

APPLICATION DEVELOPMENT WITH ANDROID: A REVIEW

Prof. Bhushan Thakare¹, Rachana Bhole², Sanket Wankhede³, Kajal Warade⁴,
Suraj Patil⁵, Subhan Khan⁶

¹Assistant Professor, Computer Science and Engineering Department,
Padmashree Dr. V.B.Kolte College of Engineering, Malkapur, Maharashtra, India

^{2,3,4,5,6}Final Year Student, Computer Science and Engineering Department,
Padmashree Dr. V.B.Kolte College of Engineering, Malkapur, Maharashtra, India

ABSTRACT

In the advancing world of technology, Mobile applications are a rapidly flourishing segment of the global mobile market. Smartphone comprise of an important features through which user can easily distribute applications via online market store. Mobile applications are evolving at a rapid rate to give user a rich and quick user experience. Google released Android in 2007 which is an open-source Mobile phone operating system with Linux-based platform. It consists of the operating system, middleware, and user interface and application software. Android tries to allow users experience the best service quality, and allow developers get a more open level for more convenient software developing. Thus mobile applications with more convenient functions can be developed via Android. In this paper, Android mobile platform for the mobile application development, layered approach and the details of core features of android platform is discussed in this paper.

Keywords:- android, application, segment, open -source, platform.

1. INTRODUCTION

Mobile applications are becoming increasingly prevalent today, particularly in the world of business. As such, many approaching business students are interested in creating mobile applications but lack the knowledge to do so. Android operating system is a linux-based operating system. It is developed by google in 2007. In 2008 the first version of android in launched officially. It is an operating system in which each app runs within a distinct identity. The Linux kernel provides android with several security features as a base for a mobile computing environment. In Android operating system, it provides facility to isolate user resources from one another so acting as multi user operating system. The Android system assigns each application with a unique user ID and runs hit as a separate process from other users.[1]

2. METHODOLOGY

The mobile app development process is a project that requires a unique form of management. Taking control of the process means coordinating the various underlying activities and resources involved that will lead to success. The type of framework to choose for your project can be determined if you go through the list of new and traditional options we have provided in this article. Here are some of the top five methodologies you can use in your mobile app development endeavor.

2.1 Waterfall Methodology

Before user experience was a determining factor in mobile app development, the waterfall model was the apple of the eye of every developer. The model executes phrases sequentially; you need to complete one phase entirely before moving to the next stage. The worst thing about this methodology is that there was no turning back. It is a complete process from the start to the end. You could only make changes in the maintenance stage, which comes after launching the product.

2.2 Prototype Methodology

Also known as evolutionary design or incremental development. The model is a refined version of the waterfall methodology allow making changes during the design phase with the use of prototypes as the name suggests. This solution is for the idea that you need a prototype to fully understand the requirements of the project before you proceed with its design. This framework subjects the prototype to client evaluation before it proceeds to the development stage. The prototyping methodology managed to reduce the risk of failure through the use of prototyping and client feedback.

2.3 Spiral Methodology

The spiral methodology is a risk-driven model in mobile app development. The framework relies on the risk pattern of the project and uses more than one method in its process. It is a combination of waterfall methodology and iteration. You are required to set requirements for each phase of the framework. The number of stages in the spiral the model highly depend on the project risk. It is, therefore, suitable for large and complex projects with high risks.

2.4 Agile Methodology

The Agile model is a project management methodology purposely adopted for the development of sophisticated software. The framework allows for iterations, which helps a lot in minimizing mistakes and errors that commonly occur. The model divides the project into a series of development cycles or short time boxes, which are assigned to each professional on the project team. It is a collaborative approach the allows a response to rapid change. It is flexible enough to accommodate changes in project requirements throughout the mobile app development lifecycle.

Other methodologies fall under the Agile umbrella include:

- Feature Driven Development (FDD) – a lightweight and incremental model that focuses on features as the name suggests. It features a series of iterations and inspections. This framework demands a high level of design expertise and planning.
- Lean software development – it is an integration of the agile methodology and lean manufacturing principles and practices. Aim at optimizing time and reducing waste, cost and effort.
- Scrum – focuses on the management aspects of software development in intricate knowledge work, research and advanced technologies with an emphasis on teamwork, iteration and accountability.
- Crystal Methods – It is one of the most lightweight agile methodologies. It focuses on team member talent skills, interactions and communication—this model groups projects in terms of system criticality, team size and priorities.
- Rapid Application Development (RAD)/ Rapid-application building (RAB)- focuses on timely delivery in a fast-paced environment with the use of prototyping and iterative development.
- Adaptive Software Development (ASD) – It is an outgrowth of the RAD that provides continuous adaptation to change in project requirement or market needs.
- Dynamic Systems Development Method (DSDM)- it is an iterative and incremental Agile approach based on RAD, but with governance and strict guidelines. It is applicable in four principal phases.
- Extreme Programming (XP) – Focuses on software quality and responsiveness with emphasis on the changing needs of the client. It features a high-level collaboration with minimal documentation.
- Kanban– a framework that utilizes a lean manufacturing scheduling system to develop software in a long development cycle. The aim is to manage and improve flow systems in mobile app development.

2.5 Lean Startup Methodology

The main idea of the methodology to offer a scientific solution for startups. Since it is a principled approach to new product development, the framework has found application in mobile app development. This approach requires a minimum viable product (MVP) to initiate mobile app development. The lean startup model lets you build, measure and learn from the mobile app you develop. It focuses on building and launching mobile applications fast and at a lower cost.

