

Mobile Application Development

Produced
by

David Drohan (ddrohan@wit.ie)

Department of Computing & Mathematics
Waterford Institute of Technology

<http://www.wit.ie>
<http://ddrohan.github.io>
<http://ddrohan.gitbooks.io>



Waterford Institute of Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE





Introducing Android





Agenda

- ❑ Background
- ❑ Android History & Versions
- ❑ Setting up the Environment

Background (1)



- ❑ Android is a comprehensive open source platform designed for mobile devices.
- ❑ It is championed by Google and owned by Open Handset Alliance.
- ❑ The goal of the alliance is to “accelerate innovation in mobile and offer consumers a richer, less expensive, and better mobile experience.” (November 2007)



Background (2)

- ❑ Android, along with IOS, is revolutionising the mobile space.
- ❑ Unlike IOS, Android is an open platform that separates the hardware from the software that runs on it.
- ❑ This allows for a much larger number of devices to run the same applications and creates a much richer ecosystem for developers and consumers.



ANDROID WEAR



PHONES



TABLETS



ANDROID TV

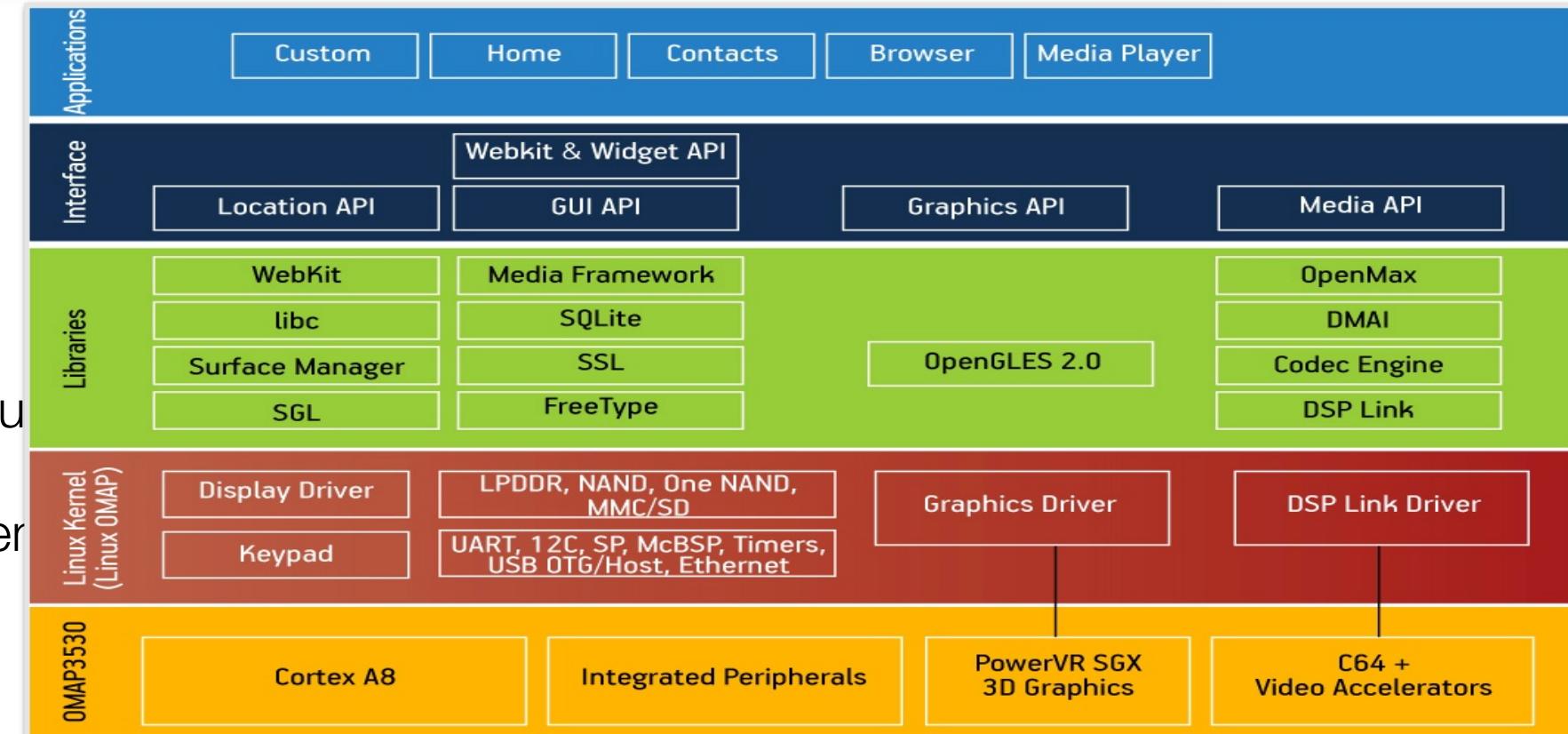


ANDROID AUTO



Comprehensive

- ❑ Android is a comprehensive platform, which means it is a complete software stack for a mobile device.
- ❑ The Android SDK is all you need to start developing for Android; you don't ever need a physical device.



