

BiCycle Model: A New SDLC Model for an Individual Mobile Application Developer

P. KarthiK

*Asst Prof, PG Department of Software Technology
Sacred Heart College (Autonomous)
Tirupattur – 635 601. Vellore DT
pkksa@gmail.com*

Abstract— Software Development Lifecycle is crucial in Desktop or web application development. The mobile application is entirely different from desktop and web-based application. Mobile applications are being developed for deployment in smart phones such as ANDROID, WINDOWS and APPLE. Looking at the rising need of applications and the associated development complexity, it is imperative to have a dedicated lifecycle for the application development. Now-a-days quite numbers of mobile apps are being developed by individual developer in an unprofessional manner which results in poor quality software. BiCycle Model includes all the necessary activities for the individual developer to develop an app in a professional manner which will result in quality. Because the traditional SDLC Model and recent MSDLC Model don't address the solution for the problems faced by the individual developer in developing professional apps. Those are proposed only for the team based environment.

Keywords— *Bicycle, MSDLC.*

I. INTRODUCTION

The mobile applications market is currently undergoing rapid expansion, as mobile platforms continue to improve in performance, and as the users' need for a wide variety of mobile applications increases. The latest mobile platforms allow for extensive utilization of network resources, and thus offer a strong alternative to workstations and associated software[1].

Software development for mobile platforms comes with unique features and constraints that apply to most of the lifecycle stages[2]. The development environment and the technologies that support the software are different compared to "traditional" settings. Environment particularities include: a high level of competitiveness; necessarily short time to delivery; and added difficulty in identifying stakeholders and their requirements[3].

Developer has to face the challenge of a dynamic environment, with frequent modifications in customer needs and expectations. Technological constraints apply to mobile platforms in the form of limited physical resources and rapidly changing specifications[4]. The mobile application has limited screen, reduced data entry capability (due to a limited keypad for example), memory capacity, processing power and limited power reserve, are permanent, at least relative to desktop environments. Due to significant differences in the environment and in platform specifications, mobile application

development requires a suitable development methodology. Current mobile industries are predominantly occupied by the ANDROID phone, because of simple and efficiency of supporting all variety of people and need.

II. EXISTING MOBILE APP DEVELOPMENT LIFE CYCLE

Intel IT has created "A Mobile Application Development Framework". This framework defines specific activities, tools and resources, to support the planning, development and deployment of mobile applications. Intel has a two-step process defined for developing a mobile application.

- Use a pre-defined decision matrix; to verify whether the application is suitable for mobile development.
- Determine the process to be used in delivering the application.

A. Slalom Consulting

Have defined the mobile application development lifecycle as "Enterprise Mobile Application Lifecycle". This process is for an end-to-end mobile application development. They have defined the lifecycle in the following four different phases:

- Discovery.
- Design.
- Development/Testing.

B. A Mobile Application Development Lifecycle [7] (MADLC)

It has been proposed to bring out a formal lifecycle for mobile application development. MADLC will aid the mobile application developers in developing high-end apps. This lifecycle includes the following phases:

- Identification.
- Design.
- Development.
- Prototyping.
- Testing.
- Maintenance.

C. The “Mobile-D process” [8]

It should be 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.
- User-Centred Focus.

The Architecture Line in the methodology is a new addition to the already established agile practices. An architecture line is used to capture an organization’s knowledge of architectural

solutions, from both internal and external sources, and to use these solutions when needed. Mobile-D comprises five phases: Explore, Initialise, Productionise, Stabilise, and System Test & Fix. Deployment.

III. COMPARISON OF SDLC MODELS

PC Application Development and Mobile Application Development have the following differences.[9]

- Life Span.
- Complex Functionalities.
- Fewer physical interfaces.
- More Number of screens for Interaction.
- Battery & Memory usage by the application.
- Cross Platform Development.

TABLE I
MOBILE SDLC COMPARISON

Process Models	Strength	Weakness
A Mobile Application Development Framework[10] provided by Intel IT Work	<ul style="list-style-type: none"> • Efficient for team-based environment • It has two step process and within that many other sub process • It is cross-platform framework 	<ul style="list-style-type: none"> • This model is provided only for well-trained team-based environment. • Not suitable for individual person. • No unique features for android-based app.
A Mobile Application Development Lifecycle (MADLC)[12]	<ul style="list-style-type: none"> • It also proposed the team consists of at least 10 members. • Well-defined activities to develop a cross-platform mobile app. 	<ul style="list-style-type: none"> • This model is provided only for well-trained team-based environment. • To perform each activity, we need dedicated team. • Not suitable for individual person.
Mobile-D process	<ul style="list-style-type: none"> • It also proposed the team consists of at least 10 members. • Well-defined activities to develop a cross-platform mobile app. 	<ul style="list-style-type: none"> • This model is provided only a team of at most ten co-located developers. • To perform each activity, we need dedicated team. • Not suitable for individual person.

