

# Review and comparing the lifecycle models of mobile apps development

Oleksandr Lypovetskiy

Slovenská technická univerzita v Bratislave  
Fakulta informatiky a informačných technológií  
xlypovetskiy@stuba.sk

30. september 2021

## Abstrakt

This article will focus and deal with the consideration and comparison of different life cycle models for modeling and development of mobile applications. It will show their basic concepts and characteristics, give schematic pictures of these models, give some advantages and disadvantages of their use. We also find out which models should be used appropriately in the development of mobile applications, which steps of different life cycles are more suitable for the purposes of mobile software development and which are less or not suitable for it. Article will point the key features and differences in the process of planning, developing and testing mobile applications compared to other types of software.

## 1 Introduction

The second half of the 10s of the XXI century and the beginning of the 20s were marked by great technological progress in the field of mobile devices and software for their use. At least every month we can monitor the introduction of new technologies in, as now, our usual smartphones. These can be 5G technologies, artificial intelligence tools, augmented reality tools, physiological methods of information protection (fingerprint, face, retina scanning, etc.) and, of course, increasing the number of microprocessor transistors and, consequently, the number of its cores.

In recent years, due to portability and ease of use, maximum population has shifted to mobile devices. This remarkable growth of mobile devices has taken over desktop computers and is becoming a very important part of our life. The applications developed for mobile devices are becoming more and more advanced and complex, adjusting to the constantly improving computational power of hardware. The trend for mobile application development has spread so vastly that today everyone is using applications for everything from health to sports, from education to games; every imaginable domain in layman's life is making use of mobile apps. Smart devices have reached everyone's hands; yesterday it seemed like a luxury, today it is a necessity. The mobile internet usage has brought forward the trend of connectivity on-the-go. The applications are becoming more useful because the user gets to use them anywhere and

everywhere.

A Software development lifecycle is abstract representation of various processes used in the development of any software. There exist different lifecycle models. Some of them are: waterfall model, spiral model, agile model and prototyping model. Although there is not much difference between developing applications for desktops, Web or for mobile devices, the basic steps are always the same: requirements gathering, designing, implementing, testing, and delivery; but the details are different. So it is not possible to simply transfer the models of traditional software development to mobile application development without making significant amendments [1].

## 2 Key features of software for mobile devices

As more companies build their own mobile apps, and as many make the transition from desktop to mobile (or mobile web) apps, it is very important for developers with no prior experience in mobile development to understand the main key differences in the process of creating mobile apps and traditional web / desktop development.

When creating a mobile version of a website or application, developers need to be sure to think about how the target user of the application will interact with the software. Mobile screens are much smaller, interactions tend to be much faster, and expectations are very different from those for desktop applications. It is almost impossible to transfer an entire website or desktop application to a mobile device, users of such an application will be overwhelmed, and on a 5-inch screen it just won't make any sense. It's better to pick the features that matter most to mobile users and focus on making those features really work well.

Another important difference between mobile applications and desktop and web applications is that mobile devices use a touchscreen interface that is less accurate than a mouse interface. For example, the mouse-based interface is accurate. Tiny buttons and small interactive areas work. Button location is not a problem. The touch interface is not accurate. This requires buttons that are large enough to be pressed with your thumb. Also, the location of the button is very important. The buttons should be within reach, the area of the screen that is accessible to your thumb.

Designing an application for touch requires a fundamental development shift. From various studies, buttons should be at least 72x72 pixels to account for the average thumb width. In addition, it should be borne in mind that users usually hold their phones with one hand. Therefore, the interface should be designed with this fact in mind - keeping common tasks in the lower half of the screen for easy access [2].

## 3 Main steps of every lifecycle model

As the mobile applications have complex functionality and are different from the desktop applications, the following Mobile Application Development Lifecycle model (MADLC) is proposed to enable a systematic approach in development.