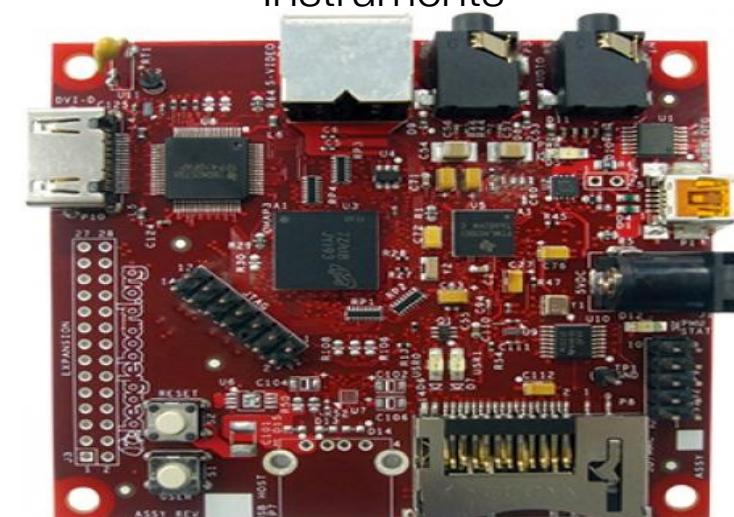
- ❑ Users can customize their phone experience substantially.
- ❑ Manufacturers can also customise the platform in substantial ways - even generating complete 'forks' of the original project (Amazon).



Open Source

- ❑ Android is an open source platform.
- ❑ Aside from the Linux kernel itself, Android is licensed under business-friendly licenses (Apache/MIT/BSD) so that others can freely extend it and use it for variety of purposes.
- ❑ Manufacturers can port Android OS to specific hardware. with minimal legal issues.
- ❑ Android has many hooks at various levels of the platform, allowing anyone to extend it in unforeseen ways.

BeagleBoard, a low-power open-source hardware single-board computer produced by Texas Instruments



Designed for Mobile Devices

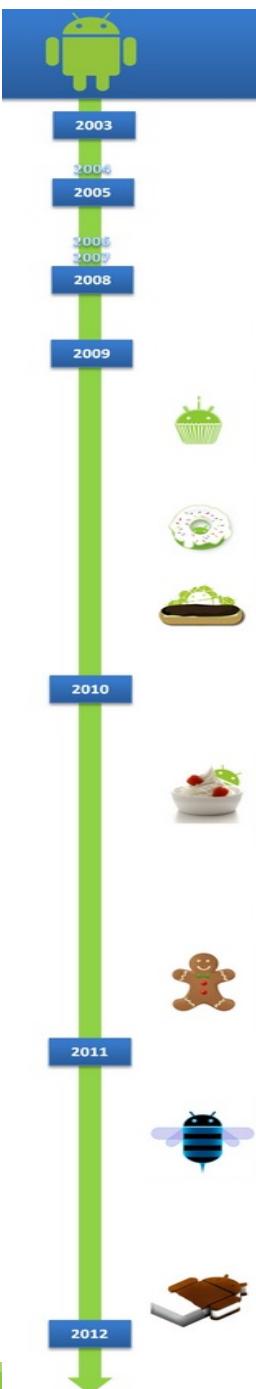
- When designing Android, the team looked at which mobile device constraints were likely not going to change for the foreseeable future:

- *Battery power, and battery performance* is probably not going to get much better anytime soon.
- In general, small size of mobile devices means that they will always be limited in terms of memory and speed.
- However, a device's screen size, resolution, chipset may vary considerably

- These constraints have been taken into consideration throughout the platform.