IV. STUDY ON BAD APPS

According to the top negative reviews and statistics, “44% verbally express they would rub out a mobile app immediately if app did not perform as expected”. The numbers clearly point out that there are good apps, and lamentable apps in the app market. App users not only uninstall the app, but withal provide negative reviews on the app when customers do not relish the app.

With social media and word of mouth being so popular negative reviews spread rapidly, which rigorously affects the reputation of developers and poses a threat to their future releases. Hence it is very critical for the developers to understand the criterion for good apps vs. bad apps and develop accordingly[6][9].

The percentage of low quality apps (22%) is a very consequential number and shows that there is an immediate need to fixate on the quality of apps. With several apps on board and with hundreds of thousands of options, the consumers are persuaded to buy highly polished apps, and so the market is no longer favorable to amateur apps.

A. What is Bad/Low Quality Apps?

“Bad” Refers - not conforming to Standards. An App can be considered to be Deplorable if it has the following Reasons

- A poor design/UI (Inexperienced developers).
- Has lot of clutter.
- Has poor navigation.
- Does not meet the user requirements.
- Does not address a specific issue.
- Have security issues.
- Fails at certain essential times.
- Has downloading issues.
- Is not consistent across various platforms.
- Have compatibility issues.
- Consumes lot of battery power.
- Has a very slow replication function.
- Has very high ad frequency.
- Is not appropriately priced.
- And has no endeavors made to fine-tune issues/ concerns raised by the users.
- Apps must be updated regularly to keep customers focused and engaged.

If an app is made and has never been looked at again, then the app could be counted as a bad app.

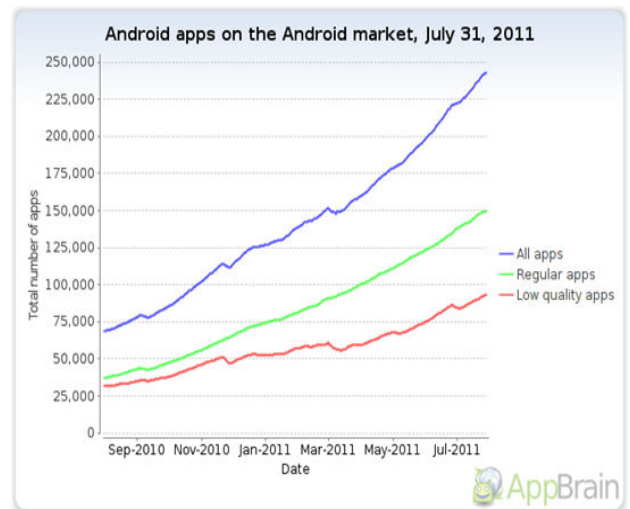


Fig. 1 App Status in 2011

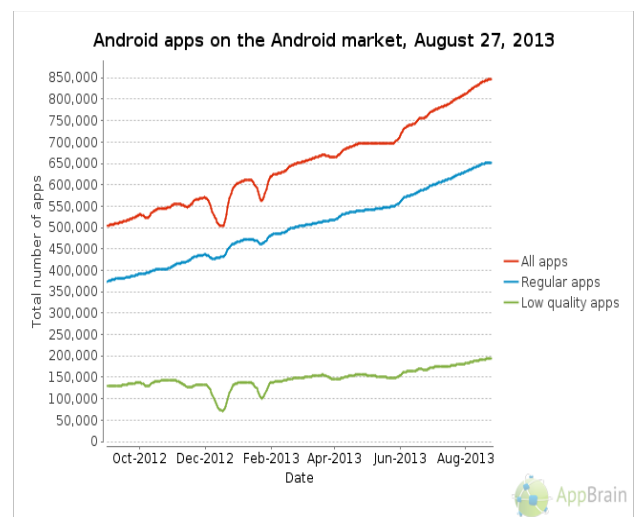


Fig. 2 App Status in 2013

V. SOURCE OF PROBLEM

Developer is the first and foremost one who is responsible for the failure of an app. There are various causes which include lack of expertise in terms of app development, minimal resources, minimum/no knowledge of user demands and expectations, no knowledge of target audience, and lack of communication between developers of an app. Users say there are frustrations that would lead them to give an app a bad review” including the following:[7]

- Application/system freezes–76%.
- Application/system crashes–71%.
- Slow responsiveness–59%.
- High battery consumption–55%.
- Considerable amount of ads and promotions–53%.

