Propose New SDLC Practices Model for Mobile Native Application

Dina Samy, Prof. Abdelfatah Hegazy, Prof. Mona kadry
{dinahsamy@gmail.com, ahegazy@aast.edu, monakadry@hotmail.com}

Faculty of Computer and Information Science, Arab Academy for Science and Technology and Maritime Transports, Cairo, Egypt

Abstract. The mobile application development is increasing rapidly due to the popularity of smartphones among end-users. The need to present the business as a mobile application has become essential to guarantee the value of the presented business services. The currently used software development methods for mobile application development could be appropriate, but it is not facilitating the complexity of the mobile application development and how it is built. Although some of the existing software development lifecycle models are adapted for mobile application development, but there are certain challenges that need to be addressed in each phase of the development lifecycle despite of the used software development method, such as; performance, user interface, and testing. The objective of this paper is to compare the different software development methods that were applied for mobile application development, and shows the required practices that should be applied specifically in each phase of the software development lifecycle to address the mobile native application challenges. Accordingly a qualitative study was conducted, in which 18 mobile native application professionals working in different companies were interviewed to collect the information about the current applied practices in their jobs and decide on the most effective practices that help them to facilitate the mobile application development challenges.

Keywords: Software engineering, Mobile application, Mobile native development, Software development lifecycle, Business application.

1 Introduction

Mobile software development is the process of developing software which can be used by small, low-power handheld devices such as mobile phones. This software is either factory pre-installed applications on mobile phones or downloadable from application stores and mobile software distribution platforms.

1.1 Software Development Models

The software development models are the various processes or methodologies that are being selected for the development of the project depending on the project's aims and objectives. There are many development life cycle models that have been developed in order to achieve different required objectives.

Software engineering book by Sommerville [1] stated the following question; "What are the best software engineering techniques?", While all software projects have to be professionally managed and developed, different techniques are appropriate for different types of systems. For example; games should always be developed using a series of prototypes whereas safety critical control systems require a complete and analyzable specification to be developed. Accordingly, it can't be claimed that one method is better than another. There are many different software processes, but all must include four activities that are fundamental to software engineering; Software specification, Software design and implementation, Software validation and Software evolution.

1.2 Mobile Application Software Development

Although there is no much difference between building application for desktops, Web or for mobile devices, the basic development lifecycle phases are always the same but the details are different. Accordingly, it is not possible to simply transfer the models of the traditional software development to mobile application development without making modifications to match the nature of the mobile application.

In addition, the mobile application is developed faster and thus have a faster lifecycle than traditional application and so the lifecycle supervision of mobile application development should be changed that way. The differences in the development of application in desktop/laptop and in mobile phones with respect to Hardware, Operating System (OS), Nature and size of the application, Operational environments, and Display functionalities [2].

2 Mobile Application Development Challenges

The mobile application development has specific practices different from the desktop, or web application development, this part includes a survey result that was conducted to gain an understanding of the main challenges developers face in practices when they build an application for different mobile devices. The outcome is an overview of the current challenges faced by mobile developers and their causes [3].

Table 1. The survey findings of the current practices and challenges, and highlight areas that require more attention from the research and development community.

Challenge	Causes
Moving toward Fragmentation rather than Unification	• Different platform (marketplaces, languages, tools, design guidelines) • Different versions of the same OS • Different smartphone devices supporting the same OS with different capabilities • No reusable code across different platforms (against quality)
Monitoring, Analysis and Testing Support.	• Limited tools • Automated testing support is currently very limited for native mobile Application • Current tools and emulators do not support important features for mobile testing such as mobility, location services, sensors, or different gestures and inputs

Open/Closed Development Platforms Data Intensive Application.	• iOS & windows are not open source vs. Android • The manufacturer customization of open source • So much data cannot be stored on the device • Using a network connection to sync up with another data source in the backend is challenging
Behavioral Consistency versus Specific HCI (Human Computer Interaction) Guidelines	• Each platform follows a set of specific HCI guidelines to provide a consistent look-and-feel across applications on the same device. • Developers would like their application to behave similarly across platforms, e.g., User interaction with a certain feature on Blackberry should be the same as on the iPhone and Android.
Time, Effort, and Budget are Multiplied	• Developers have to redesign and re-implement most of the application due to lack of support for automated migration across platforms
Testing challenges	• Manual Testing is Prevalent • Test the App for Each Platform Separately • Beta Testers and Third Party Testing Services • A unit test is preferred for small projects or critical code & not applicable for each platform • Limited, Unit Testing Support for Mobile Specific Features • Better support is needed to mimic real environments (e.g., Network latency, sensors) for testing • Rooted simulators and emulators are needed in order to access features outside of the application, such as settings, play store, Bluetooth and GPS, which could be part of a test case. • Performance of emulators is a key factor mentioned by many of our participants (android emulators are slow compared to iOS ones) • Mobile app projects change rapidly and very often over time, so there are some testing tools, but cannot be used because of the test updates for the dynamic Application. • It is difficult to identify all the usage scenarios and possible use cases, while there are a lot of hidden states; for example, enabling/disabling of the location services, and weak and strong network for network connectivity. • The relation between applications should be well managed, you might be interrupting other Application, or they might be interrupting yours.
Monitoring and Analysis	• Better analysis and monitoring support is needed to monitor, measure, and visualize various metrics of their Application such as: – Memory management (to spot memory leaks) – Battery usage (to optimize battery life) – CPU usage, pulling/pushing data, and network performance over various networks
Handling Crashes	• Application crashes are often intermittent, non-deterministic, and irrecoverable