Android History



- In 2005, Google buys Android, Inc.
- In 2007, the Open Handset Alliance is announced. Android is officially open sourced.
- In 2008, the Android SDK 1.0 is released. The G1 phone, manufactured by HTC and sold by the wireless carrier T-Mobile USA, follows shortly afterward.
- 2009 sees a proliferation of Android-based devices. New versions of the operating system are released: Cupcake (1.5), Donut (1.6), and Eclair (2.0 and 2.1). More than 20 devices run Android.
- In 2010, Android is second only to BlackBerry as the best-selling smart phone platform. Froyo (Android 2.2) is released and so are more than 60 devices that run it.
- In 2011, Android is the #1 mobile platform by number of new activations and number of devices sold. The battle for dominating the tablet market is on.
- In 2012, GoogleTV, powered by Android and running on Intel x86 chips, is released. Android is now running on everything from the smallest of screens to the largest of TVs.
- In 2013, Google Glass, a wearable computing platform with an optical head-mounted display powered by Android is released to a select few.
- Beyond phones, tablets, and TVs, Android continues to be the big challenger to Embedded Linux as the platform for developing a number of specialized devices, such as home automation systems, car dashboards and navigation systems, as well as NASA satellites.



Versions – So Far



Cupcake
Android 1.5



Donut
Android 1.6



Eclair
Android 2.0/2.1



Froyo
Android 2.2.x



Gingerbread
Android 2.3.x



Honeycomb
Android 3.x



Ice Cream Sandwich
Android 4.0.x



Jelly Bean
Android 4.1.x



KitKat
Android 4.4.x



Lollipop
Android 5.0



Marshmallow
Android 6.0



Nougat
Android 7.0

Versions – API Level

- ❑ The Android version number itself partly tells the story of the software platform's major and minor releases. What is most important is the API level. Version numbers change all the time, sometimes because the APIs have changed, and other times because of minor bug fixes or performance improvements.
- ❑ As an application developer, you will want to make sure you know which API level your application is targeting in order to run. That API level will determine which devices can and cannot run your application.

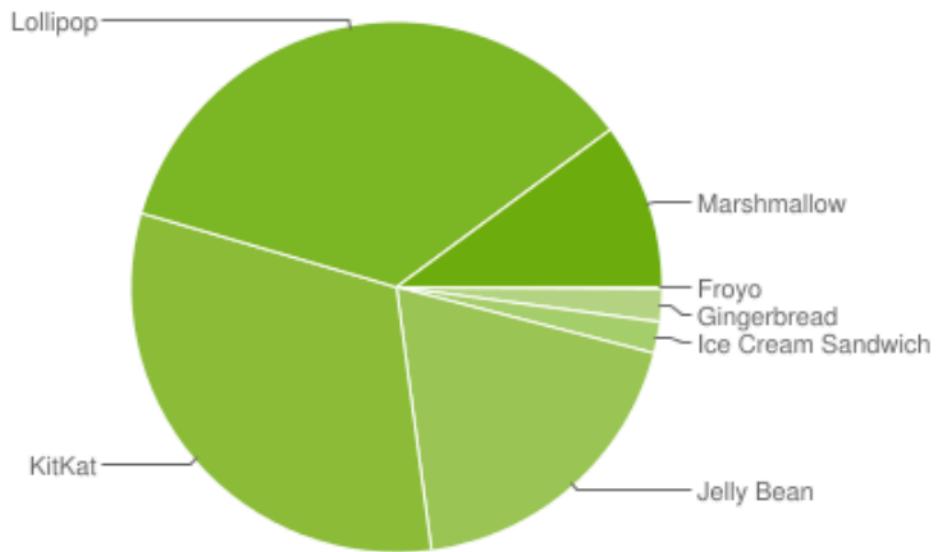
Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	2.0%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.9%
4.1.x	Jelly Bean	16	6.8%
4.2.x		17	9.4%
4.3		18	2.7%
4.4	KitKat	19	31.6%
5.0	Lollipop	21	15.4%
5.1		22	20.0%
6.0	Marshmallow	23	10.1%

*Data collected during a 7-day period ending on June 6, 2016.
Any versions with less than 0.1% distribution are not shown.*



Versions - Platforms

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	2.0%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.9%
4.1.x	Jelly Bean	16	6.8%
4.2.x		17	9.4%
4.3		18	2.7%
4.4	KitKat	19	31.6%
5.0	Lollipop	21	15.4%
5.1		22	20.0%
6.0	Marshmallow	23	10.1%





