices are 1, 2, 3, and 4, respectively. Ranking of XP and Crystal are equal.

If calculate MAAF is above these ranking. It means our proposed model will be suitable for mobile apps development.

4.3. Development process

The development process of any software is a series of phases described in the planning phase. So those increase the code reusability, scalability, maintainability, and robustness. In our model, we applied explicit phase risk analysis for each stage to outcome a schedule design consisting of interfaces, components, data structure, etc. The risk is the contingency that a product will deteriorate an event of a given adverse impact is determined from different aspects: including non-functional requirement, maintainability, and reusability. If any such kind of risk occurred during app development, then it will discuss with other developers, the project in charge in an informal way and resolve it quickly. When coding and testing of apps have been finished developer or tester investigate the gross system after that whole app tested by other people who are not associated in development process after that app is ready for deployment.

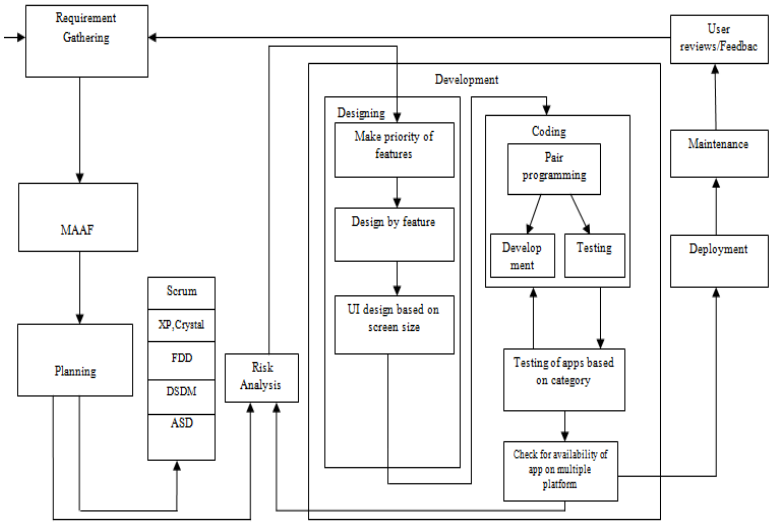

Figure 3. A lifecycle model for mobile based application development

After establishing a strategic foundation for an app, we move on to developing the app. In this step user interface, functional performance is significant. Firstly app developers make a priority of all those features which developer has panned in the planning phase. In the next step, i.e., design by feature, the chief programmer selects a small that is to be developed within minimum time. Class owner and chief programmer work out together and detailed sequence diagrams for each feature and refine the overall model. Next, the class and method prologues are written and finally a design inspection is held. Next step is user interface (UI) design based on screen size, initially app designed for the small screen size of a smart-phone or other handheld devices but now a day there is the various size of handheld devices are available in the market. Keeping these things in mind developer have to design multiple interfaces for multiple screen size of handheld devices.

4.4. Check the availability of the app on different platform

There are numerous mobile operating systems are available in the market.

4.5. Pair programming

Most of the time an app crashes due to sloppy coding, designing error made by human not allocating enough memory. In few cases that will, app crashes null pointer exception, wrong states, a feature that touches another feature. So there is need of pair programming so that a driver writes the code while the other observes to the code.

4.6. Testing of apps based on category

There are three categories of mobile apps available in the market: Native, web and hybrid apps. All apps cannot test similarly. It required different testing pattern for different category of apps.

4.7. Evolution process

4.7.1. Deployment

The deployment phase pledge with entire facet included in pushing the concluded apps out into the product market. Now app is ready to be released to the world after meticulous a development and testing duration. This is the point for broadcasting and marketing squad to step in and support publicity of exorbitant app statures. An excellent publicity crusade is essential to a successful app launch.

4.7.2. Maintenance

Last but not fewest after handover app to the hand of users provide the necessary support and amends. This phase is also known as post-development phase.

4.7.3. User feedback (Reviews)

User reviews of mobile apps usually consist of annoyance or opinions which are helpful for app developers to boost user involvement and amusement. In our proposed model there is a provision all kind of user reviews connected with requirement phase so that that developer could understand the need of users.

