

INF4188: Development of Mobile Applications

Mobile phone development environments



```
fun cases(obj: Any?) {  
    when (obj) {  
        1 -> println("One")  
        "Hello" -> println("Greeting")  
        is Long -> println("Long")  
        is String -> println("Not a string")  
        else -> println("Unknown")  
    }  
}  
  
back.on("save-image", function(filepath, filename, data){  
    console.log(data);  
    var buffer = new Buffer(data, "base64");  
    fs.writeFileSync(path.join(filepath, filename), buffer);  
    console.log("saved image");  
})  
  
int addSomeNums(int x, int y, [int z]) {  
    int sum = x + y;  
    if (z != null) {  
        sum += z;  
    }  
    return sum;  
}
```

The image displays two programming language logos: the Kotlin logo (a blue and red stylized 'K' with the word 'Kotlin' to its right) and the Flutter logo (a blue and green stylized 'F' with the word 'Flutter' to its right).

Azanzi Jiomekong

University of Yaounde I, Department
of Computer Sciences

October 28, 2022



Copyright (c) 2013 Jiomekong Azanzi Fidèl.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. More information about the license is included in the section entitled " The GNU Free Documentation License".



This support is under construction. To help me improve it, please send your comments to jiofidelus@gmail.com.



Objectives (1/2)

At the end of this course, the students should know:

- Technologies and tools for mobile phones application development
- Frameworks for bootstrapping mobile application development
- Advices on the technologies to adopt when developing applications for mobile phones



Objectives (2/2)

At the end of this course, the students should know:

- Set up an Android development environment
- Set up an iOS development environment
- Set up a Web based mobile development environment
- Set up a cross platform mobile development environment
- Set up a compiled based mobile development environment

Overview



- Types of mobiles applications
- Native mobile applications
- Mobile Web application
- Cross-platform mobile application
- Compiled mobile application



Types of mobile application

- Native: Applications are written using a native language
- Web Mobile: Web sites for mobile devices
- Hybrid: also called Crossplatform mobile application, it is the combination of Native and Web app
- Compiled: applications created and compiled to a native code

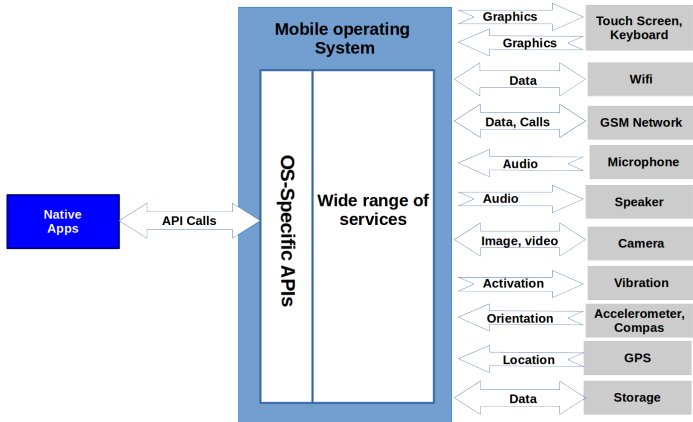


Native mobile applications

- Most common type of mobiles app
- Rich UI experiments
- Build for specific Platform
- Writes the app in the native language for the given Platform
- Android provides framework based on Java language
- iOS provides a native framework based on Objective-C / Swift language