A. Problems faced by the Individual Developers

1. No systematic approach is followed.
2. Difficult to track progress.
3. Don't know where and how to start and stop.
4. Difficult to implement the desired functionalities.
5. Poor quality.
6. No in-depth testing.
7. Inability to reach application standard.

The traditional SDLC Model and recent MSDLC Model don't address the solution of the problems faced by the individual developer in developing professional mobile apps. Because the mobile application is entirely different from desktop and web-based application, it is impossible to follow the traditional SDLC Model and recent MSDLC Model. Because those are proposed for the team-based environment.

VI. SOLUTION TO THE BAD APPS

A. Suggestions to the Problems of the Individual Developer

- Developer should follow the process oriented approach while developing a mobile application. Though the process is sometimes time consuming, it is easy to refer back to the process and rectify the app if any errors are reported.
- Developer should accumulate input and information from the users through surveys.
- Developer should possess adequate training and enough practice before beginning to develop apps.
- Developer should use well known and certified tools during development of an app. Superior quality apps can be developed through effective tools.
- Developer should spend sufficient time on testing the apps with respect to security and performance criterion which will play a vital role in the success/failure of an app.
- Developer can use any software management tool to plan their mobile app development activities. Efficient planning will help the developer to meet strict deadlines.

B. Role of App Development Life Cycle

1) *The reasons for low quality apps from SDLC point of view:*

- The first and foremost reason is that the app developer is not conforming to the development life cycle phases.
- Most of the app developers start developing the app without accumulating requirements and without having a design.

- Lack of training and experience on the app development SDKs.
- Not enough testing is done. App developers are more fixated on functional aspects of the app and hence they sometimes ignore security and performance testing, which are the key components of any app.
- Poor maintenance.

Why do we need a software development life cycle and what happens if we do not use systematic approaches while developing the software product?[13]. The result is lower quality software products.

A mobile application is nothing but a software product with a different level of complexity. One can apply same conventional methods/methodologies along with different mobile app techniques and tools to design, develop, test and deploy a mobile application.

C. Proposed Mobile App Development Life Cycle Model

The "BiCycle Model for Mobile Application Development" is proposed to the problems faced by individual developer in developing professional and standard mobile apps.

It addresses all the necessary activities to be carried out by individual to develop apps in a systematic way. In this model, the process is divided into two different cycles.

VI. PROPOSED SDLC MODEL

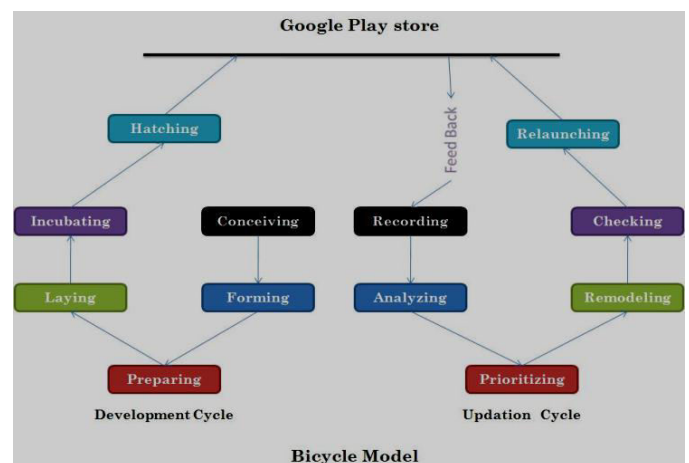


Fig. 3 BICYCLE Model

A. TWO CYCLES

1) Development Cycle:

- Conceiving (Idea Generation).
- Forming (Problem Definition).

- Preparing (Planning).
- Laying (Designing).
- Incubating (Developing & Testing).
- Hatching (Launching the App to Google Play Store).

2) Updation Cycle :

- Recording (Collecting and Documenting the Feedback).
- Analyzing.
- Prioritizing (Deciding).
- Remodeling (Redesigning and Reconstructing).
- Checking.
- Re-launching.

VII. DEVELOPMENT CYCLE

This is the main cycle of the BiCycle model. It starts when the idea or concept for new application arises. For the development of every new application this is the first cycle. During this development cycle, the idea or concept will be transformed into the professional android app. This will be over once for all when the app is launched. For the next iteration or updation, separate cycle will be followed based on the feedback and/or technological advances.