5. Calculating rank of different agile methods

Qumer &Sellers (Qumer *et al.*, 2015; Genco *et al*., 2018) proposed an analytical framework, 4-DAT, for calculating agility degree of six agile methods. Here we calculated the degree of agility based on agility characterization such as flexibility, speed, leanness, learning, and responsiveness. Features and description have given in Table 7.

Two steps can measure the degree of agility; the first one is phasing and second is practices or techniques. The agile practices research was used as the cause of the mathematical inputs for the investigation, determined using the following equation (Quamer *et al*., 2006).

$$DA(object)=(1/n)\sum nDA(object, phase\ or\ techniques) \quad (7)$$

Table 7. Agility characterization

Sr. No.	Feature	Description
1	Flexibility	Does the practice contain predicted or unpredicted modification?
2	Speed	Do the outcomes instantly?
3	Leanness	Does the method follow time span, use economical, uncomplicated and quality appliances for output?
4	Learning	Does the practice employ amended previous observation and experience to establish a learning environment?
5	Responsiveness	Does the method perform consciousness?

Table 8. Illustrative table for the computation of the agility of 'a' particular phases and 'b' techniques

	Features of agility					
Phases/Techniques	Flexibility	Speed	Leanness	Learning	Responsiveness	Total
Phases						
Phase 1	1 - 0	1 - 0	1 - 0	1 - 0	1 - 0	0 - 5
Phase 2	1 - 0	1 - 0	1 - 0	1 - 0	1 - 0	0 - 5
Phase 3	1 - 0	1 - 0	1 - 0	1 - 0	1 - 0	0 - 5
etc	1 - 0	1 - 0	1 - 0	1 - 0	1 - 0	0 - 5
Total	0 - a	0 -a	0 -a	0 -a	0 -a	(0-5)*a

Degree of agility	(0-a)/a	(0-a)/a	(0-a)/a	(0-a)/a	(0-a)/a	overall devided number of blocks in table
Techniques						
Technique 1	1 - 0	1 - 0	1 - 0	1 - 0	1 - 0	0 - 5
Technique 2	1 - 0	1 - 0	1 - 0	1 - 0	1 - 0	0 - 5
Technique 3	1 - 0	1 - 0	1 - 0	1 - 0	1 - 0	0 - 5
etc	1 - 0	1 - 0	1 - 0	1 - 0	1 - 0	0 - 5
Total	0-b	0-b	0-b	0-b	0-b	(0-5)*b
Degree of agility	(0-b)/b	(0-b)/b	(0-b)/b	(0-b)/b	(0-b)/b	overall devided number of blocks in table

The degree of Agility of Proposed Model has shown in Table 9.

Table 9. Degree of agility of proposed model

Features of agility						
Phases/Techniques	Flexibility	Speed	Leanness	Learning	Responsiveness	Total
SDLC Phases						
Requirement	1	1	0	1	1	4
Planning	1	1	0	1	1	4
Development	1	1	1	1	1	5
Design by feature	1	1	0	1	1	4
Maintenance	1	0	0	1	1	3
User feedback	1	1	1	1	1	5
Total	6	5	2	6	6	25
Degree of agility	6/6	5/6	2/6	6/6	6/6	25/(6*5)
Techniques						
MAAF	1	1	1	1	1	5
Pair programming	1	1	0	1	1	3
Design	1	1	1	1	1	5
Testing	1	1	0	1	1	4
Active user involvement	0	1	0	1	1	3
Iterative & incremental design	1	1	0	1	1	4
Total	5	6	2	6	6	25
Degree of agility	5/6	5/6	2/6	6/6	6/6	25/(6*5)

The degree of agility on the bases of phases=0.83

The degree of agility on the bases of techniques (practices)=0.83

5.1. Comparison

Comparison of Degree of Agility of Previous Agile Methods and Proposed Method has shown in Figure 9.