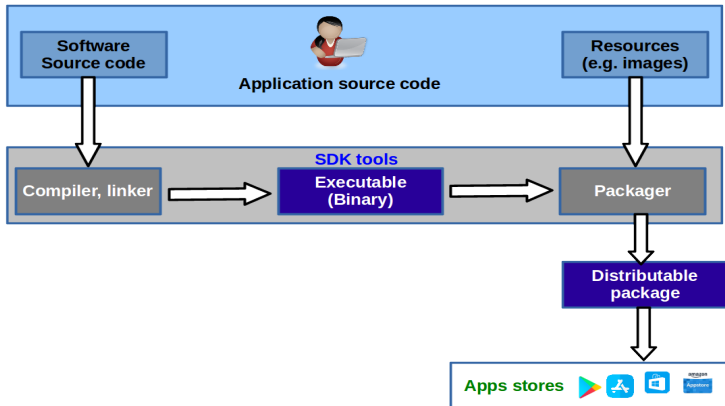
Native mobile applications

Interaction with the phone



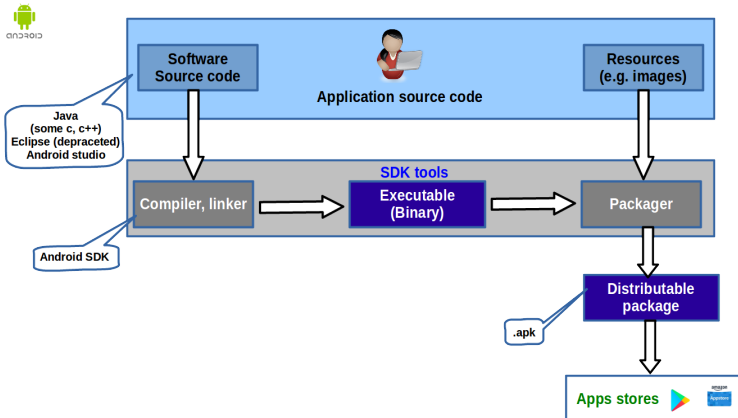
Native mobile applications

Development process



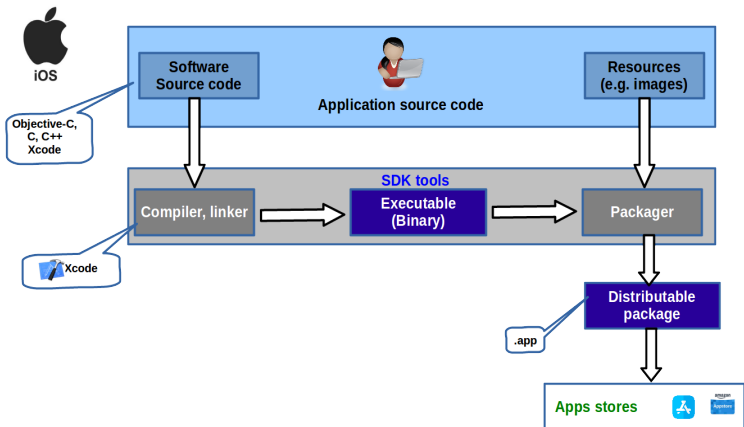
Native mobile applications

Android



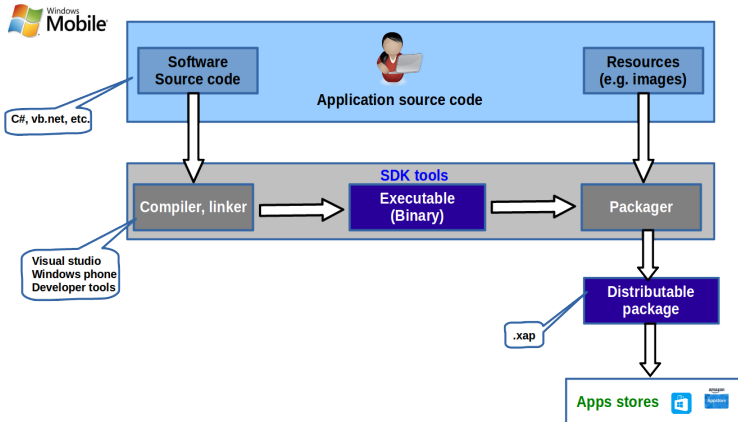
Native mobile applications

iOS



Native mobile applications

Windows mobile





Native mobile applications

Keys technologies

Android:

- Programming language: Java, Android
- Software Development Kit (SDK): JDK, Android SDK
- Integrated Development Environment: Eclipse (deprecated), Android Studio

iOS:

- Programming language: Objective C, Swift
- Software Development Kit (SDK):
- Integrated Development Environment: XCode



Android

Development tools

- OS: Microsoft Windows, Linux, Mac OS
- SDK: JDK 5, Android SDK
- IDE: Android Studio, IntelliJ IDEA, etc.
- Android simulators: some are installed on Android SDK

Android

Development tools



titanium



Android NDK



Android

Set up the development environment

- Download and install JDK
- Download and install Android SDK
- Set the PATH and JAVA_HOME environment variables to refer to the library
- Download and install Android Studio



iOS

Development tools

- Device: any Apple device such as MacBook Pro, Mac Mini
- OS: OS X operating system
- IDE: Xcode
- Apple developer account
- Simulator: virtual iPhone/iPad, installed
- A real device to use certain devices features



iOS

iOS developers

iOS developers community

- Worldwide Developers Conference (WWDC) videos
- Join iOS developer program teams when invited.
- Register for the iOS Developer Program: helps developers to provide quality applications to users