- It's helpful to have a set of tools that would enable capturing state data as a crash occurs and creating a bug report automatically.

3 Software Process Models for Mobile Application Development

For developing a mobile application, traditional software development methods are applied, ignoring special characteristics of the mobile devices such as memory capacity, processing power, OS user experience, interface design, connectivity factor, bandwidth factor, lower battery factor, the input interface factor, which are different as compared to desktop applications. Accordingly, it is required to have a distinct mobile application development lifecycle model [5].

3.1 Comparison of Various Process Models adopted in Mobile Application Development

There are many software development lifecycles models adapted to mobile application development lifecycle, the below comparison shows the appropriateness of existing process models adapted to mobile application process models with respect to mobile application development which has been assessed on some specific characteristics [2,5].

Table 2. The comparison between various processes models used in mobile application development.

Process Model Applicable Characteristics of Mobile Applications	Spiral model Proposed by Ann Nosseir et al (2012)	Iterative model Proposed by Kemper and Wolf (2005)	Agile models (Many)	MADLC Proposed by TejasVithaniandAnand Kumar (2014)	Model-Driven Proposed by Fernandez and Hussmann (2008)
Environment	Stable	High volatile environment	High volatile environment	High volatile environment	High volatile environment
Focus	Risks involved	The main focus is on producing a new version of the app at the end of iteration to satisfy customer needs.	Human aspects of software Engineering	The main focus is on dividing functional requirements are into various modules and they are delivered As prototype at different Interims.	User-centered design
Team size	Large	Medium	Small team	Small	Small
Reliability	Less	High	Less	N/A	Less

Application Size	Large	Large	Small	Small	Small
Time to market	Long	Short	Short	Short	Short
Multiple Platform	N/A	N/A	N/A	Yes	Reduction in rewriting code again and again, Easy to understand and Non-experts can easily create specialized mobile applications
Suitability	Large, expensive, and complicated projects	Complex and dynamic Applications	For small organizations, developers and non-sequential projects	For Applications which have similar idea and are already existing in the market.	Non-experts can easily create specialized mobile applications.
Architecture	Designed for current and foreseeable requirements	Designed when requirements of the complete system are clearly defined and understood.	Designed for current requirements	Designed for users requirements and users himself comes out with an idea of how to develop, the idea is further detailed and analyzed.	Designed for more focus on the design and logic of the application
Refactoring	Expensive	Inexpensive	Inexpensive	Inexpensive	Inexpensive
Users Involvement	Throughout the life cycle	At the end of every iteration	Constant feedback from the user	Constant feedback from the user	Not much
Documentation	Heavy	High	Low	High	Low

The result from the above comparison showed that there are many process models for mobile application development. The literature review concentrated on specific characteristics of the Mobile application development which were not stressed upon and were not examined in the existing process models which are adapted in Mobile application development.

4 Mobile Application Development Practices

In order to overcome the mobile development challenges and to adopt the different nature of the mobile application development, there was a need to highlight and discuss the mobile application development practices that are currently used by the mobile application development professional regardless the followed software development model.

4.1 Study Design

Qualitative approach was applied to collect relevant data about the mobile native application development practices.

In-depth Interviews. include both individual interviews (e.g., One-to-one) as well as "group" interviews (focus groups). The data are recorded in a different ways, including audio recording and written notes. The interviews were based on set of previously prepared questions related to each phase in the application development life cycle.

Projective Techniques. was applied as an indirect form of questioning that encourages interviewees to project their underlying motivations, beliefs, attitudes or feelings regarding the issue of concern.

Table 3. The interviewees' information, coded in P1, P2, etc.