- **A. IdentificationPhase** In the first phase, ideas are collected and categorized. The main objective of this phase is to come out with a new idea or

improvements to the existing application. The ideas can come from the customer or from the developers. If the customer himself comes out with the idea, the idea is further detailed and analyzed. Developers can brainstorm to generate ideas for new applications. The filtered list of ideas is discussed by the mobile application idea team comprising of the business and IT representatives for the feasibility to launch a project around the idea. The existing applications on any of the standards platforms are searched to establish the novelty of the idea. If a similar application exists in the market, the popularity of the application and the features supported are studied and compared. The differences with the existing application(s) are documented. If no similar application exists on any mobile platform, then the idea with its core functionality should be documented. The other important task in this phase is to define the time required to develop the application. The initial requirement gathering should also be completed. The work done by the mobile application idea team should then be documented and forwarded to the design team.

- **B. Design Phase** In this phase, the idea from the mobile application team is developed into an initial design of the application. The feasibility of developing the application on all mobile platform is determined. Alternatively, the specific target mobile platform is identified. A decision has to be made on whether the developed application is to be released as a free version or trial version with limited features or released only as a premium version. The application functionality is broken down into modules and into prototypes i.e., combination of modules which are to be released in the prototype fashion. The functional requirements are defined. The software architecture of the application is created. Then the prototypes and associated modules are defined. A very important part of the design phase is to create the storyboard for the user interface interaction: this storyboard describes the flow of the application. The design team's work is documented and forwarded to the development team for coding.
- **C. Development Phase** In this phase, the application is coded. Coding for different modules of the same prototype can proceed in parallel. The development process can be in two stages: Coding for Functional Requirement and Coding for UI requirements. The code is developed first for the core functionalities. Parallel development can be done for modules of the same prototype that are independent of each other. Subsequently, these modules can be integrated. In the second stage, user interface is designed so that it can be supported on as many mobile operating system platforms as possible; it is not good practice to have a different look and feel for the same application on different platforms. The minimum set of interface components present in all mobile OS platforms should be used in the design. Finally, the documentation of the development phase is then forwarded to the prototyping phase.
- **D. Prototyping Phase** In this phase, the functional requirements of each prototype are analyzed; the prototypes are tested and sent to the client for feedback. After feedback is received from the client, the required changes are implemented through the development phase. When the second prototype is ready, it is integrated with the first prototype, tested and then sent to the

client. The development, prototyping and testing phases are repeated until the final prototype is ready. The final prototype is sent to the client for a final feedback. The work done in this prototyping phase is documented and then forwarded to the testing phase.

- **E. Testing Phase** Testing is one of the most important phases of any development lifecycle model. The testing of the prototype types is performed on an emulator/simulator followed by testing on the real device. The emulator/simulator is often provided in the SDK. The testing on the real device, for example in the case of Android operating system development, should be performed on multiple operating system versions, multiple models of handsets with variable screen size. The test cases are documented and forwarded to the client for feedback.
- **F. Deployment Phase** Deployment is the final phase of the development process. After the testing is completed and the final feedback is obtained from the client, the application is ready for the deployment. The application is uploaded to the appropriate application store/market for user consumption. Before the application is deployed, the following steps are to be checked.
- **G. Maintenance Phase** The maintenance is the final phase of this model and this maintenance is a continuous process. Feedback is collected from users and required changes are made in the form of bug fixes or improvements. Appropriate security patches, performances improvements, additional functionality, new user interfaces should be provided at regular intervals in the form of updates to the application. The maintenance phase also includes the marketing of the application: advertising and highlighting its unique features. If any application requires a backend server: this server and related operating system must be maintained as well [3].

## 4 Review of software development lifecycle models

### Literatúra

- [1] Anureet Kaur and Kulwant Kaur. Suitability of existing software development life cycle (sdlc) in context of mobile application development life cycle (madlc). *International Journal of Computer Applications*, 116(19), 2015.
- [2] Johnson Nicolaus and Moazed Alex. *Platform: Practical using of revolutionary business model*. Alpina Publisher, 2019.
- [3] Tejas Vithani and Anand Kumar. Modeling the mobile application development lifecycle. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 1, pages 596–600, 2014.