Benefits to join the developer program:

- Run the apps you develop on the real iOS device.
- Distribute the apps to the app store.
- Get access to the developer previews.



iOS

Set up the development environment

- XCode includes everything to develop OS X, Watch OS, TV OS applications
- To use it, develop must have:
 - A device running OS X
 - SDK
 - An Apple ID
- iOS simulator consist of two types of devices:
 - iPhone device with different versions
 - iPad device with different versions



Native mobile applications

Advantages & Disadvantages

Advantages:

- Proven path for mobile development
- Very fast
- Distributed app stores: Google Play, Windows Store, ...
- Can be approved very easily
- Interactive and intuitive
- Interact with devices utilities

Disadvantages:

- Single Platform
- Harder language
- Very expensive
- Hard to maintain

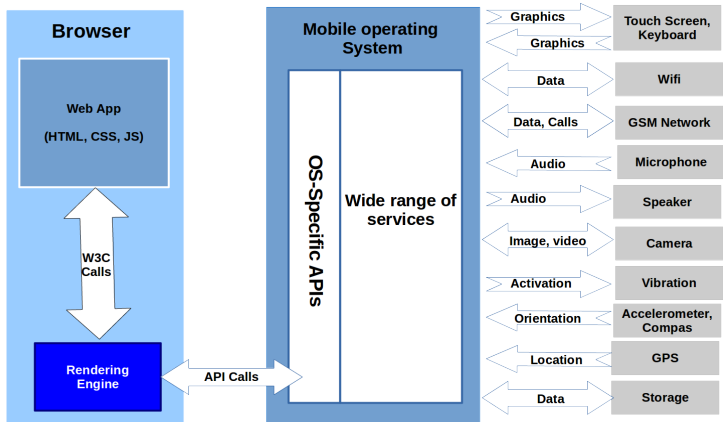
Web Mobile application development



- Build like a regular Web Site or Web App
- Runs inside a browser

Web Mobile application development

Interaction with the phone





Web Mobile application development

Keys technologies

- Programming language: HTML, CSS, JS
- Frameworks: NodeJS, Sencha Touch, Ratchet, JQuery mobile, Angular, React, VueJS
- IDE: Eclipse, Netbeans, Visual Studio Code, Sublime Text, etc.
- Testing: mobile phone itself, Chrome dev tool, Riple
- Useful CSS libraries and frameworks:
 - Bootstrap, Materialise, Bulma, UIKit, Meteor, Foundation, Semantic UI
 - animate.css, imagehover.css, loading.io, hint.css, skeleton.css, mini.css, Font awesome



JQuery Mobile

- Open source UI framework
- Build on jQuery core and developed by the jQuery project team in 2010
- Used for developing responsive websites/applications
- Applications developed are accessible on mobile, tablet, and desktop devices
- Uses features of both jQuery and jQueryUI to provide API features for mobile web applications
- Provides a set of touch-friendly form inputs and UI widgets



JQuery Mobile

Advantages

- "Write less, do more" UI framework
- Cross-platform and Cross-browser compatible
- Compatible with other frameworks
- Easy to learn and develop applications
- Includes ThemeRoller for creating custom theme without writing the line of code
- The code scale automatically from the mobile screen to desktop screen



JQuery Mobile

Disadvantages

- Limited options for CSS themes
- Slower on mobiles
- Time consuming when combined with other mobile frameworks
- Difficult to provide complete customized visual design
- All the devices features are not accessible



JQuery Mobile

Development tools

- OS: Windows, Linux, OS X
- Programming languages: HTML, CSS, JS
- Frameworks: jQuery Mobile
- IDE: Eclipse, Netbeans, IntelliJ, Sublime Text, etc.
- Miscellaneous: large collection of demos and the library's unit test suite



Ionic

- Open source front-end HTML framework build on top of Angular and Apache Cordova
- Provides tools and services for building Mobile UI with native look and feel
- Need native wrapper to be able to run on mobile devices
- Handles all the look and feel and UI interactions of mobile applications
- Support a broad range of common native mobile components



Ionic

Features

- Angular MVC architecture: for building rich single page applications optimized for mobile devices
- CSS components: involved almost all elements a mobile app needs with the native look and feel
- JavaScript components: cover all mobile elements that cannot be done only with HTML and CSS
- Cordova Plugins: used to access native device functions with JavaScript code