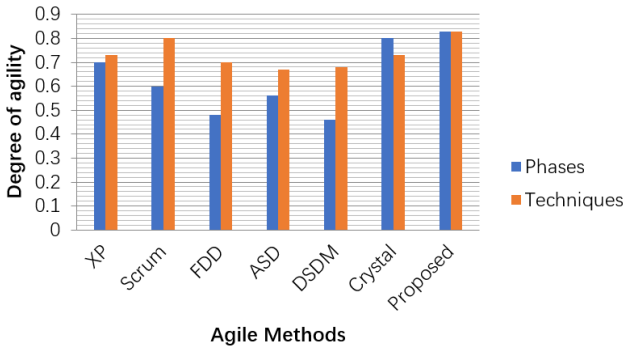

Figure 9. The degree of agility of different agile methods and the proposed method

6. Conclusions and future work

Above investigation of current desktop apps development practices and literature analysis, some investigation came out from the study that available methods cannot apply to mobile app development. As mobile based applications are more evolutionary, changing of requirement very frequently. Traditional approach (waterfall model and spiral model) are unable to manage such changing of frequently requirement, while agile practices are efficient to advocate persistent change but developing a mobile-based application over agile conventions and methods can outcome in undetermined iterations. Further, this paper determines professional and organizational characteristics between traditional and mobile-based development. MAAF has been calculated and the degree of agility determined. The product of the project has been released within time and budget and the architecture of the system is well organized which leaves sufficient scope for the future evolution and can be efficiently collaborated into another systems.

References

Almasri A. K. (2016). A proposed Hybrid Agile framework model for mobile apps development. *International Journal of Software Engineering & Applications*, Vol. 7, No. 2, pp. 1-9. https://doi.org/10.5121/ijsea.2016.720

Amalfitano D., Fasolino A. R., Tramontana P., Amatucci N. (2013). Considering context events in event based testing of mobile applications. *In the Proceeding of the 2013, Sixth International Conference on Software Testing Verification and Validation Workshops*, pp. 126-133. https://doi.org/10.1109/ICSTW.2013.22

Bavota G., Vasquez L. M., Cardenas E. C. (2015). The impact of API change- and fault-proneness on the user ratings of android apps. *IEEE Transactions on Software Engineering*, Vol. 41, No. 4, pp. 384-407. https://doi.org/10.1109/TSE.2014.2367027

Boehm B. (1986). A spiral model of software development and enhancement. *ACM SIGSOFT Software Engineering Notes*, pp. 14-24.

Boehm B. W. (1988). A spiral model of software development and enhancement computer. *Journal of Computer*, Vol. 21, No. 5, pp. 61-72. https://doi.org/10.1109/2.59

Brereton P., Budgen D., Hamilton G. (1998). Hypertext: the next maintenance mountain. *Computer*, Vol. 31, pp. 49-55. https://doi.org/10.1109/2.73585

Chaffey D. (2016). *Mobile Marketing Statistics.* Available from http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics

Clancy T. (2015). *The Standish Group Report.* Available from: https://www.projectsmart.co.uk/white-papers/chaos-report.pdf

Corral L. (2013). Software development process for mobile system. *Proceeding of 2013 1st International Workshop on the Engineering of Mobile Enabled System*, pp. 19-24. https://doi.org/10.1109/MOBS.2013.6614218

Cunha T. F., Dantas V., Andrade R. (2011). SLeSS: A scrum and lean six sigma integration approach for the development of software customization for mobile phones. *Proceeding of the 2011, 25th Brazilian Symposium on Software Engineering*, pp. 283-292. https://doi.org/10.1109/SBES.2011.38

Dantas V. L., Morinho F. G., Costa A. L., Andrade R. M. (2009). Testing requirements for mobile applications. *In the Proceeding of the 2009 24th International Symposium on Computer and Information Sciences*, pp. 555-560. https://doi.org/10.1109/ISCIS.2009.5291880

Ei E., Nwe N. (2016) Model-driven development of mobile applications. *International Conference on Software Engineering Research, Management and Applications*, pp. 179-185. https://doi.org/10.1007/s10270-016-0559-4

Eom H., Lee S. (2013). Human-cantered software development methodology in mobile computing environment: Agent-supported agile approach. *EURASIP Journal on Wireless Communications and Networking*. https://doi.org/10.1186/1687-1499-2013-111

Flora K. H., Wang X., Chande V. S. (2014). An investigation of the characteristics of mobile applications: A survey study. *International journal Information Technology and Computer Science*, Vol. 6, No. 11, pp. 21-27. https://doi.org/10.5815/ijitcs.2014.11.03

Flora D., Wang X. (2014) Adopting an agile approach for the development of mobile applications. *Journal of Computer Applications*, Vol. 94, No. 17, pp. 43-50. https://doi.org/10.5120/16454-6199