ID	Role	Mobile / SW experience	Team size	Company platform support
P1	Senior iOS developer	8	8	Android and iOS
P2	iOS developer	3	7	Android and iOS
P3	Android developer	2	7	Android and iOS
P4	Senior Android developer	4	7	Android and iOS
P5	Android and iOS team leader	8	8	Android and iOS
P6	Senior Android developer	8	12	Android and iOS
P7	Senior Android developer	5	5	Android and iOS
P8	Senior quality control	9	5	Android and iOS
P9	Senior quality control	3	5	Android and iOS
P10	Senior quality control	8	6	Android and iOS
P11	Quality control lead	9	5	Android and iOS
P12	Quality control lead	8	9	Android and iOS
P13	System architect	11	7	Android and iOS
P14	System designer	9	7	Android and iOS
P15	UX – UI designer	8	7	Android and iOS
P16	UX – UI designer	6	5	Android and iOS

4.2 The Interviews' Discussions

The interview questions were designed to focus on how to facilitate the mobile application development challenges with respect to the common development lifecycle phases; which are requirements gathering, design, implementation, and testing. During the interviews, the discussions were conducted based on the interviewee knowledge and years of experience, some

interviewees were able to discuss and answer all development phases' questions, and some interviewees were only able to address the phase which they were working on.

Table 4. The interviews questions with respect to the basic development lifecycle phaes.

Phase	Questions
Requirements gathering	• How do you define the application supported operating systems' versions (iOS, Android) and devices types? • What are the utilized smartphone features within the application business' services? • What is the impact of the offline and online service operating modes? • What do you specify the UX/UI requirements? • What are the mobile application security requirements?
Design	• How do you define the application user experience and design the application wireframes? • How do you manage offline and online operating modes and data caching? • How do you apply the application security requirements and avoid the mobile application security threats?
Implementation	• What are the best practices for iOS development? • What are the best practices for Android development? • How do the mobile application developers participate in the application testing?
Testing	• What are the mobile application testing methods and techniques? • What are the mobile application automated testing tools?

4.3 The Interviews' Answers Analysis

A Content analysis is conducted to categorize the interviews' answers for purposes of classification, summarization and tabulation of the recommended and the most effective practices in each development phase.

Table 5. The recommended practices with respect to the basic development lifecycle phase.

Phase	Practices
Requirements gathering	• Define the application, OS versions, referring to the App store and play store, and based on the business need • Define the application device types based on the business need and the customer preferences • Define the application layout specification based on the business need and the customer preferences • Give the mobility edge to the application by adopting smartphone features within the application business features. • Define the features data processing offline/ online mode based on the app performance and the business need • Define the application security requirements by considering the mobile native app security threats

Design	• Define the application user experience considering the OS nature and preserving the application consistency • Design the application interface based on the business need and following the Apple developer and Android developer tips for interface design and user experience • Adopting the native UX APIs to maximize the future computability and preserve the application quality and performance • Design the database in different processing offline/ online mode based on the app performance and the business need • Consider the application security by covering the mobile native app security threats and the business need
Implementation	• Using the default APIs as much as you can means that they will likely be updated while preserving backwards compatibility. • Abandoning deprecated APIs as soon as they get deprecated because they are scheduled to be removed from the OS soon. So you should stop using them as fast as possible. • Using a library/framework manager instead of manually adding them to keep the packages up to date in case of critical bugs or fixes • Following the general guidelines guarantees best future compatibility • Apply the unit test for the business functionality • Monitor the application battery consumption and memory usage
Testing	• Using the automated mobile testing tools to minimize the application testing effort and cost and due to the tools following advantages: – Support simulated and real-world testing to best understand usability, design and defects across any device, OS or network – Record the failure scenario automatically – Track the reported defects to the code level – Save the bug reporting time and effort – Include the defects reports and fixes operations – Can be used in the production phase to track the reported issues by customers – Gain insight into how end customers are using the application to get actionable data to improve development – Measure and simulate the impact of load using a combination of real devices and virtual users, for a realistic assessment – Determine response times, speed and quality of an application across real world devices – Accurately capture and share critical defect information of a device

5 Mobile Application Development Lifecycle Practices Model

The mentioned research result can be represented in a new model that shows the mobile application development special practices in each common phase of the software development life. Each Mobile development model or method (Spiral, Iterative, Agile, etc.) Could apply the same practices model with respect to how these phases are combined together in each model or method.