Ionic

Features

- NodeJS utility: for starting, building, running and emulating Ionic applications
- Ionic View: for uploading, sharing and testing applications on native devices
- Components can be overridden to accommodate developer designs
- Pre-generated app with simple layouts
- Build apps are easy to maintain and to update



Ionic

Development tools

- OS: Windows, Linux, OS X
- Programming languages: HTML, CSS, JS
- Frameworks: Ionic, NodeJS
- SDK: Android SDK
- IDE: all HTML and JS IDE and XCode
- XCode: if the application is for iOS

React



Coming soon

Angular



Coming soon



Web Mobile

Advantages

- Single app for all Platforms
- Use any Web language/technology
- Easy to build and easy to maintain
- Install locally to the device and runs as an installed app
- Does not require an app store
- Cheaper than native and hybrid



Web Mobile

Disadvantages

- Cannot interact with some devices utilities
- Slower than native app
- Less interactive and less intuitive
- No icon on desktop
- Cannot be submitted to App Stores



Web Mobile

Disadvantages

- Need to run in a browser
- Some hardware and software on the device cannot be used
- Required an Internet connection to work (HTML5 - solves this limitation)

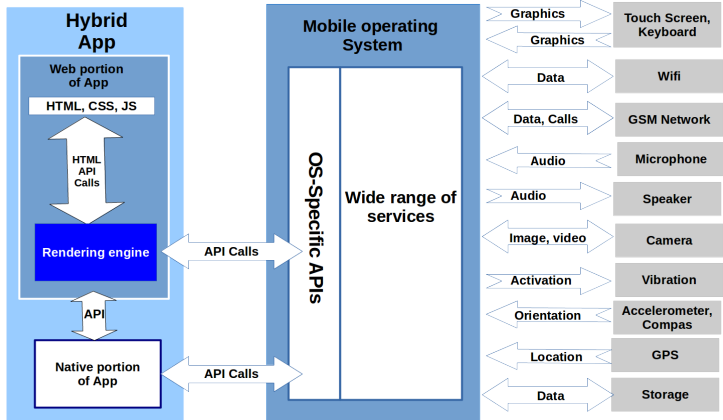


Hybrid Mobile application development

- Run inside a container/WebView through a framework
- App hosted in a native WebView component (Wrapper)
- WebView: UI component available on Mobile OS

Hybrid Mobile application development

Interaction with the phone



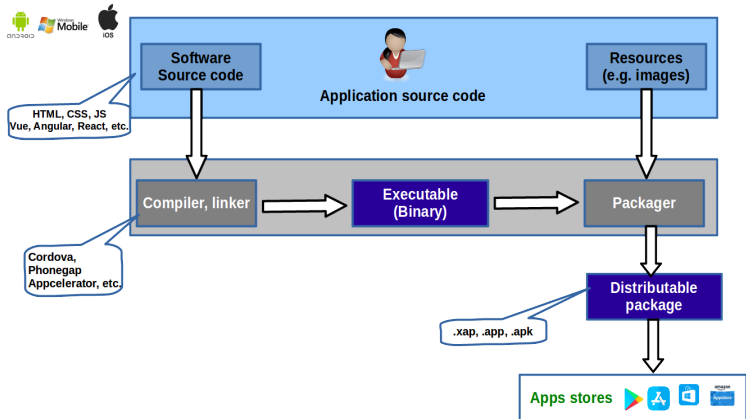


Hybrid Mobile application development

Key technologies

- Programming languages: HTML, CSS, JS
- Frameworks: jQuery mobile, Sencha Touch, Angular, ionic, Vue.JS, Cordova, PhoneGap, Appcelerator, Whitelight, etc.
- IDEs: Eclipse, Netbeans, Sublime Text, Visual Studio Code, etc.