Flora H. K., Chande S. V. (2013). A Review and analysis on mobile application development processes using agile methodologies. *International Journal of Research in Computer Science*, Vol. 3, No. 4, pp. 8-18. https://doi.org/10.7815/ ijorcs.34.2013.068

Galeano I., Casariego J., Merln M. (2016). Mobile application development approaches a systematic mapping study. *In 2016 35th International Conference of the Chilean Computer Science Society*, pp. 1-12. https://doi.org/10.1109/SCCC.2016.7836006

Garg R., Thakur H., Tripathi B. (2017). Nonlinear numerical analysis of convective-radiative fin using MLPG method. *International Journal of Heat and Technology*, Vol. 35, No. 4, pp. 721-729. https://doi.org/10.18280/ijht.350405

Gasimov A., Tan C. (2010). Visiting mobile application development: What, how and where. *Proceeding of the 2010 Ninth International Conference on Mobile Business*, pp. 74-81. https://doi.org/10.1109/ICMB-GMR.2010.20

Genco A., Viggiano A., Magi V. (2018). How to enhance the energy efficiency of HVAC systems. *Mathematical Modelling of Engineering Problems*, Vol. 5, No. 3, pp. 153-160. https://doi.org/10.18280/mmep.050304

Habra N., Alexandre S., Desharnais J. M., Laporte Y. C., Renaults A. (2008). Initiating software process improvement in small enterprises. *Information and Software Technology*, Vol. 50, No. 7, pp. 763-771. https://doi.org/10.1016/j.infsof.2007.08.004

Hammershoj A., Sapuppo A., Tadayoni R. (2010). Challenges for mobile application development. *International Conference on Intelligence in Next Generation Networks*, pp. 1-8. https://doi.org/10.1109/ICIN.2010.5640893

Hart S. (2016). *Mobile VS. Desktop: 10 Key Differences.* Available from https://www.paradoxlabs.com/blog/mobile-vs-desktop-10-key-differences/

Heberto M. F. (2015). Best practices for mobile applications development. *Developments Carried Out by the Small and Medium Enterprises in Mexico*, pp. 1-6.

Herman M., Subaihin A., Jia Y., Martin W., Sarro F., Zhang Y. (2016). Mobile app and app store analysis, testing and optimization. *In proceeding of the International Conference on Mobile Software Engineering and Systems*, pp. 243-244. https://doi.org/10.1145/2897073.2897076

Hussain A., Ferneley E. (2008). Usability metric for mobile application: A goal question metric (GQM) approach. *In the Proceeding of the 2008, 10th International Conference on Information Integration and Web-based Applications & Services*, pp. 567-570. https://doi.org/10.1145/1497308.1497412

Iacob C., Harrison R. (2013). Retrieving and analyzing mobile apps feature requests from online reviews. *In Proceeding of 2013 10th Working Conference on Mining Software Repositories Mining Software Repositories*, pp. 41-44. https://doi.org/10.1109/MSR.2013.6624001

Ihara S. (1993). Information theory for continuous systems. *World Scientific*, Vol. 1, pp. 293-299. https://doi.org/10.1142/1676

Inukollu V. N., Keshamoni D. D., Kang T., Inukollu M. (2014). Factors influencing quality of mobile apps: Role of mobile app development life cycle. *International Journal of Software Engineering & Applications*, Vol. 5, No. 5, pp. 1-20. https://doi.org/10.5121/ijsea.2014.5502

Jawadekar W. (2004). Software engineering: principles and practice. *McGraw- Hill, New York, Computer Engineering Series.*

Jeong J. Y., Lee H. J., Shin S. G. (2008). Development process of mobile application SW based on agile methodology. *In 10th International Conference on Advanced Communication Technology*, Vol. 1, pp. 362-366. https://doi.org/10.1109/ICACT.2008.4493779

Jiang L., Eberlein A. (2008). Towards a framework for understanding the relationships between classical software engineering and agile methodologies. *Proceedings of the International Workshop on Scrutinizing Agile Practices or Shoot-out at the Agile Corral*, pp. 9-14. https://doi.org/10.1145/1370143.1370146

Joorabchi E. M., Mesbah A., Kruchten P. (2013). Real challenges in mobile app development. *In 2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement,* pp. 15-24. https://doi.org/10.1109/ESEM.2013.9

Kamthan P. (2008). Towards high-quality mobile applications by a systematic integration of patterns. *Journal of Mobile Multimedia,* Vol. 4, No. 3, pp. 165-184.

