

CHAPTER 4



Android Development— Business Overview and Considerations

Data is a precious thing and will last longer than the systems themselves.

—Tim Berners-Lee

The Android Industry has its own requirements for entry. It's helpful to have a full understanding of these requirements and specifications when you're deciding whether to pursue a business venture. As discussed in previous chapters, the Android OS is already the industry leader for mobile devices, and its market share is growing with each passing year. With the introduction of x86 to the Android software stack, applications written for x86 systems are now supported, which expands the market even further.

The Android Market Share

According to International Data Corporation (IDC), Android's second quarter 2013 sales have reached 187 million units shipped. The previous record—100 million units sold—was broken by Android in 2012. These numbers bring Android up to 79 percent market share of smartphone operating systems. Compared to last year's 136 million units, Android has grown by over 74 percent. Much of Android's growth can be counted from Samsung's success in the last quarter, with over 39 percent of the smartphone manufacturing market share. IDC's senior analyst Kevin Restivo had this to say about Android's success:

The share decline of smartphone operating systems not named iOS since Android's introduction isn't a coincidence. The smartphone operating system isn't an isolated product, it's a crucial part of a larger technology ecosystem. Google has a thriving, multifaceted product portfolio. Many of its competitors, with weaker tie-ins to the mobile OS, do not. This factor and others have led to loss of share for competitors with few exceptions.

The future of Android's market share is a much more debated topic. Some industry experts have made claims that this is the peak of Android's dominance. Experts have suggested that with the release of new iPhone models, and new iOS updates, Apple may steal the market share back from Google over the coming years. Others claim that this is just the start, best stated by John Koetsier:

Android is a train that has left the station, and it is stopping for no one. The number of Android phones sold in this quarter alone is greater than the total number of smartphones of all kinds sold in the entire year of 2007.

It is hard to say what the future of the smartphone operating system market looks like. No one can say for sure, but how does success in market share translate of profits? With Android being a free and open source platform, how is Google making money from it?

How Android Makes Money

Google is a giant successful company; there is no denying it. But how does Google make money from the Android project? Apple charges not only for their operating system, but also for upgrades and applications. Google's strategy is in line with their primary company revenue streams, *advertisement*.

Google has various avenues for collecting money from Android and its software stack. To start with, Google collects money from Android through advertising in the browser, and in its Google Play store. This advertising avenue may seem small; however, this is where the majority of Google's money is made. Google also takes royalties from applications on Google Play, and charges for adding content to the Google Play store.

The industry leader makes billions of dollars in revenue each year from advertising on Google sites, and on networked "partner" sites through AdSense. In Q3 of 2012, Google's total advertising revenue was over 14 billion dollars. Larry Page, Google's CEO, put Google's mobile business on an 8 billion dollar run rate. This has grown by almost three times the amount since last year's 2.5 billion dollar run rate. Page did not further break down what sections this revenue was coming from; he only commented on the growth saying, "users paying for content and apps in Google Play." Google's SVP and CFO Patrick Pichette commented on the growth saying, "clearly we don't break down the categories. Ads continue to be the bulk of the 8 billion—the vast majority of it."

Why Android Is Successful

What is it exactly that makes Android so successful and commercially viable? Some experts claim that it's the powerful financial companies that sit behind its products; others claim that it's the backing of the open source community and the free sharing mindset; and yet others claim it's the features and the extensive development capabilities. It is hard to find a single reason, but much easier to point out the areas of success.

Free

Android has always been, and will always be, free. This enables massive bonuses for all parties involved. Manufacturers can build hardware with confidence that the Android stack will be free, developers can write applications without needing to worry about complex platform costs, and researchers can find flaws and improve the underlying systems without needing to purchase licenses and agreements.

Open Source

The open source community is a gigantic collection of developers and likeminded individuals who believe that sharing information can lead to stronger and better products. Android is a child of this community, and has been open sourced thanks to Google. This allows programmers from all over the world to contribute and extend the software that they use every day.

Customization

The Android software stack allows you to personalize your device to the point that you feel comfortable. You can change the colors, the way your phone responds to notifications, remove and add applications, and even change the way you secure your device. This level of customization allows users to enjoy and utilize the device to its fullest potential.

Application Base

The Google Play store has over 1 million Android applications available at various prices. These applications can use the device in any way a developer can imagine. In comparison to iOS's App Store, the Google Play store consists of predominantly free applications. This is a primary reason the Apple App Store generates higher revenue. With this in mind, the Google Play store outpaced the Apple App Store for number of apps downloaded in the second quarter of 2013. Developers can earn revenue from advertising inside of the applications, which allows consumers to use the applications free of cost.

Hardware Choices

Unlike the other mobile operating system choices, Android has the largest selection of devices to choose from. There are over 3,900 Android devices that exist in the market right now. Consumers can find a device that has all of the features that they need. These choices also cover all of the major cellular service providers in the United States, and European countries, providing greater outreach.