B. This cycle has following six activities from Conceiving to Hatching.

- Conceiving.
- Forming.
- Preparing.
- Laying.
- Incubating.
- Hatching.

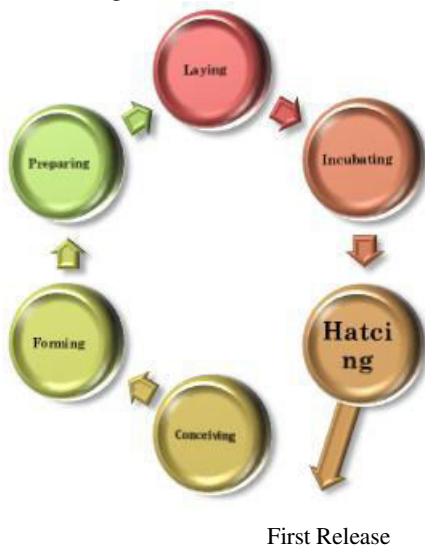


Fig. 4 Development Cycle

VIII. UPDATION CYCLE

This is the second cycle of the BiCycle model. It starts when the feedback or suggestion comes from the customer, Google play store and other sources. This cycle focuses on the next iteration of the app or updation of the feedback. It keeps on continuing until the app is really out of the industry.

A. The Updation Cycle has the following actions

- Recording.
- Analysing.
- Prioritizing.
- Remodelling.
- Checking.
- Re-launching.



Updated Version

Fig. 5 Updation Cycle

IX. RESEARCH REFLECTIONS

The primary motive of this research is to show that the methodological aspect of software engineering can be successfully applied to mobile applications, with the help of the software development life cycle.

A. Following are the Observations of this Research

- Mobile app is not a tiny/simple piece of software anymore.
- Apps are continually growing in number and complexity.
- Lots of android apps are being developed by individual, the numbers are still growing.

- Process oriented approaches and techniques are required to handle mobile application development.
- Superior quality mobile applications can be developed with the help of process-oriented methods.
- Process oriented can also be applied by the individual developer in developing high quality mobile apps.
- Unique SDLC model for the individual android developer will assist them in developing high quality apps.

X. OUTCOME OF THE IMPLEMENTATION

The outcome of the development process is that this model helped the developers to plan and execute the application involving various features in a more effective manner. It produced the following outputs

- It directed the developer where to start and stop his task.
- It provided with list of activities and predefined documents.
- Easily track the progress of the development.
- Developer knows where he/she is now and also knows what to do next.
- They don't want to discuss about upcoming tasks. Because it provides all the necessary activities.
- Resulted in high quality software with customer satisfaction.
- Able to fulfill all the requirements.

REFERENCES

- [1] Mobile application history of information available URL is "http://en.wikipedia.org/wiki/Mobile_app".
- [2] Mobile Stats available URL is, "www.slideshare.net/vaibhavkubadia75/mobile-web-vs-mobile-apps-27540693?from_search=1".
- [3] Mobile Website vs Apps, <http://www.hswsolutions.com/services/mobile-web-development/mobilewebsite-vs-apps/>
- [4] Users Reveal Top Frustrations that Lead to Bad Mobile App Reviews, statistics report, https://blog.apigee.com/detail/users_reveal_top_frustrations_that_lead_to_bad_mobile_app_reviews_infographic
- [5] Android applications stats report, <http://www.appbrain.com/stats/number-of-android-apps>
- [6] When mobile apps go bad, <http://www.infoworld.com/d/mobile-technology/when-mobile-apps-gobad-178063>
- [7] Make a good app to great app <http://gigaom.com/2008/03/26/what-makes-a-good-mobile-applicationgreat/>
- [8] Pekka Abrahamsson, "Mobile-D: an agile approach for mobile application development' Approach" Conference on Object Oriented Programming Systems Languages and Applications, pp 174 - 175, 2004
- [9] Agboma, Florence, and Antonio Liotta. "Addressing user expectations in mobile content delivery." *Mobile Information Systems* 3, no. 3 (2007): 153-164.
- [10] IT Intel White Paper, "Building a Mobile Application Development Framework", August 2012.
- [11] Daniel Mavcock, Slalom Consulting, "Enterprise Mobile Application Lifecycle."
- [12] Make a good app to great app <http://gigaom.com/2008/03/26/what-makes-a-good-mobile-applicationgreat/>
- [13] Factors influencing quality of mobile Apps: role of mobile app development Life cycle Venkata N Inukollu, Divya D Keshamoni, Taeghyun Kang and Manikanta