Review Paper on Suitability of Traditional Prototype Model and Spiral Model used for Mobile Application Development Life Cycle

[1]Akansha Khandelwal, [2]Garima Tyagi
[1,2]Department of Computer Application,
School Of Computer Sciences,
Career Point University, Kota

Abstract- **Software development for desktop applications and Web based applications is based on Software development life cycle (SDLC) that plays a vital role in the development of mobile applications. With the invent of mobile, mobile application development is rising tremendously. The development of mobile applications is somehow complex in comparison to other applications as it should include some other features those are not the part of desktop and web applications like Graphical and Physical interface, performance etc. In this paper we are going to take different models for the development of mobile applications cannot be included in the basic phases of SDLC but also focusing on the other factors including cross platform development, memory usage, way of interaction with user and the model is proposed for the Mobile Application Development Life Cycle (MADLC).**

Keywords-MADLC, Mobile Application Development, SDLC, Mobile Applications

I. INTRODUCTION

With the vast development in the technical market, the enhancement in the mobile world has increased which had lead the software companies to move towards the mobile devices and shift from the traditional way of using computers. Desktop applications are now replaced from the mobile applications which have lead to the advanced increase in the demand of these applications although suitable with the hardware demand of the mobile device. By inspecting the use and demand of the mobile devices it is very much clear that by the development with time, the traditional computer system will be almost replaced by the mobiles and tablets. With the analysis about the increasing demand of the mobile and so the usage of mobile applications has put away to the developers to think and make more and more advanced applications those are suitable for mobile. Today apart from being an normal electronic gadget the mobiles have now emerged as a necessity for the human beings and that seems as a vital resource for a normal person.

II. LITRATURE SURVEY

A. Current software development process model

A software life cycle is a way to describe the process in which design can be developed for any type of structure for development of software. The software development life cycle (SDLC) provides the steps for specifying that how the initialization of the software development can be taken place. The steps included can be explained within the different models like waterfall model, incremental model, prototype model and spiral model with the variation of steps in every model. The spiral model focuses on the risk regarding the development of model whereas prototype model focuses on customer requirement. There is no major difference between the different types of models, the steps included are the same apart from the details within it: requirement gathering, planning, designing, construction and at last deployment.

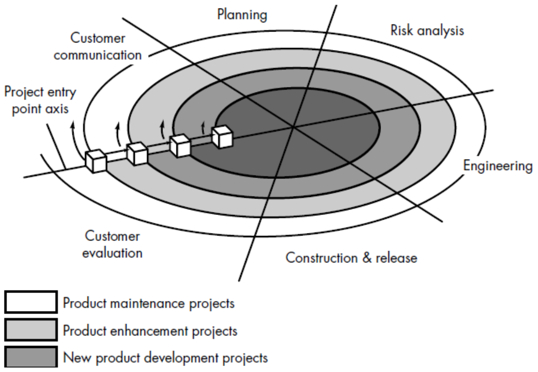
A.1. Spiral model

The *spiral model*, originally proposed by Boehm [BOE88], is an evolutionary software process model that includes the iterative n of prototyping with the controlled and systematic aspects of the traditional linear sequential model. In such a manner it can provides the potential for the rapid development of incremental versions of the software. With the help of spiral model, software is developed in a sequence of incremental releases. During the early phases of iterations, the incremental release might be a paper model or prototype model. During later phases of iterations, increasingly more complete versions of the new system are produced. The spiral model is a risk-driven generator for software projects. Based on the exclusive risk patterns of a specified project, with the help of spiral model one can guides a team to adopt elements of one or more process models.

Boehm lists six characteristics common to all applications of the spiral model. A spiral model is divided into a number of framework activities, also called *task regions* typically, there are between three and six task regions. Figure 2.8 depicts a spiral model that contains six task regions:

- **Customer communication**—tasks required to establish effective communication between developer and customer.
- **Planning**—tasks required to define resources, timelines, and other project related information.
- **Risk analysis**—tasks required to assess both technical and management risks.
- **Engineering**—tasks required to build one or more representations of the application.