Device Price

Android devices offer various price points for users. Consumers can choose how much they want to pay for an Android device based on the features that are important to them. Some third-party cellular service providers offer Android devices on pay-as-you-go plans.

For much higher prices and service contract packages, the latest and greatest Android devices are available. The more money you spend, the greater the hardware and features your device will contain.

Legacy and Future Platform Support

Android has been commercially available for five years, and has produced a multitude of software versions and hardware platforms. With the robustness of the platform, Android must make choices about how to deal with past versions, as well as future revisions. Backward compatibility is a problem that all large platforms face.

Legacy Support

Android, from version 1.5 and upward, is packaged onto hardware platforms, and exists in commercial markets. Software for Android is fully backward compatible. This means that applications built for version 1.5 can be used to their full extent on version 4.2 devices. This alleviates some concerns that developers have about losing market share in new devices. It is important to note, however, that this compatibility does not utilize the new features of Android. For the best quality and functionality, applications should be rewritten for new versions.

Android also has some interesting hardware requirements. In order to gain Google's Android stamp of approval, manufacturers must meet certain qualifications like having a cellular radio, GPS capabilities, and Wi-Fi chips. However, many of the things users take for granted are not mandated or regulated. This list includes screen size, screen resolution, internal storage size, GPU speed, and even processor specifications. This provides manufacturers the freedom to create high-powered devices, cheap consumer friendly devices, and anything in between.

Future Support

Google's strategy for Android's future releases is catered toward backward compatibility and engaging the largest market share. New versions of Android will have mechanisms for running applications developed years before. It is important to note that new applications cannot run on old systems. If you use the Android Software Development Kit version 4.1 to develop an application, that application cannot run on a 2.3 Android device. However, an application that is written for a 2.3 Android device will run without modification on a 4.1 device.

Why x86 and Android Are Right for You

Android and the x86 family is a business with low barriers to entry and significant probability of success. These low entry requirements give even the smallest companies the opportunity to be successful, mainly because startup costs are minimal. The question is not whether Android and x86 are right for you, but rather what steps you need to take to be successful.

Cross Compatibility

The x86 architecture offers a diverse network of systems deployed in many industries. From cash registers, to televisions, to mobile devices, and even major utility control systems, x86 platforms are everywhere. Few industries in the world exist that don't have the x86 architecture incorporated in some manner. With the combination of the Android platform, the outreach is even larger.

Applications written for Android will run on all Android devices that are the same version, regardless of the underlying processor. This means that applications written and tested on ARM Android devices require little to no effort to be used to their full capabilities on Intel x86 Android devices. The exception to this is applications that use Android's NDK. However, with a simple recompile, the application should be up and running. For more information about software migration, refer to **Chapter 7: Creating and Porting NDK-based Android Applications**.

Barrier to Entry

Economically speaking, to use Joe S. Bain's classic definition, "a *barrier to entry* is an advantage of established sellers in an industry over potential entrant sellers, which is reflected in the extent to which established sellers can persistently raise their prices above competitive levels without attracting new firms to enter the industry."

Barrier to entry is the reason why many companies can aggressively dominate certain industries. For example, the oil industry has an incredibly high barrier to entry. The cost of starting a successful oil company is outrageously high, since you are required to have so many resources and tools to compete with the existing leaders.

The Android industry has a very low barrier to entry. The costs and requirements to create a successful business in the Android space are much lower than most other technical industries. The largest challenges involve finding a product idea and building a development team. With a strong idea and a foundational team, success is only a matter of development and marketing. Indeed the costs are low enough that even individuals can succeed in this marketplace.

Security of Android

Security, when it comes to software systems, is often of much deeper concern to the provider than the customer. A good security system can be explained in depth, without being compromised. Android is one of these systems. The security that surrounds the platform and its components is well documented and researched. For more information and technical details about the system, consult *Android on x86 Security Guide*.

Application Security

With the introduction of an application market, Android's security model has grown incredibly complex. Android must secure its own applications and also provide some level of security for third-party applications. Its security system must be simple enough that the average user can understand the applications, and it must allow the users to decide whether or not to use them.

Android's solution to the problem was to use *permissions*. In order to access certain functions of a device, you must register for the related permission. For example, to use data or Wi-Fi services in the application, you must register to use the Internet permission. When the application is displayed to the users, the different permissions are displayed as well. If an application attempts to use a feature without registering for the permission, the application simply crashes.

Another major section of application security is the separation of information between applications. If an application can freely interact with other applications on the phone, malicious things can occur. There are various situations when users need applications to be able to send messages to each other. Android's inner-application messaging system uses the concept of *intents* to relay information across the operating system.