Target Version

- ❑ A developer's objective may be to have an application run on as many devices as possible.
 - shoot for the lowest API level possible. Keep in mind the distribution of Android versions on real devices out there.
- ❑ You may notice that there are a lot of users of Android 4.4 & 5.x. This places these few versions as the largest versions currently in the wild.
- ❑ This hasn't always been the case because OEMs tended to be very slow in upgrading their OS versions. However, this has changed with Google's strong push to get everyone onto the latest versions.
- ❑ Unfortunately, there are still a 'not insignificant' number of people who have anything from the 2.3.3 version to the 4.1 version because they have yet to upgrade their phones to a phone with the hardware capable of handling the newer versions.



Latest Version 7.0 – Android Nuguat

- ❑ Provides a lot of major improvements and refinements over Marshmallow (6.0), along with quite a few nice new usability changes.
- ❑ Multi-tasking and Split Screen functionality.
- ❑ Bundled Notifications & Quick Reply
- ❑ API Level : 24
- ❑ ART is still the default runtime



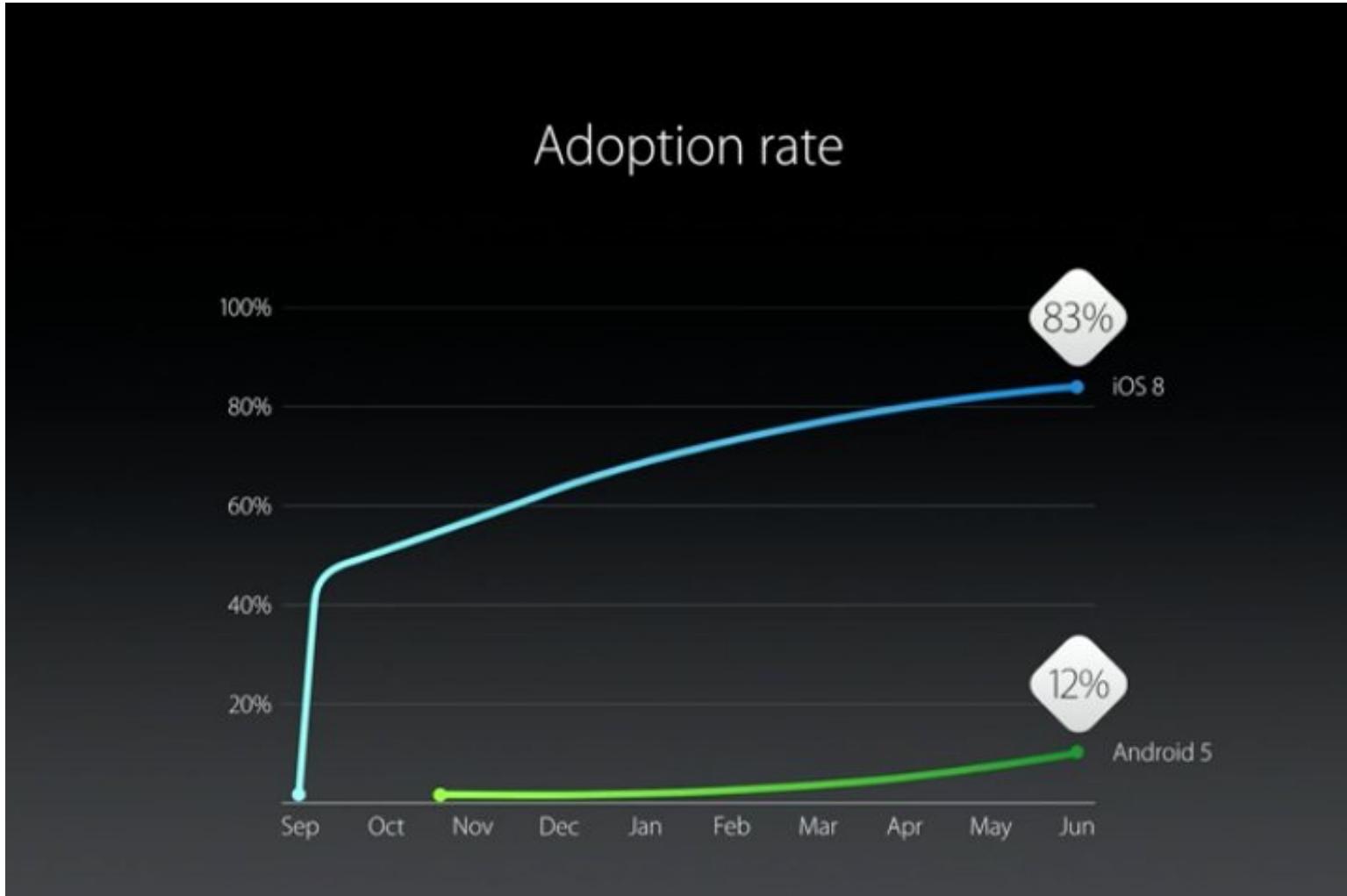


Adoption Rates

- ❑ Android adoption traditionally lags behind Apple's because Apple can make its latest iOS available for all users at once because it makes all the hardware and the software.
- ❑ Android, on the other hand, is much more fragmented. Because Google lets many different hardware makers use its Android software, it can't control when all phones update to the latest software. So, the newest version of Android is always made available for Google's lineup of Nexus smartphones and tablets, but it can't release software updates to every single Android phone at once.
- ❑ Smartphone manufacturers often add their own modifications to Android, which means that the update needs to be approved by each individual carrier first.
- ❑ This means that depending on which phone you own, it could take ages to receive the latest software update.
- ❑ Here's the brutal chart:



Adoption Rates



<http://uk.businessinsider.com/ios-vs-android-adoption-rate-2015-6?r=US&IR=T>



Android Applications (re : the installation stuff)

- ❑ Android applications get distributed in a .apk file
- ❑ APK == “Android Package”
 - It is simply a compressed file that has a particular file structure (similar to JAR files that take snapshots of the file system)
 - An APK contains
 - ◆ The Android Manifest file (an XML file with lots of metadata)
 - ◆ A Resource bundle containing sounds, graphics, etc.