Kathuria R., Chowdhury S. R., Kedia M., Srivastav S. (2015). An inquiry into the impact of India'S app economy. *Indian Council for Research on International Economic Relations Master Thesis,* pp. 1-52.

Kim H., Oh S. J., Moon N. H. (2013). Development of smart mobile app assessment model. *In the Proceeding of the 2013 16th International Conference on Network-based Information Systems,* pp. 300-304. https://doi.org/10.1109/NBiS.2013.100

Kim Y., Park G. (2011). The 4-tier design pattern for the development of an Android application. *In the Proceeding of the 2016 International Conference on Data and Software Engineering,* pp. 196-203. https://doi.org/10.1007/978-3-642-27142-7_23

La H. J., Lee H. J., Kim S. D. (2011). An efficiency-centric design methodology for mobile application architectures. *Proceeding of the Wireless and Mobile,* pp. 272-279. https://doi.org/10.1109/WiMOB.2011.6085388

Li D., Guo B., Shen Y., Li J., Huang Y. (2016). The evolution of open source mobile applications: An empirical study. *Journal of Software: Evolution and Process,* Vol. 29, No. 7, pp. 1-18. https://doi.org/10.1002/smr.1855

Litoriya R., Kothari A. (2013). Cost estimation of web projects in context with agile paradigm: Improvements and validation. *International Journal of Software Engineering,* Vol. 6, No. 2, pp. 91-114.

Litoriya R., Kothari A. (2013). An efficient approach for agile web based project estimation: agile MOW. *Journal of Software Engineering and Applications (Scientific Research USA),* Vol. 6, No. 6, pp. 297-303.

Liu Z., Gao X., Long X. (2010). Adaptive random testing of mobile application. *In the Proceeding of the 2010, 2nd International Conference on Computer Engineering and Technology,* pp. 297-301. https://doi.org/10.1109/ICCET.2010.5485442

Main S. (2013). *Search Light.* available from https://code.google.com/p/search-light

Minelli R. (2012). Software analytics for mobile applications. *Master Thesis,* pp. 1-94.

Nagappan M., Shihab E. (2016). Future trend in software engineering research for mobile apps. *In 2016 IEEE 23rd International Conference on Software Analysis, Evolution and Reengineering,* pp. 21-32. https://doi.org/10.1109/SANER.2016.88

Nezerwa M., Chen T., Stahl F. J., Joshi N., Terranova J., Wright R., Carlson C., Citardi D., Dimattia M., Howansky S., Coppola F. J. (2015). Universal design with mobile app development: Bridging the gap for the forgotten populations. *Long Island Systems, Applications and Technology,* pp. 1-6. https://doi.org/10.1109/LISAT.2015.7160201

Nosseir A., Flood D., Harrison R., Ibrahim O. (2012). Mobile development process spiral. *In the Proceeding of the 2012 Seventh International Conference on Computer Engineering & Systems,* pp. 281-286. https://doi.org/10.1109/ICCES.2012.6408529

Nahak M. P., Triveni M. K., Panua R. (2017). Numerical investigation of mixed convection in a lid-driven triangular cavity with a circular cylinder using ANN modeling. *International*

Journal of Heat and Technology, Vol. 35, No. 4, pp. 903-918. https://doi.org/10.18280/ijht.35042

Okediran O. O., Arulogun O. T., Ganiyu R. A., Oyeleye C. A. (2014). Mobile operating system and application development platforms: Asurvey. *Int. J. Advanced Networking and Applications*, Vol. 6, No. 1, pp. 2195- 2201.

Perez S. (2013). *Users Have Low Tolerance for Buggy Apps.* Available from https://techcrunch.com/2013/03/12/users-have-low-tolerance-for-buggy-apps-only-16-will-try-a-failing-app-more-than-twice/

Perez S. (2016). *Worldwide App Store Downloads.* Available from https://www.statista.com/statistics/266488/forecast-of-mobile-app-downloads

Prasad G. P., Hamsini R., Smitha G. R. (2016). Agile development methodology and testing for mobile applications-a survey. *International Journal of New Technology and Research*, Vol. 2, No. 9, pp. 98-101.

Pressmen R. (2014). Software engineering: A practitioner's perspective. *7th Edition, McGraw-Hill*, pp. 769-798.