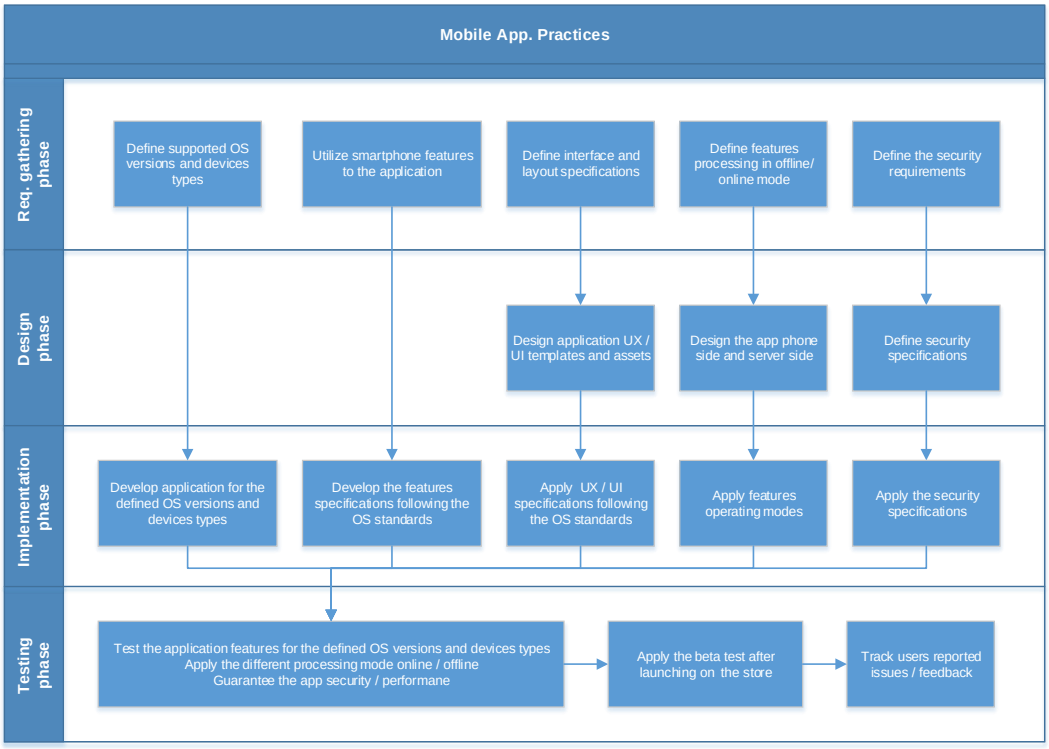

Fig. 1. The mobile application development practice and how they are related across the application development lifecycle phases.

6 Case study

The research results were applied in a mobile development team who are working on e-Banking mobile native application. The application was developed using iOS and Android technologies. The application was planned to be delivered in releases following Agile software development method.

Description of the problem. during the application development, the development team faced some issues; the application Android version development and testing was done in single Android OS version, many security issues were reported due to not following Android

development standards and guidelines, also handling the offline / online operation modes were not considered in iOS and Android versions.

The proposal. Apply the recommended practices in the application analysis, design, development and testing phases. The mobile app development team agreed how to apply the recommended changes within the current application release and commit to this in future releases. The system analyst and the system designer helped the team to define the required specifications in each phase.

Results and conclusion– by applying the recommended practices within the application analysis, development, and testing. The produced application was enhanced and the application was satisfying to the stakeholders' needs.

7 Conclusion

The mobile application development life cycle phases have specific practices, which are not matched with the desktop and the web application development. The research result was targeting to collect those practices, and present the value of applying those practices.

The mobile application development practices are a set of empirically proved approaches to the mobile Application development, when used in combination they strike at the root causes of mobile Application development challenges.

8 Future work

The researchers can be working on a comparison between the mobile development practice for iOS versus Android and record the advantages and disadvantages of both operating systems practices to improve the mobile application development activities.

Moreover, working on the mobile application development life cycle automation tools will help in preserving the application development standard practices and guide the application development team on how to facilitate the development process.

References

[1] Sommerville, Software engineering. Boston: Pearson, 2011

[2] Anureet Kaur and Kulwant Kaur (2015), "Suitability of existing Software development Life Cycle (SDLC) in context of Mobile Application Development Life Cycle (MADLC)", International Journal of Computer Applications (0975 – 8887) Volume 116 – No. 19, April 2015

[3] Mona Erfani, Ali Mesbah and Philippe Kruchten, "Real Challenges in Mobile App Development", Empirical Software Engineering and Measurement, 2013 ACM/IEEE International Symposium on, PP 15–24, 2013

[4] Mario Korf and Eugene Oksman, Native, HTML5, or Hybrid: Understanding Your Mobile Application Development options

[5] TejasVithani and Anand Kumar(2014) ,"Modeling the Mobile Application Development Lifecycle", Proceedings of the International MultiConference of Engineers and Computer Scientists 2014 Vol I,IMECS 2014, March 12 - 14, Hong Kong