Hybrid Mobile application development Technologies





Hybrid Mobile application development

Cordova

- Open source framework
- Used to make hybrid mobile apps that can use many mobile phones features
- Offers a bridge for connection between web app and mobile device
- Applications execute within wrappers targeted to each platform
- Offers one platform for building hybrid mobile apps: IOS, Android, Windows Phone, Amazon-fireos, blackberry, Firefox OS, etc.
- Save the development time compared to the development of native apps



Hybrid Mobile application development

Cordova: Features

- Cordova CLI for:
 - Starting projects
 - Building processes for different platforms
 - Installing plugins
- Cordova core components: set of core components that every mobile application need
- Cordova Plugins: API used for implementing native mobile functions



Hybrid Mobile application development

PhoneGap

- Mobile development framework developed by Adobe System
- Produces apps for all popular mobile OS platforms: iOS, Android, BlackBerry, and Windows Mobile OS etc.
- Allows users to upload data contents on website and it automatically converts it to various App files
- Applications are uploaded on <https://build.phonegap.com/> and the mobile application is generated automatically



Hybrid Mobile application development

Cordova & PhoneGap: Development tools

- OS: Windows, Linux, OS X
- Programming languages: HTML, CSS and JS
- Frameworks: Cordova framework, NodeJS
- IDE: XCode, all HTML and CSS IDE



Hybrid Mobile application development

Advantages & Disadvantages

Advantages:

- Multi-Platform: Single app for all Platforms
- Fast to develop and update
- Access to many devices utilities using APIs
- No browser needed
- Much Cheaper than a native app

Disadvantages:

- Slower than native apps
- Less interactive than native apps
- More expensive than Web Apps

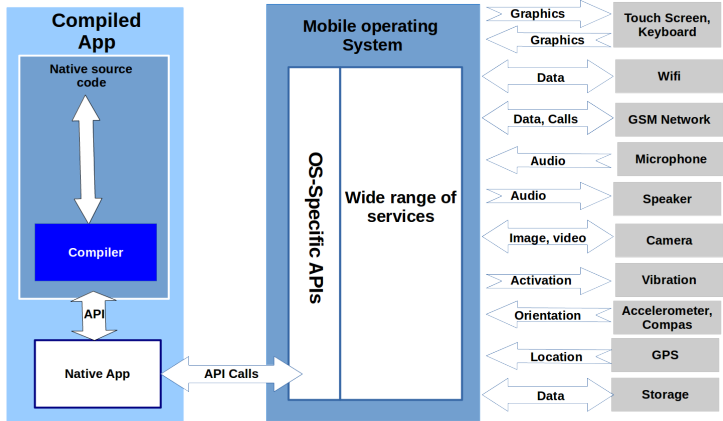
Compiled



- Essentially native apps
- Uses Web technologies, but convert the apps into native components
- Close to native apps in terms of performance

Compiled Mobile application development

Interaction with the phone





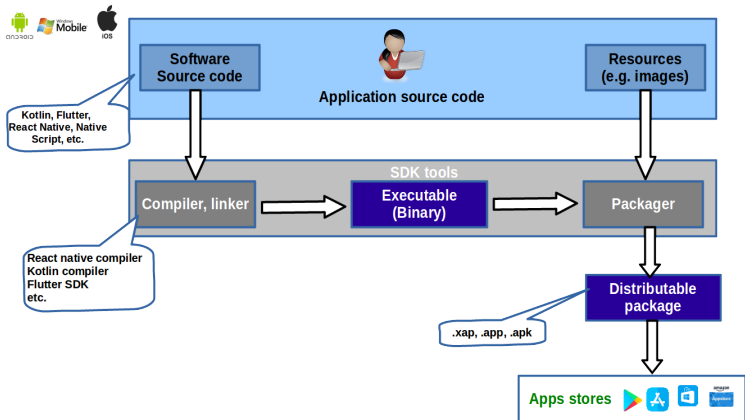
Compiled

Key technologies

- Programming languages: HTML, CSS, JS, Dart
- Frameworks: React Native, Kotlin Native, Native Script, Flutter, etc.

Compiled Mobile application development

Technologies





Compiled Mobile application development

Kotlin

- Open source programming language introduced by JetBrains
- Influenced by other programming languages: Java, Scala, Groovy, etc.
- Combines object-oriented and functional features into a unique platform
- Set as the official programming language in 2017 by Google



Compiled Mobile application development

Kotlin: Advantages & Disadvantages

Advantages:

- Easy to learn
- Better performance and small runtime

Disadvantages:

- Namespace declaration: the functions must be declare at the top level
- No static declaration: can cause some problem to the conventional Java developer



Compiled Mobile application development

Kotlin: Development tools

- OS: Windows, Linux, OS X
- SDK: JDK 8 or above
- IDE: Eclipse, Netbeans, IntelliJ IDEA
- Kotlin plugin for Eclipse