• **Construction and release**—tasks required to construct, test, install, and provide user support (e.g., documentation and training).



A.2. Prototype model

The basic idea about the prototype model is that instead of finalising the requirements before a design or coding can proceed, a throwaway prototype is to be built to understand the requirements. This prototype is actually developed on the basis of the currently known requirements. By using this prototype, the client will be able to find that what will be the actual system, since the interactions with prototype enables the client to better understand the requirements of the desired system. The prototyping paradigm starts with the requirements gathering regarding the system. Developer and customer meet and define the overall objectives of the software, identify whatever requirements are known, and outline areas where further definition is mandatory. A "quick design" can be developed. The quick design focuses on a representation of those aspects of the software that will be visible to the customer/user (e.g.,input approaches and output formats). The quick design leads to the construction of a prototype. The prototype is evaluated by the customer/user and used to refine requirements for the software to be developed. Iteration occurs as the prototype is tuned to satisfy the needs of the customer, while at the same time enabling the developer to better understand what needs to be done. The prototypes are usually not complete systems and many of the details are not built in the prototype.

Advantages of Prototype model:

- Users are actively involved in the development
- Since in this methodology a working model of the system is provided, the users get a better understanding of the system being developed.
- Errors can be detected much earlier.
- Quicker user feedback is available leading to better solutions.
- Missing functionality can be identified easily
- Confusing or difficult functions can be identified Requirements validation, Quick implementation of, incomplete,but functional, application.

Disadvantages of Prototype model:

- Leads to implementing and then repairing way of building systems.
- Practically, this methodology may increase the complexity of the system as scope of the system may expand beyond original plans.
- Incomplete application may cause application not to be used as the full system was designed Incomplete or inadequate problem analysis.

When to use Prototype model:

- Prototype model should be used when the desired system needs to have a lot of interaction with the end users.
- Typically, online systems, web interfaces have a very high amount of interaction with end users, are best suited for Prototype model. It might take a while for a system to be built that allows ease of use and needs minimal training for the end user.
- Prototyping ensures that the end users constantly work with the system and provide a feedback which is incorporated in the prototype to result in a useable system. They are excellent for designing good human computer interface systems.

B. Mobile Application Development Process

The mobile application is developed by the framework created by INTEL. It is a two step process which defines already defined decision which is according to mobile application or not. In the second step the process is defined used for the delivery of the application.

The slalom consulting has defined the mobile application development process as the “Enterprise mobile application lifecycle” The phases defined within this are totally different from that of normal phases use within for mobile development. It uses four phases for defining any application- DISCOVERY, DESIGN, DEVELOPEMENT/TESTING, DEPLOYEMENT. The mobile application development has received a boost after the invention of 2G and 3G mobile network service; due to which desktop based applications were transferred to web based applications and developers got an opportunity and a boost to make their skills use and make best and advance web application.

The desktop and mobile applications have large difference between two-

- Language independency- the use of many different languages like java, c++ and many more have given the developers to develop an application within their level of language comfort and independence. But with this context the life span of desktop is large as compared with that of mobile application.
- Functional dispatch – The mobile application involve many complex functionality with interference with the telephone system, GPS, message handling etc. While considering desktop application, the application developed is restricted o the desktop or laptop level only strictly.

- Physical interface-the physical interfaces like keyboard, mouse etc are attached separately with the desktop while in the mobile only the touch screen or with touch keyboard is inbuilt within it which is very user friendly and can be carried within everywhere without any mess due to small size.
- Memory and power consumption- while developing any application the major thing to be kept in mind s the applications memory and battery. The mobile application is used all day with many applications running simultaneously within. While in the desktop only the running application consume the battery and memory whereas when no application run i.e, when idle processes are then no battery is consumed by the laptop or desktop.
- Different operating systems- the operating system used by mobile applications are cross platform dependent. The soul factor in mobile application is its platform, i.e. the Android platform is user-friendly as well as developer friendly. Whereas other mobile platforms like iOS, Windows, Blackberry are restrict and closed developers access. This limitation does not exist on desktop application development.
- Updating facility- mobile application is served to the user in format that is to be downloaded which provides the user registration automatically whereas desktop application is served within a CD which does not provide the facility of user registration and thus no updating is done within it.