2.6 Bottom Line

There are enough methodology options for your mobile app development venture. Choosing the right methodology requires that you understand the nature of your application development project. There is always a solution for you if you think in terms of scope, time, change, risk, cost and quality.[2]

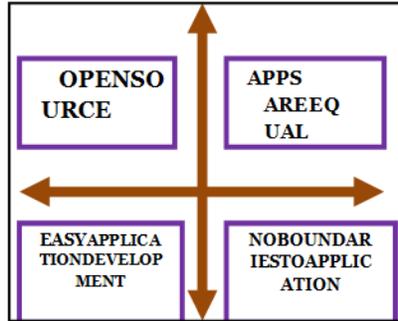


Figure1: Core features of android

3. ANDROID PLATFORM ARCHITECTURE

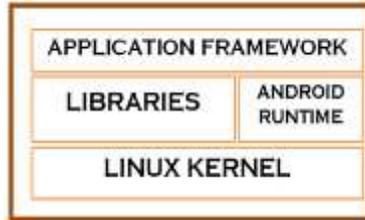


Figure2: Android Architecture

3.1 Application Framework:

Android applications are written in the Java programming language. The java code is assembled with the data related to application and resource files. All these is bundled by the app tool into an android package. It is an archived file with suffix of .apk. this file works as a lorry for issuing the application which is installed on mobile devices. These file is considered to be an one application which can be taken by user.

In many ways, each Android application lives in its own world:

- By default, every application has its own Linux course. Android run the application when it is desired by the user and close the application when it is no longer essential. As the assets are required by the system.
- Each application code run separately. Each process has its own JVM Java virtual machine
- By avoidance ,each app is allotted a unique user ID of Linux Consents are set so that the application's files are detectable only to that user. It's possible to organize for two applications to part the same user ID, in which case they will be able to see each other's files. To conserve system resources, applications with the same ID can also arrange to run in the same Linux process, sharing the same VM.[3]

3.2 Android Runtime

Android Run-time is the equivalent of JVM (Java Virtual Machine)that runs on your Windows and Linux machines .Only difference is that it's called DVM (Dalvik Virtual Machine),because it's a different software than what Sun Micro-system shad developed Android Run time is made up of a Core Java Libraries and Dalvik virtual machine (The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory footprint). Most functions of the core java International libraries is provided by Core library. DVM is register virtual machine and makes some specific developments for mobile devices. Android system library is reinforcing the application framework; it is also a main connection linking between application Framework and Linux Kernel. This system library is enlarged in C or C++ language. These libraries can also be used by the different elements in the Android system.[4]

3.3 The Linux Kernel

The Linux Kernel is at the bottom of the architecture. It provides a level of abstraction between the device hardware and the upper layers of the Android architecture. Based on Linux version 2.6, the kernel delivered preemptive multitasking, low-level core system services such as memory, process and power management in addition to providing a network stack and device drivers for hardware such as the device display, Wi-Fi and audio.

The original Linux kernel was developed in 1991 by Linus Torvalds which was combined with a set of tools, utilities and compilers developed by Richard Stallman at the Free Software Foundation to create a full operating system referred to as GNU/Linux. Various Linux distributions have been derived from these basic underpinnings such as Ubuntu and RedHat Enterprise Linux.

It is important to note, however, that Android only uses the Linux kernel. That said, it is worth noting that the Linux kernel was originally developed for use in traditional computers in the form of desktops and servers. In fact, Linux is now most widely deployed in mission critical enterprise server environments. It is a testament to both the power of today's mobile devices and the efficiency and performance of the Linux kernel that we find this software at the heart of the Android software stack. Also, the kernel takes care of all the things that Linux is great at such as networking and a huge array of device drivers, which take the burden out of interfacing to peripheral hardware. The kernel system provides the operations like internal storage, process management, internet protocol, bottom-drive and other core services are all based on Linux kernel.[5]

4. FUTURESCOPE

On the face of it, Android seems like a very successful platform. It's the operating system that powers 85 percent of the new smart phones shipped, and, along with iOS, has decisively crushed the competition. But Android is also plagued by dogged problems that Google can't seem to solve.

First off, there's the issue of fragmentation, which means that developers have to create software that will work reliably on many hundreds of different devices from dozens of manufacturers. Sure, developers could just focus on devices from the big players - and some, such as Sales force, are already doing just that - but that defeats the purpose of having a platform in the first place. Then, there's the update problem. Google has a yearly release schedule in place for Android, but it takes the new version a good part of a year to break the 10-percent usage mark. And it takes about four years for a release to fully penetrate the ecosystem. Four years is a crazy long time in tech. While Google is speedy at getting new releases onto its Nexus hardware, it can take months for owners of Android devices from other manufacturers to start seeing the update. And many never see the update. In fact, outside of buying a Nexus device, the easiest way to get your hands on a new Android release is to buy a new device.[6]

5. CONCLUSION

There are number of utilities for implementing application services one of them is android. And only because of this consistent platform, any developer is able to perform their thoughts into reality. Also, with the help of various libraries, we can implement tremendous features in an android application that can be used worldwide.

6. REFERENCES

- [1] [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))
- [2] http://file.scirp.org/Html/2-9302032_54397.html
- [3] <https://docs.microsoft.com/en-us/xamarin/android/app-fundamentals/android-api-levels>
- [4] <https://dev.to/dhruvitshukla/android-developers-have-switched-from-the-java-programming-language-to-kotlin-4fa>
- [5] https://google-developer-training.Gitbooks.io/android-developer-fundamentals-courseconcepts/content/en/Unit%201/33_c_the_android_support_library.html
- [6] <https://stackoverflow.com/questions/18271429/difference-between-android-support-v7-appcompat-and-android-support-v4>
- [7] <https://developer.android.com/studio/debug/dev-options.html>
- [7] <https://www.androidcentral.com/why-you-ll-never-have-latest-version-android>