 - ◆ The Dalvik classes / ART bytecodes that make up your application



What You need to Get started

- ❑ The Android SDK supports several different Integrated Development Environments (IDEs). Here we will focus on Android Studio because it is supported by Google, with the SDK and it's free (*and* Google aren't supporting Eclipse anymore!)
- ❑ No matter which operating system you are using, you will need essentially the same set of tools:
 - IDE
 - Sun's / ORACLE's Java Development Kit (JDK)
 - The Android Software Developer's Kit (SDK)
 - (if you're working with Eclipse) The Android Developer Tool (ADT) plugin, a special Eclipse plug-in
 - Phone driver (for specific device development)
- ❑ Android offer an “ADT Bundle” with everything you need
<http://developer.android.com/sdk/index.html> just confirm your SDK with mine (next slide) re solutions.....



Android SDK Setup for this Module

Android SDK Manager

SDK Path: /Users/ddrohan/env/appdev/android

Packages

Name	API	Rev.	Status
Tools			
Android SDK Tools	23.0.5		Installed
Android SDK Platform-tools	21		Installed
Android SDK Build-tools	20		Installed
Android SDK Build-tools	19.0.2		Installed
Android 5.0 (API 21)			
Documentation for Android SDK	21	1	Installed
SDK Platform	21	1	Installed
Android TV ARM EABI v7a System Image	21	1	Installed
Android TV Intel x86 Atom System Image	21	1	Installed
ARM EABI v7a System Image	21	1	Installed
Intel x86 Atom_64 System Image	21	1	Installed
Intel x86 Atom System Image	21	1	Installed
Google APIs	21	1	Installed
Google APIs ARM EABI v7a System Image	21	2	Installed
Google APIs Intel x86 Atom_64 System Image	21	2	Installed
Google APIs Intel x86 Atom System Image	21	2	Installed
Sources for Android SDK	21	1	Installed
Android 4.4.2 (API 19)			
SDK Platform	19	4	Installed
Samples for SDK	19	6	Installed
ARM EABI v7a System Image	19	2	Installed
Intel x86 Atom System Image	19	2	Installed
Google APIs (x86 System Image)	19	9	Installed
Google APIs (ARM System Image)	19	9	Installed
Google APIs ARM EABI v7a System Image	19	2	Installed
Google APIs Intel x86 Atom_64 System Image	19	2	Installed
Google APIs Intel x86 Atom System Image	19	2	Installed
Sources for Android SDK	19	2	Installed
Extras			
Android Support Library	21.0.1		Installed
Google Play services for Froyo	12		Installed
Google Play services	21		Installed
Google Play APK Expansion Library	3		Installed

Show: Updates/New Installed Select [New or Updates](#)

Obsolete [Deselect All](#)

[Install packages...](#) [Delete packages...](#)

Done loading packages.