III MOBILE APPLICATION DEVELOPMENT LIFE CYCLE (MADLC)

Here we are implementing the MADLC with the combination of Prototype model and Spiral Model that will be having the following phases of the development life cycle.

1. **Customer communication**- this phase includes the requirement gathering by communicating with the customer. This includes the evaluation and collection of new ideas out of the customers mind and is further analysis by the developer himself. The list of ideas is filtered by the developer and is moved on further to the mobile application team to develop the application.
2. **Planning**- in this phase the information gathered by the customer is proceeded on by making the strategy of how to implement the ideas and the requirement of the customer into the application in a perfect way under the time interval decided by the developers team.

3. **Construction of the Prototype**-With the planning being finalised by the developers, a prototype is constructed i.e., a rough model is constructed for the customer to let him made aware that the model will look exactly the same after being finally constructed. This gives the customer a feeling of relief in his mind that his requirement is being evaluated in a correct manner.

4. **Evaluation of prototype until confirmation**-The prototype is evaluated by the customer and used to refine requirements for the software to be developed. Iteration occurs as the prototype is tuned to satisfy the needs of the customer, while at the same time enabling the developer to better understand what needs to be done. Ideally, the prototype serves as a mechanism for identifying software requirements. If a working prototype is built, the developer attempts to use existing program fragments or applies tools (e.g., report generators, window managers) that enable working programs to be generated quickly.

5. **Engineering**-It includes the task required to build one or more representations of the application. It gives the developer and the customer the illustration of the application being built which is suitable in one or two manner.

6. **Risk analyses**- It includes the tasks required to assess both technical and management risks.

7. **Construction**- With this phase the actual starting of the application is made. The coding of the prototype constructed is done and made final after being evaluated by the developer team.

8. **Testing**-It is the most important phase of mobile application development life cycle. In this phase the prototypes build of the application are tested at various levels step by step with the use of stimulator to test prototypes on multiple devices.

9. **Release**- This the final phase of the life cycle .the application after being tested and evaluated by the customer is made ready to be deployed. The application is launched in the market for the users to use.

IV RESULTS AFTER APPLYING THE MODEL

An application was developed using the above MADLC model. This application includes use of telephony services like sending SMS, configuring device profile and other general application parts. The development of this application development provided a test case for the proposed MADLC. All the functional requirements of the application were divided into various modules and delivered in a prototype fashion to the organization at various intervals. 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.

V. CONCLUSION

A Mobile Application Development Lifecycle (MADLC) has been proposed to bring out a formal lifecycle for mobile application development. MADLC will help the mobile application developers in developing mobile applications in a easier and flexible manner. This lifecycle includes the following phases: Customer communication, Planning, Construction of the Prototype, and Evaluation of prototype un till confirmation, Engineering, Risk analysis, Construction, Testing, Release, Maintenance. The different tasks and activities in the various phases of MADLC have been discussed. This lifecycle also addresses some of the distinguishing characteristics of mobile applications like complex functionalities, fewer physical interfaces, screen interfaces, memory usage, cross platform development and maintenance. Preliminary testing of the MADLC indicates that this lifecycle will help developers and project managers in efficiently execute projects and deliver solutions on time.

REFERENCES

[1] Tejas Vithani, Member, IAENG and Anand Kumar, “Modeling the Mobile Application Development Lifecycle”, International MultiConference of Engineers and Computer Scientists 2014 Vol I, IMECS 2014, March 12 - 14, 2014, Hong Kong

[2] Daniel Maycock, “Enterprise Mobile Application Lifecycle Developing a Process for End to End Mobile Application Development”, Slalom Consultant

[3] “Mobile Application Development Lifecycle: A Bird’s Eye View from Discovery to Launch**”,** Jul 1, 2014