Compiled Mobile application development

React Native

- JS framework for building native Android and iOS mobile apps
- Designed on the head of React by Facebook
- Comes with fast development cycles and declarative self-contained UI components
- Offers large amount of inbuilt components and APIs
- Developers can use native components controlled by JS
- Apps developed are indistinguishable from app built using Objective-C or Java



Compiled Mobile application development

React Native: Development tools

- OS: Windows, Linux, OS X
- Programming languages: HTML, CSS, JS, Python
- Frameworks: NodeJS, create-react-native-app
- SDK: Python, JDK 8 or above, yarn
- IDE: Android Studio, XCode
- Miscellaneous:
 - React Native CLI
 - "expo": client in the Android device and scan the QR code
 - yarn, npm



Compiled Mobile application development

NativeScript

- Open source framework developed by Progress Telerik
- Used for creating native iOS and Android apps



Compiled Mobile application development

NativeScript: Development tools

- Programming languages: HTML, CSS, JS, XML
- Frameworks: Angular, TypeScript, JS, NodeJS
- SDK: Android SDK, JDK 8 or above
- IDE: Playground which is a WebIDE, XCode
- Miscellaneous: tns doctor to verify the dependency and summarize the results



Compiled Mobile application development

Advantages & Disadvantages

Advantages:

- Developers can use existing JS knowledge to build native mobile apps
- Code can be share among platforms

Disadvantages: Need to write platform specific code for new native components → tedious task



Compiled Mobile application development

- Write one code and compile to mobile apps of different platforms
- The applications obtained after the compilation don't run in a web view



Compiled Mobile application development

Flutter

- Open source framework to create high quality, high performance mobile applications
- Allows to developed applications for Android and iOS
- Modern and reactive framework
- Provides SDK to write mobile application in Dart
- Dart: programming language developed by Google



Compiled Mobile application development

Flutter: Advantages

- Simple, powerful, efficient and easy to understand
- Runs same UI for multiple platforms
- Lesser testing: tests for both platforms
- High performance application
- Beautiful and customizable widgets
- Fast development and great developers tools



Compiled Mobile application development

Flutter: Disadvantages

- Developers needs to learn new language
- UI and logic are intermixed: smart coding can be use to separate them



Compiled Mobile application development

Flutter: Development tools

- OS: Windows, Linux, OS X
- SDK: Flutter SDK, Android SDK
- IDE: Eclipse with Flutter and Dart plugins, Android studio / XCode
- An Android emulator / iOS simulator or a real android device
- Miscellaneous: Flutter doctor to check if all the requirement of flutter development is met



Difficulties when developing mobiles apps

- Separation between operations and development teams
- Lack of collaboration between development team
- Lack of communication between development team

Solution: Break the wall between the development and the operation



Mobile DevOps

DevOps overview

- Software development approach
- Focuses on the effective collaboration between software developers, operations staff and project managers
- Allows to overcome the challenges associated with continuous software delivery
- Used to make continuous integration and continuous delivery more efficient
- Dev and Ops teams came together to meet the challenges of continuous application delivery



Mobile DevOps

DevOps overview

- Facilitate seamless and timely delivery of mobile applications from design to final production, delivery, and deployment in the app store
- Key principle:
 - Continuous integration:
 - Continuous delivery: deploying the software and the environment on which it runs, automatically and on-demand, during any stage of the development cycle

Mobile DevOps



- Continuous mobile apps delivery
- Enables clients to reduce time-to-market, new features
- Seize new market opportunity easily
- Challenges:
 - Different OS vendors
 - Different OS versions
 - Different Smartphone manufacturers



Mobile DevOps

Adoption

- Continuous Planning: bring the team (devs, managers, operations, etc.) on the same platform to finalize the scope of the mobile app in the preparation of a launch plan
- Continuous Integration:
 - The use of the source code control system
 - The code submitted by one developer work with the code submitted by the others
 - Frequently builds integrated with the developed code
- Continuous Testing:
 - Impossible to cover the large game of mobiles devices (manufacturers, OS) with manual tests
 - → Welcoming automatic testing



Mobile DevOps

Adoption

- Continuous Monitoring: done via third-party SDKs
- Continuous delivery: deploy each change to the production-like environment
- Continuous deployment:
 - Next step of Continuous Delivery
 - Each change approved by Continuous Testing gets deployed to the production environment automatically



Mobile DevOps

Why?