References

❑ Books (in rough order of preference)

- Android Developer's Cookbook (Steele & To)
- Busy Coder's Guide to Android Development (Murphy)
 - ◆ Online only: <http://commonsware.com/Android/>
- Android in Action, 2nd Edition (Ableson, Sen, & King)
- Android Application Development for Dummies (Felker)

❑ Online references

- <http://developer.android.com/>
 - ◆ By far the most important single reference.
- Android forum on StackOverflow
 - ◆ <http://stackoverflow.com/questions/tagged/android>
- Android widget gallery
 - ◆ <http://www.droiddraw.org/widgetguide.html>
- Marty Hall's Tutorials
 - ◆ <http://www.coreservlets.com/android-tutorial/>



Questions?



Appendix



Android Apps vs iPhone Apps



Installing Apps

- ❑ For Generic (on the market) apps
 - iPhone has larger selection
 - Android catching up
- ❑ For In-house-developed corporate apps
 - iPhone apps can only (mostly) be installed via the App Store
 - ◆ iPhone requires you to submit app to the Apple App Store and get approval, even for apps from your own company
 - Unless you setup a Provisioning profile or
 - you use something like TestFlight or
 - jailbreak your phone of course.....
 - Android apps can be installed through
 - ◆ Google App Store / Google Play
 - ◆ Amazon App Store
 - ◆ **USB connection from PC**
 - ◆ **Email**
 - ◆ **Corporate Web site**



Languages for Apps

❑ iPhone

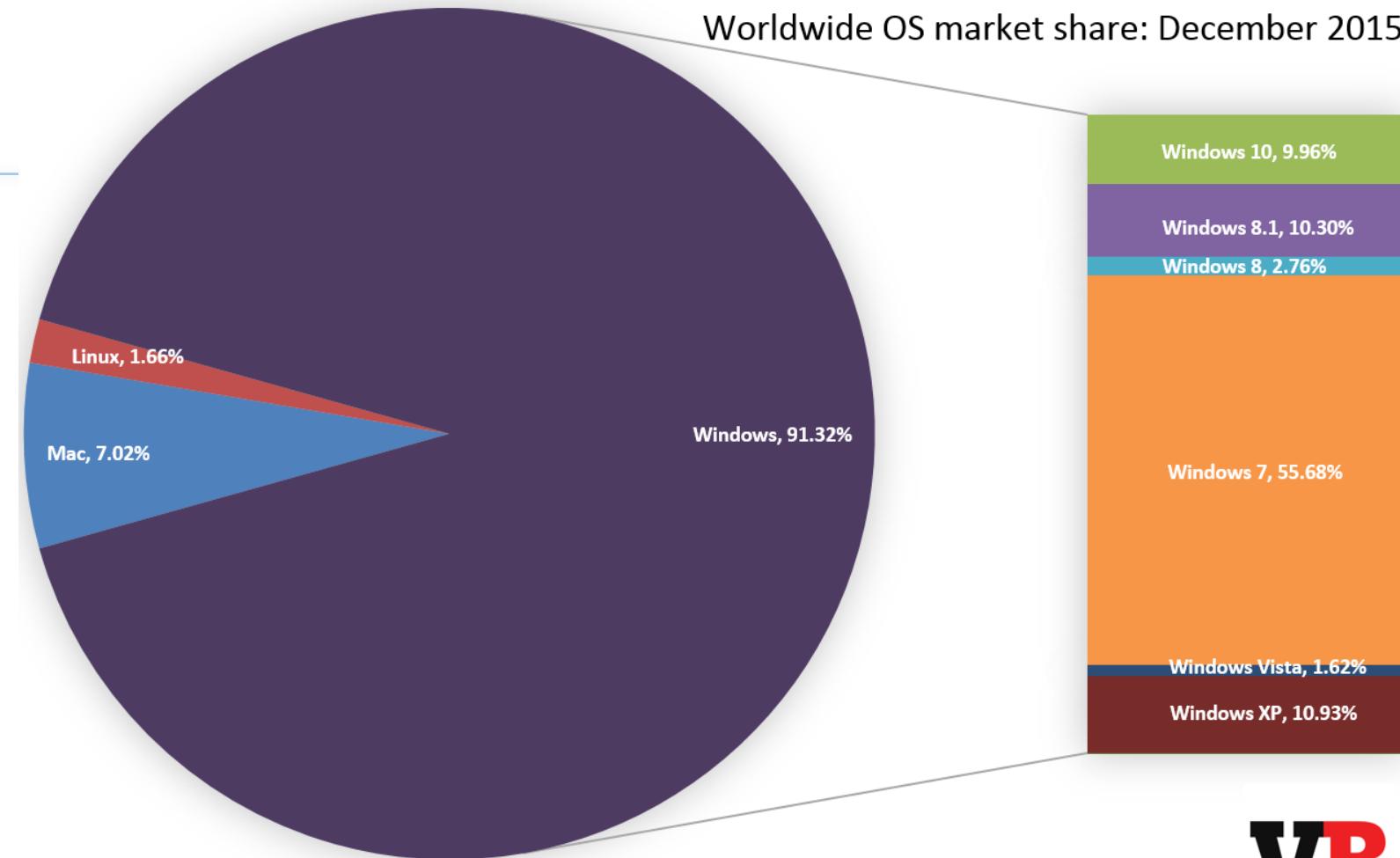
- Objective-C
 - ◆ Similar to, but not exactly the same as, C++
 - ◆ Virtually no corporate presence for Objective-C, other than for mobile apps
 - ◆ Swift starting to gain adoption