Putra E. P. (2012). Industrial extreme programming practice's implementation in rational unified process on agile development theme. *International Conference on Advanced Computer Science and Information Systems (ICACSIS '12). IEEE, Depok, Indonesia,* pp. 137-142.

Quamer A. Seelers B. (2006). Measuring agility and adoptability of agile methods: A 4 – dimensional analytical tool. *In proceeding of International Conference Applied Computing*, pp. 503-507.

Qumer A., Sellers B. H. (2008). An evaluation of the degree of agility in six agile methods and its applicability for method engineering. *Information and Software Technology*, Vol. 50, No. 4, pp. 280-295. https://doi.org/10.1016/j.infsof.2007.02.002

Rahimian V., Ramsin R. (2008). Designing an agile methodology for mobile software development: A hybrid method engineering approach. *In proceeding of the 2008, Second International Conference on Research Challenges in Information Science*, pp. 337-342. https://doi.org/10.1109/RCIS.2008.4632123

Rajput G. S., Litoriya R. (2014). Corad agile method for agile software cost estimation. *Open Access Library Journal (OALIB)*, Vol. 1, pp. 1-13.

Sa D. M., Carrigo L. (2008). Lessons from early stages design of mobile applications. *In: Proceeding of the 2008, 10th International Conference on Human Computer Interaction with Mobile Devices and Services*, pp. 127-136. https://doi.org/10.1145/1409240.1409255

Sambasivan D., John N., Udaykumar S., Gupta R. (2011). Generic framework for mobile application development. *In Second Asian Himalayas International Conference on Internet*, pp. 1-5. https://doi.org/10.1109/AHICI.2011.6113938

Scacchi W. (2011). Process models in software engineering. *Encyclopaedia of Software Engineering.*

Scharff C., Verma R. (2010). Scrum to support mobile application development projects in a just-in-time learning context. *In: Proceeding of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering*, pp. 25-31. https://doi.org/10.1145/1833310.1833315

Shannon C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, Vol. 27, No. 3, pp. 379-380. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

Singh G. M. (2015). Empirical study of agile software development methodologies: A comparative analysis. *ACM SIGSOFT Software Engineering Notes*, pp. 1-6. https://doi.org/10.1145/2693208.2693233

Solinski A., Petersen K. (2016). Prioritizing agile benefits and limitations in relation to practice usage. *Software Qual J.*, Vol. 24, No. 2, pp. 447-482. https://doi.org/10.1007/s1121

Song S. L. (2018). Application of gray prediction and linear programming model in economic management. *Mathematical Modelling of Engineering Problems*, Vol. 5, No. 1, pp. 46-50. https://doi.org/10.18280/mmep.050107

Stojanovic Z., Dahanayake A., Sol H. G. (2003). Modelling and architectural design in agile development methodologies. *In Proceedings of the 8th International Workshop on Evaluation Methods in System Analysis and Design*, pp. 180-189.

Summerville I. (2000). Software engineering. *6th Edition, Addison-Wesley*, pp. 50-55

Vallon R. (2015). An agile and lean process model for mobile app development: Case study into Austrian industry. *Journal of Software*, Vol. 10, No. 11, pp. 1245-1264. https://doi.org/10.17706/jsw.10.11.1245-1264

Voas J., Micheal B. J., Genuchten V. M. (2012). The mobile software app takeover. *IEEE Software*, Vol. 29, pp. 25-27. https://doi.org/10.1109/MS.2012.104

Vu P. M., Pham H. V., Nguyen T. T., Ngyun T. T. (2015). Tool support for analyzing mobile app reviews. *In Proceeding of 2013 30th IEEE.ACM International Conference on Automated Software Engineering*, pp. 789-794. https://doi.org/10.1109/ASE.2015.101

Wasserman A. I. (2010). Software engineering issues for mobile application development. *In the Proceeding of the FSE/SDP Workshop on Future of Software Engineering Research*, pp. 397-400. https://doi.org/10.1145/1882362.1882443

Zhang D. B., Adipat B. (2009). Challenges, methodologies, and issues in the usability testing of mobile applications. *Int. J. Human Comp. Interaction*, Vol. 18, No. 3, pp. 293-308. https://doi.org/10.1207/s15327590ijhc1803_3