An intent is simply a free-form message produced by an application and handed to Android. These messages can have various types of data, and come in two primary flavors. An *implicit intent* is a message that exists for any application that can access it. For example, on many Android applications, when you click a link, a window appears, letting you choose from all several applications that can view the site. In the background, the application fired an implicit intent that was then relayed to all available applications. *Explicit intents*, on the other hand, are directed at one very specific application. This intent can be seen and handled only by the application for which it was crafted.

Third-party applications also operate as separate users on the underlying operating system. This means that third-party applications cannot access files and resources that another application owns. The exception to this is with system applications. System applications can access all sections of the device that are required for operation.

Platform Security

The security surrounding the phone and its onboard features is a widely discussed topic. Android provides many different security features to help secure users and their data. Some of these features include screen locks, text and e-mail encryption, multiple types of passwords, and extra password prompts when you access certain sections of the device. These features verify authenticated users.

Third-party applications have also been created to aid this effort. Applications exist that can help you find your device if it is stolen, remove data from your phone, lock the device remotely, and add even more customization for the way you authenticate with the device.

Licensing

In the software development industry, licensing fees are very common. From library licensing costs, to platform licensing costs, and even device licensing costs, there are plenty of situations where you'll need to spend a little money to develop and sell products. Since Android is an open source and developer-friendly community, it aims to keep these costs low.

Android Licensing Cost

There is none. Android is open sourced under the Apache 2.0 license, which permits commercial usage, modification, and distribution absolutely free of any cost. This means that anyone can play with the Android operating system's source code, and create a whole new product out of it.

Application Licensing Cost

Creating applications for Android is a slightly more involved process. The development suite and software development kit are free to anyone who wants to download them. With these tools you can build, test, and deploy to development devices, any Android application from the source. Releasing your Android application is a different story.

Android devices can install third-party applications via USB and from the SD card, provided the option is enabled, but these mediums are not very convenient for consumers. Enter the Google Play Store, an Android application market where developers can quickly and easily upload and update applications for the entire Android population to see and purchase.

The process of selling applications in the Google Play Store is straight-forward. The first step is setting up a Google account. This gives you access to the Android development site, where you can manage your applications. Once you create your account, Google requires a one-time \$25 fee to distribute applications inside of the Google Play market. You can then upload and configure your applications to distribute in the market.

Now that you have an application you want to distribute, how do you earn money, and what does that look like? To collect money from your Android application sales, you need a Google Checkout Merchant Account. Purchases of your application in the Google Play store will contribute to the your Google Wallet, which can be transferred to USD using various methods. Google does take a percentage of all your profits, based on many factors such as application sales and rate of sales.

Physical Development Costs

When you're starting off in the software business, it is important to remember that there are physical requirements for the virtual product you'll create. Unlike in traditional engineering, the thing that is being sold won't necessarily have a physical component. To create and fully test an Android application, there are a couple of physical platforms that are required.

Software Development Systems

Developers use software development systems to write the code that will run on the Android devices. The Android Software Development Kit (SDK) can run on Windows, Linux, and Mac, so choosing an operating system is up to the developer. When it comes to hardware choices, there are a couple of things to consider.

The Android SDK is a relatively large application, especially after it downloads the files needed to run the different versions of Android. Getting a hard drive large enough to hold all of the required tools and files is a must, but hard drive speed is also a serious concern. With the price of solid state hard drives dropping, it's worth considering one for development. A solid state hard drive operates at read and write speeds often more than double the speeds of a traditional hard drive. This translates to significant time savings—the system will boot faster, Integrated Development Environments (IDEs) will operate more quickly, and the applications will run faster.

If you plan to run the Android Emulator on a laptop, it is also worth investing in a significant amount of RAM for the laptop. Anywhere from 4GB and higher should be sufficient. If you have less than 2GB, you will have problems. With more memory on the system, more applications can be run at the same time.

Finally, you need a processor powerful enough to run the SDK and the required development tools. The choice is up to your development team; however, the newer the better. Something with multiple cores is always a plus.

Android Testing Systems

The Android Emulator is capable of nearly all the functionality provided by physical Android devices. Even so, it's paramount that testing occur on true hardware. The real hardware system will respond exactly as customers can expect, so testing on true hardware allows for a much more natural experience.

A single Android device will not be enough. It is important to test on multiple devices of the same Android version as your target version. If the application is being developed for a previous version of Android, it is imperative to test on devices that have newer versions as well. The greater variety of devices you use to test software and confirm usability, the more bugs and problems you'll identify pre-launch. Finding and fixing problems before a customer deals with them will help establish your product.

Overview

Android and x86 are a powerful combination. With a low barrier to entry, a strong security backing, low licensing costs, and an easy mechanism to deploy applications, Android is an easy and profitable industry to get involved with. Thanks to Android's success and Google's profits, the Android stack is something that has an ever-growing backing. Android and x86 are the right choice for development. With careful consideration for the development team and the product, a viable strategy, and successful management, there is much money to be made from the Android ecosystem.