❑ Android

- Java
 - ◆ The single most widely used language inside corporations
- C/C++
 - ◆ Can call native apps (with some difficulty) via an approach similar to JNI for desktop Java

OS for DevApps

- iPhone
 - Macs
- Android
 - Anything with Java and Eclipse
 - ◆ Macs
 - ◆ PCs
 - ◆ Linux
 - ◆ Solaris

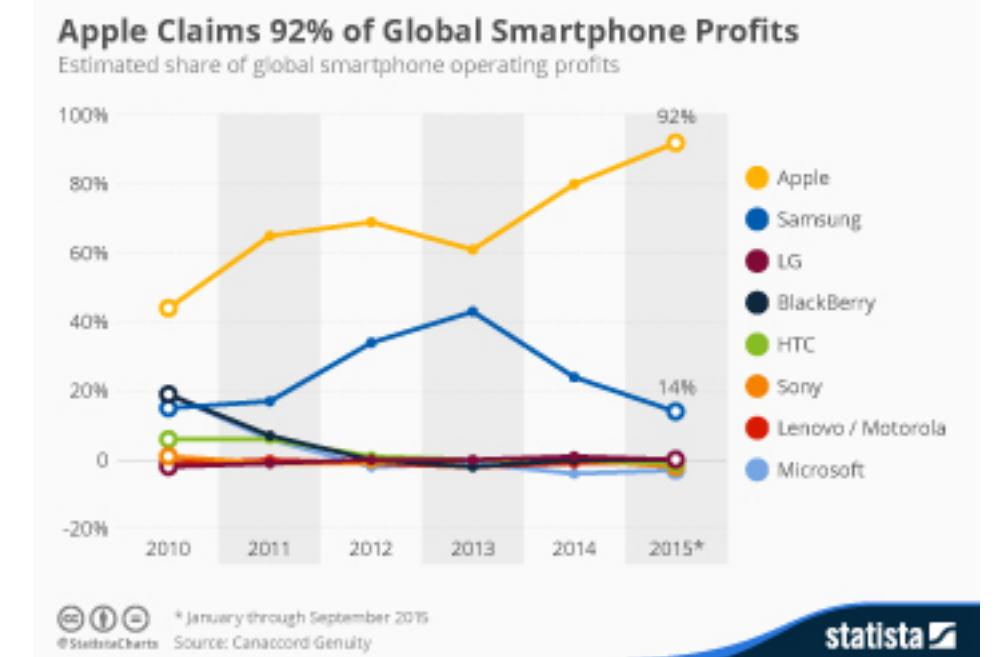
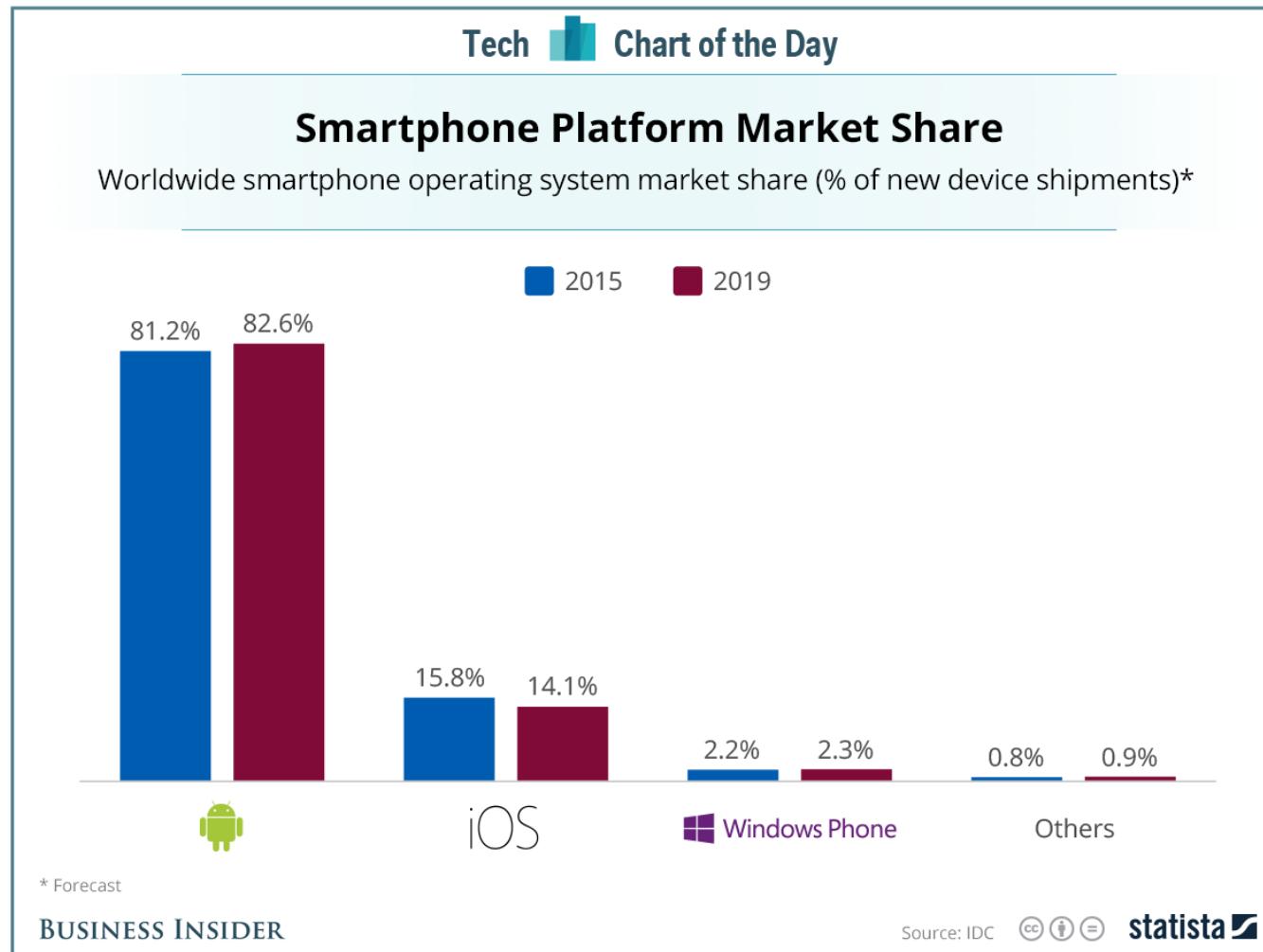


VB

- Issue
 - Not so much which is cooler and which you personally prefer, but rather which is **already** installed in corporate environments.



Market Share Vs Operating Profits





Other Issues

- ❑ Phone features, quality of apps, **customer loyalty** and coolness factors
 - Matter of opinion, but iPhone **very** strong here
- ❑ Market presence based on sales / app downloads etc.



Bottom Line: Android vs iPhone

❑ Which to use personally

- iPhone has bigger app store, and more loyal users
- Android more open and growing more rapidly
- Bottom line: no clear winner, personal preferences prevail, but iPhone has the edge (for me!)

❑ Which to use for in-house apps

- iPhone apps very hard/restrictive to install, Android very simple
- iPhone uses mainly Objective-C (Swift on the up), Android uses Java
- Bottom line: Android is clear winner 😊 IMHO