- Mobile requires rapid iterations
- Automating the deployment:
 - Reduces the time it takes to roll out new iterations
 - Enables teams to react faster to user feedback
- Mobile apps are less forgiving: users are much less forgiving when it comes to worse user experience
- Mobile tools are constantly changing: new technologies for keeping up with the ever-growing market needs



Mobile DevOps

Why?

- Mobile DevOps enables:
 - Instant reports and feedback
 - Developers to troubleshoot errors and fix bugs faster
- Certain features might not work as expected on all supported platforms
- Apps performing well during tests but failing in the hand of user is a common scenario



Mobile DevOps

Tools

- git, svn, Github, GitLab, etc.: Source code management
- Integrated mobile development lifecycle solution, etc.: Visual Studio App Center, Azure DevOps
- Jenkins: Continuous integration with
- pCloudy, Appium, Appcelerator, etc.: Continuous testing



Mobile Devops

How to include DevOps in your mobile application project?

1. Agile Planning: determine the scope of the mobile application and plan its release
2. Constant Integration: follow the best standards and practice of version control
3. Proper Testing: make sure that the product delivered to the customer is of good quality
4. Precise Monitoring:
 - The mobile app is performing well as per the requirements
 - The mobile application is stable in the environment of production
 - Faster Delivery: ensure that the source code is ready to deploy in a production environment
 - 74 / 84 Make the stable version available for deployment as soon as possible



Mobile Devops

How to include DevOps in your mobile application project: Summary

1. Continuous Integration and Delivery:

- The code should be easily integrated with others
- App development assets (scripts, text files, configurations, documents, code, etc.) are traceable

2. Testing and Monitoring:

- Make tests in real environment by considering screen size, resolution, capabilities, features, etc.
- Test the app in a continuous basis
- Automated app testing allow us to improve the testing process, identify bugs very quickly

3. Quality Control:

- Monitor ratings and feedback on the Mobile App Store constantly
- Tackle the bugs and pave the way for future change and growth



Mobile DevOps

Benefits of DevOps in mobile application development

- More time for innovation
- Better collaboration
- Quick delivery time
- Stable and secure environment for easy deployments
- Prompt bugs fixes
- Automating repetitive tasks without worrying too much about errors
- etc.



Choosing a Mobile Development approach

- Great variety of OS and devices types, all evolving at different rates
- Great variety of development frameworks and tools for each mobile development approach
- Success depends: quickly delivering applications with the quality and features that users demand, across a wide range of screen sizes, presentation formats and device capabilities
- The rapid proliferation of smart phones and release of new tablets have made choosing the right mobility solution more complex than ever



Choosing a Mobile Development approach

Use case 1

The primary users are physician and other healthcare staff, who are accustomed to a rich experience on their mobile device. The functionalities involved multiple communication channels including chat and VoIP phone calls.

A good choice should be Native Android or iOS



Choosing a Mobile Development approach

Use case 2

The existing application allows users to shop online using their mobile devices. The client want to add a new channel for their consumers. The largest number of consumers must be reach while providing services across multiple device types.

Mobile Web-based solution



Choosing a Mobile Development approach

Use case 3

The goal of the company is to build a system that will run on multiple smartphones (iPhones, Android and BlackBerry devices). The main features are: retrieving pricing information, entering line items on a quote, getting the GPS data on the customer location. Although not familiar with native technologies, the IT team was experienced with JS and CSS, which are two base technologies for PhoneGap

Cross-Platform Mobile application with PhoneGap



Why develop applications for smartphone?

- Internet access everywhere
- Social networks
- Million of users
- Open standard
- Change on the way people work, shop, stay informed, etc.
- Many use cases:
 - Sell products and services
 - Raise productivity and increase Awareness
 - Getting access to information
 - etc.



Keys questions to technologies choosing

- Who will use the app?
 - Audience identification: allows to know platforms to be supported and the type of user experience to provide
 - Platform requirement is the key input to the technology selection
- What type of experience do users expect?
- What functionality is required?
 - A higher amount of informational requirements
 - A higher amount of transactional requirements
 - Device-specific requirements
 - Application requires offline access of large files such as PDF catalogs



Keys questions

- How important is multi-platform compatibility?
- Does the app need to run offline or when the connection is bad?
- Are there industry-specific requirements?
- How much time do you have to build the app?
- Does your staff have the necessary skill sets?
- What's the total cost of ownership?
- How soon do you need new device or OS features?

Acknowledgments



I would like to express my special gratitude to all the students of Master 1 at the University of Yaounde 1 who contributed to the amelioration of this support. Specially:

