

# The Android Market: A Background

Whenever anyone writes a book, he or she always has to be mindful of the audience. If you are an experienced Android programmer, the technical portions of this book may seem quite basic. If that is the case, we apologize in advance. We decided that because the book is about the *business* of Android apps development, some of our readers might not have any Android programming experience at all.

If you have no experience with Android, we'll try to point you in the right direction. Everyone starts with no experience at one point or another, and with technology this happens all the time. It is hard to believe that the entire idea of using a mobile “app” is less than a decade old at this writing. Ten years ago, if you were to talk about an “app” in that sense, people wouldn't be sure what you meant.

## The Smartphone Revolution

If you think about all the things you do on a normal day, whether it is checking your e-mail while riding on the bus, surfing the Internet while waiting for another appointment, or running the latest application, you'll probably agree that smartphones are part of our daily life. We're sure that some of us who are Facebook and Twitter junkies wonder how in the world we lived our lives before smartphones. The technology is now something that we really take for granted, especially because most of the technology of cell phones as computers is quite new and constantly changing.

Historically speaking, the computer is a relatively new invention. The computer industry, now a giant in all types of business, is less than a century old. The Turing COLOSSUS, which was the earliest general-purpose computer developed during World War II, was designed to run aerodynamics calculations. The discoveries of Bell Telephone's transistor in 1947 and the integrated circuit, developed by Texas Instruments and Fairchild Semiconductor in 1969, helped computers make great strides during the 1950s and 1960s. Soon the IBM System/360 became the standard institutional mainframe computer. Intel co-founder Gordon Moore famously predicted that the

number of components in an integrated circuit would double approximately every two years. Moore's prediction has stood the test of time, and his simple statement has since been known as Moore's Law. Because integrated circuits could do more computations for the user with less real estate, computers became smaller and smaller. The Z3 was a massive machine that took up nearly an entire building. The processing power of this gargantuan computer is minuscule compared to what we have today on the simplest of smartphones.

Through decades of advances in technology, computers became small enough to fit on a desk at home or in the office. Soon the power of the desktop transitioned to the laptop, and computers became lighter and flatter, easily transitioning from the desk to the Wi-Fi hotspot at the local coffee shop.

As computer technology improved, so did that of cellular phones, leading to a mobile phone revolution. Cellular phones were once a toy that only the rich could afford because they often cost a few thousand dollars. There were phones such as the 1983 Motorola DynaTAC 8000x, and they were a burden on their users because they weighed almost two pounds. There was a period when the weight of cell phones worked against them, but they were very popular to have in the car during the '90s. Fortunately, phones such as the 1989 Motorola MicroTAC 9800X were light enough to fit in the jacket pocket, and others, such as the Motorola StarTAC, became popular with their clamshell design.

The next logical step was to put more features on a cell phone than just phone and texting, and they soon became "smarter." Now all the power of being connected to the Internet was in the palm of one's hand. Ericsson was the first to call its phone a "smartphone," and the Nokia 9000 Communicator had similar features and was driven by an Intel 386 CPU, the same CPU previously used in Intel desktop computers.

Most tech enthusiasts remember when Steve Jobs unveiled the iPhone, a smartphone designed with the consumer in mind. We ask a lot of our tech friends, "Where were you when the iPhone was announced?" Tech enthusiasts remember when Jobs brought out his new toy, and how its one-button goodness with its "apps" changed the mobile world forever.

The rapid development of smartphone technology in the last decade is explained by Koomey's Law (somewhat similar to Moore's Law). Dr. Jonathan Koomey of Stanford University has shown that the need for electrical power (battery capacity) halves every 1.6 years. This means that computers don't just get faster (owing to greater transistor counts) but they also get smaller and more portable! Because the battery takes up less and less room on a smartphone, smartphones can pack a lot of computing power into the remaining volume.

But the computing power is only half the equation. The other half is connectivity. Smartphones are almost always online. Always-on connectivity creates amazing possibilities. The app store, a staple of all modern smartphones, is a direct outgrowth of always-on connectivity.

Now we are in an age when we can do just about anything with our phones, thanks to the app revolution. Just think about how businesses like Instagram have flourished with this new smartphone age, something that wasn't even possible a decade ago!

Every new technology creates new opportunities. As we mentioned before, the size of a computer has decreased. With this decrease in size has come a decrease in the price of computers overall. The same applies to smartphones, which are getting cheaper for the consumer thanks to contracted deals from carriers. Today, many users in countries that can't afford desktop or laptop computers have access to smartphones, with mobile networks that take them to the World Wide Web and beyond.

Though we can't go to every place in the world and get a signal, that's changing fast. In fact, even today one in three people on Earth have Internet access, and many of them have access through cellular networks.

According to the latest research from Strategy Analytics, the number of smartphones passed one billion in the third quarter of 2012. This is a mere 16 years after the first smartphones hit the market. Very few inventions have swept the globe so quickly.

This is great news for app developers. Smartphones have already changed the way we work and play, but we're sure there are many undiscovered ideas that clever app developers will unleash upon the world.

And the good news for you, dear reader, is that Android is by far the most popular smartphone operating system. In fact, in the last quarter of 2012, 70 percent of smartphone shipments were Android phones!

## The Beginning of Android

A lot of people hail the iPhone as the first smartphone, but as we mentioned before, it was not. It was unique in its iOS operating system, and it may seem as if the Android operating system is a mere imitation. However, work leading to the Android OS began long before the iPhone was released to the public in 2007. Andy Rubin, known as one of the founders of Android (later acquired by Google) had been working on smartphone designs since January, 2000. The company he founded prior to Android was called Danger, Inc., which released the Hiptop (also known as the T-Mobile Sidekick) in October, 2002, years before Apple released its first smartphone.

Andy Rubin, along with Rich Miner, Nick Sears, and Chris White, then started Android, Inc. in 2003. In Rubin's words, there was tremendous potential in developing "smarter mobile devices that are more aware of its owner's location and preferences" (<http://www.businessweek.com/stories/2005-08-16/google-buys-android-for-its-mobile-arsenal>). The company ran out of money, but it had developed an open source operating system for mobile phones by the time it was acquired by Google in 2005. Android worked rather discreetly on its mobile operating system for about two years.

Google helped start the Open Handset Alliance (OHA), which is a consortium of a lot of companies such as HTC, Motorola, Samsung, Sprint Nextel, T-Mobile, and other big names in the telecommunications industry. This group eventually unveiled the mobile operating system that we know today as Android. The first public beta of Android was released in November 2007, a mere five months after the iPhone first hit the market.

Android and iOS are currently the two major players in the world of mobile phone operating systems. Microsoft has only a fraction of the market with its Windows Phone 8 operating system, in spite of some successful flagship phones from Nokia such as the Lumia 920. The BlackBerry market was once significant, but according to comScore MobiLens, it accounts for less than six percent of the marketplace.

However, BlackBerry has recently released some new devices with a new operating system, and its fortunes might change. In a sign of how important Android has become, the new BlackBerry devices support a "Runtime for Android apps." This is a series of tools that allow you to easily repackage your existing Android apps to work on BlackBerry phones. We'll get into this in more detail later in the book, but rest assured, even if BlackBerry is wildly successful, you're making the right choice by developing for Android!

## Why Android?

Android is by far the world's most widely used mobile phone operating system. If you want to reach the most users with a single code base, Android is the way. As we just mentioned, you can even easily get your app in front of BlackBerry users!

Android is quite simply crushing iPhone in user adoption rates. According to Strategy Analytics, in 2012 more than 3.5 Android smartphones shipped for every iPhone. Android has shown quite a lot of growth. In 2010, 100,000 new devices were activated each day. In 2011, 500,000 new devices were activated each day. According to Google chairman Eric Schmidt, as of April 2013 there are more than 1.5 million new Android users *every day*!

Even better, Google Play, which once lagged behind the Apple App Store in revenue, is coming into its own. Google Play revenue grew by 90 percent in the first quarter of 2013 relative to the last quarter of 2012. In the same time period, Apple's App Store revenues grew by only 25 percent. Particularly in Asia, Google Play revenue growth rates are astounding. Japan has surpassed the United States in Google Play revenue! South Korea is also incredibly strong. At these growth rates, it seems like only a matter of time before the Google Play store becomes the dominant app store. In terms of the marketplace for apps, Google Play currently has more than 700,000 applications, which have been downloaded more than 25 *billion* times!

Most successful Android application stories are pretty well known. As an example, *Angry Birds* by Rovio is free for Android users, and it is a mobile game giant that has produced massive funds for the company. The mobile game had more than two million Android downloads in the three days after its Android release, and seven million Android downloads one month after that. Rovio, the game's developer, is still finding ways to make money on the *Angry Birds* franchise with spin-offs and even merchandise.

Of course, there are more successful applications than just the ones made for gaming. For example, Edward Kim, the author of the *Car Locator* application, was initially excited to be making \$20 per day. Within five months, he was making more than \$13,000 per month in sales.

You will soon discover that the massive quantity of Android applications on the market can work against the developer, as the Android market is flooded with applications of all types. It's not uncommon for 20,000 new apps to appear each month.

This means that one application, as great as it might be, can get "lost in the crowd" and become very difficult to be noticed by its intended audience. Android users might pay to download one type of application even though a free one with more features is readily available. All this because some of the better applications can't get noticed in the world of many, many Android applications.

Keep in mind, however, that the Apple App Store has roughly the same number of apps as Google Play, so iOS application developers have the same problems when it comes to standing out from the crowd. Remember, unlike Apple, Google built its brand on the strength of its search abilities. You can be sure that engineers at Google are hard at work figuring out the best ways to provide Google Play users with search capabilities that let them find exactly the app they're looking for.

In fact, Roy is happy to report that searching for his apps in Google results in first page listings (see Figure 1-1).

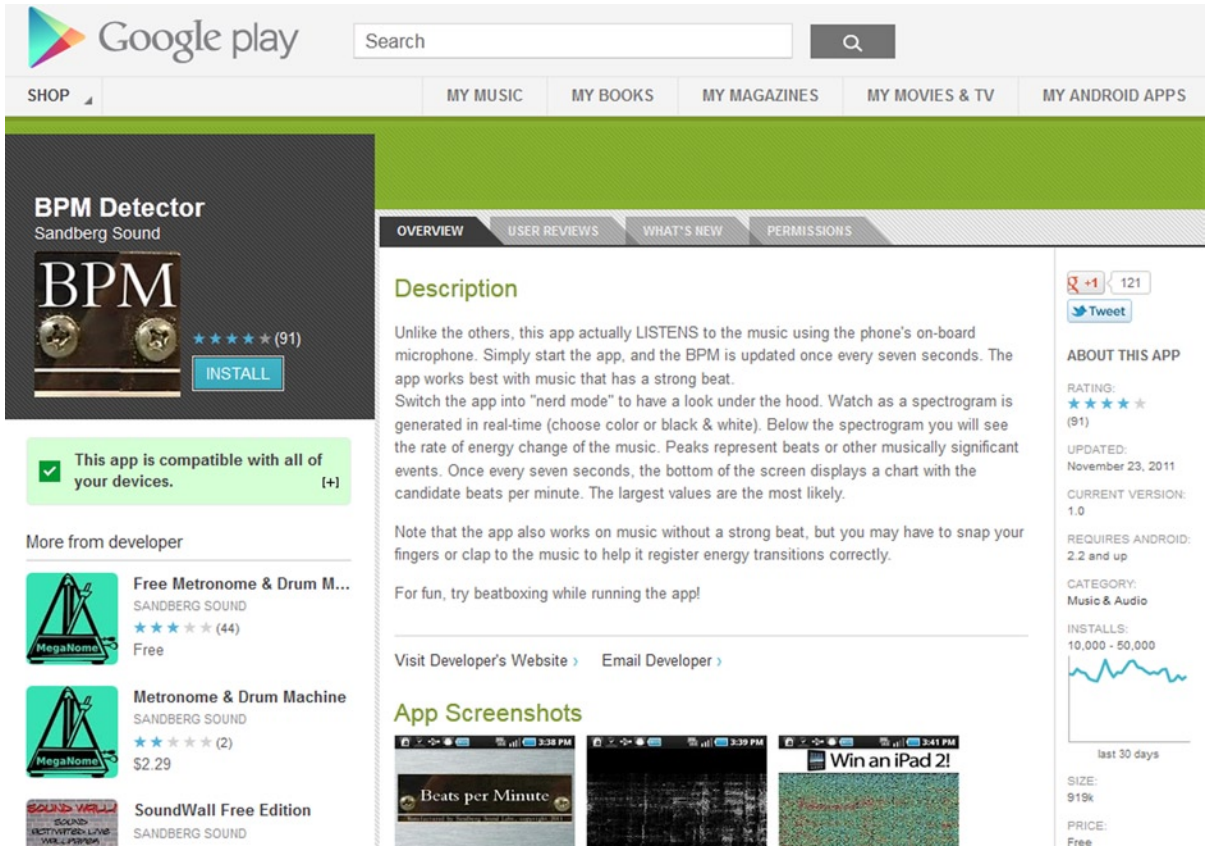


Figure 1-1. A screen shot of the Google Play Market that shows Roy Sandberg's app

## Android vs. iOS

When the iPhone was first unveiled, a new sort of business model was established for consumer electronics. Although Steve Jobs and his associates at Apple were not the first to invent the touchscreen, they were able to create a new type of software enterprise that was personable and utilitarian. Apple's "there's an app for that" slogan has promised users that the mobile software they need should be readily available where and when they need it. It works for the smartest engineer and the dumbest consumer, and it created a new type of software market. Historically, the Apple App Store has led the way by a wide margin in terms of number of apps and downloads. That, however, is set to change. The Google Play App Market has nearly reached parity with Apple, both in terms of the number of apps and the number of downloads. As of October, 2012, iOS had only 10 percent more downloads than Android.

Although iOS still has a sizable lead in terms of total revenue, that gap is also closing fast. If the trend lines continue at their current pace, Android could surpass iOS in total revenue by early 2014. In the near future, we can expect Android to take the lead in terms of total apps, app downloads, and app revenue. If we were betting types, we'd place our bets on Android!

## The Difference Between Android and iOS

As a developer, you should know how Android compares to iOS at least at some level.

Apple's iOS is a proprietary operating system, while Android is open source, which gives users the right to study, change, and improve the design through ready availability of the source code. Internally, Android uses the Linux kernel.

One of the great things about Android developer tools is they are free. This is one of the reasons why the operating system is so popular on smartphones and tablets, and why it will probably have a significant presence on televisions in the near future.

Another way that Android differs from Apple is that Android has no approval process when it comes to apps placed in Google Play. Once the user has signed up, uploading and publishing becomes a relatively simple process. Remember how we mentioned earlier that the Android marketplace is crowded with many, many applications? A simpler approval process does mean that substandard applications can be prominent in the market. This is what makes a single quality app hard to notice. On the other hand, there are many app stores other than Google Play that an Android developer can select from. Many of these app stores have a more involved approval process. For example, Roy has seen very good results with the SlideMe store, which although much smaller than Google Play, currently results in more than 15 percent of the downloads for one of his apps.

Android apps are written primarily in Java, and Java is extremely well known. It is the most popular language in the [langpop.com](http://langpop.com) normalized comparison of computer languages. Open standards mean lots of open source. Java is the third most popular language on Google Code ([langpop.com](http://langpop.com)), which gives you an idea of how much new code is being written for it. Java also has the largest addressable user base of any smartphone operating system. It is easy to write, easy to test, easy to deploy, and has worldwide reach in multiple marketplaces. Both Google Play and the Amazon Appstore are thriving marketplaces for apps, and there are others that we will discuss in later chapters when we discuss putting your applications on the market.

BlackBerry applications are a new market for Android apps. It's easy to port an existing Android app (v.2.3.3 or later, with an update to Jelly Bean 4.1 forthcoming) to the BlackBerry platform.

The Android application programming interface (API) is very well documented, and most users come up to speed very quickly. In comparison, iOS is known to have a steeper learning curve. iOS applications are usually written using Objective C, which is rarely used outside of iOS development. In comparison, Java is a very well-known language, which makes Android easy to learn for existing Java developers.

For example, Roy knew some basic Java before writing his first app, and he was able to write a complex multithreaded application in only a few weeks. Most of the learning curve relates to the Android application life cycle, which is rather different from PC or server-side Java programming. But once you understand this life cycle, it promotes easy code reuse and allows you to tap into the power of applications written by other developers. The Android application life cycle also promotes the development of energy-efficient applications that “play nicely with others.”

But the ways that the Android ecosystem makes app development easy extend past the logistics of programming. As a novice developer, Roy was also impressed with the ease with which he was able to access an international market. His first commercial app, the Sandberg Sound BPM Detector, allows a user to determine the beats per minute of any song it hears. It's used by both DJs and musicians around the world. With a few clicks, Roy was able to deploy the app worldwide. What's even more

impressive is that written text describing the app was automatically translated to dozens of languages. Users who have no knowledge of English see Roy's apps listed in their native language, and the apps can display their text in the user's native language as well. Similarly, the Android ecosystem takes care of the logistics of international banking and purchasing. Once Roy selected an app's price in U.S. dollars, Android automatically suggested pricing in native currencies around the world. Even though his apps are listed in 190 countries around the world, purchases appear in his bank account in U.S. dollars, without any intervention on his part. Such is the power of Android.

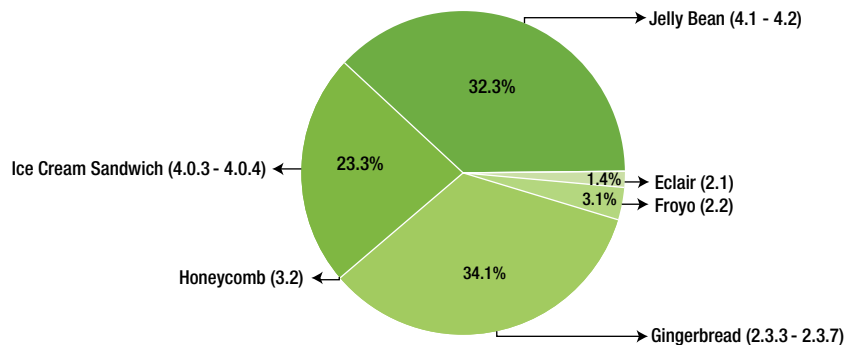
Free third-party tools also make Android easier to use, even for non-Java programmers. Scripting Layer for Android (SL4A) allows Ruby, Python, Perl, JavaScript, and a number of other interpreted languages to run on an Android device. They have access to most of the Android API and don't require a developer to follow the application life cycle. If your app is best implemented as a simple script, this might be the way to go. Currently SL4A is in alpha, but it has been under development for many years.

If you're a Ruby programmer, you could check out Ruboto ([www.ruboto.org](http://www.ruboto.org)). Ruboto uses the JRuby compiler (which translates Ruby to Java virtual machine code) to convert Ruby language code into Android application code. Because JRuby supports just-in-time (JIT) compilation, Ruby code generated by Ruboto is quite fast.

So we think Android is the easier platform to develop on. But don't just take our word for it. In a *Developer Economics 2013 Survey* of 1,200 app developers who develop for both iOS and Android, the majority stated that Android development had both an easier learning curve and a lower development cost than iOS.

## Versions of Android

Android got off to a slow start with the HTC Dream (also known as the T-Mobile G1). Since then, Android has been gaining popularity with the release of every new version. It is important that you know about them when you begin developing Android applications because newer versions contain more features than their predecessors. In terms of programming, these versions have a definite numerical designation. We will discuss that when we get into downloading Android development tools like Android SDK and Eclipse. For now, you should know that in addition to the version number, versions also have an informal name that is always a sweet treat. This cute tradition began with version 1.5.



**Figure 1-2.** Percentage of the Android universe using each platform version



Here is a very basic summary of some the latest versions of Android:

- Version 1.5 (Cupcake):
  - Allows for video recording through camcorder
  - Bluetooth enabling
  - Widgets on homescreen
  - Allows for animated screen
  - Uploading of YouTube videos and Picasa photos “on the fly”
- Version 1.6 (Donut):
  - Features a camcorder, camera, and Integrated Gallery
  - Voice Search
  - Voice Dial
  - Bookmark
  - History
  - Contacts Search
  - WVGA screen resolution
  - Comes with Google turn-by-turn navigation
- Version 2.0/2.1 (Éclair):
  - HTML5 and Exchange Active Sync 2.5 support
  - Improved speed
  - Google Maps 3.1.2
  - MS Exchange Server Integration
  - Flash for Camera
  - Bluetooth 2.1 integration
  - Option of virtual keyboard
- Version 2.2 (Froyo):
  - Screen is 320dpi with 720p
  - JIT compiler
  - Chrome with JavaScript Engine version 8
  - Wi-Fi hotspot tethering
  - Bluetooth contacts sharing
  - Adobe Flash support for version 10.1
  - Apps can be installed on expandable memory like SDcard



- Version 2.3 (Gingerbread):
  - Improved gaming graphics and audio effects
  - SIP VoIP support
  - WXGA (Xtra large screen size and resolution)
  - Near field communication
  - Copy-Paste feature
  - Download manager for large downloads
  - Better control of applications
  - Support for multiple cameras
- Version 3.0, 3.1, and 3.2 (Honeycomb):
  - The first tablet-only release
  - 3-D Desktop with newer widgets
  - Tabbed web browsing and “incognito” mode for anonymous browsing
  - Google talk Video Chat
  - Hardware acceleration
  - Multicore processor support
  - Multipane navigation
- Version 4 (Ice Cream Sandwich):
  - Streamlined user interface fit for both tablets and smartphones
  - Advanced App framework
  - Facial recognition
  - Better voice recognition
  - Web browser with up to 16 tabs
  - Resizable widgets
- Version 4.1 and 4.2 (Jelly Bean):
  - Google Now
  - Voice Search
  - Android Beam
  - Speed enhancements
  - Camera App enhancements with HDR
- Version 5 (Key Lime Pie):
  - At present, we don't know much about the features here; perhaps later versions will keep us updated.

You will discover that specific Android devices start out as one particular version of Android, and upgrades tend to come out slowly for certain devices. This is because Android releases a new version, and then hardware vendors and cellular carriers modify the source to meet their needs. In fact, the cellular carrier might not even update older phones with a new version of Android even though the hardware is most certainly capable of it.

## The Challenges of Working with Android

We have already mentioned that some of the great things about Android developer tools is they are free, and that Android differs from Apple in that the official Google Play marketplace has no approval process when it comes to apps. Once the user has signed up, uploading and publishing becomes a relatively simple process.

While Android apps are both relatively easy to develop and easy to deploy, there is no guarantee that your app will work perfectly on every Android device. You can imagine users who have a phone that your application won't work on. You won't be getting a recommendation from them!

Typical problems can include simple things like failure to format correctly for different screen sizes. Even if you follow all the best practices, sometimes a phone uses a strange resolution. Your app might still work, but the layout might be unpleasant to the user, resulting in a bad review.

It's possible to test different screen resolutions using the free Android phone emulator. But this emulator often runs too slowly for computationally expensive applications. Moreover, there are a few bugs in the emulator that can cause certain apps to behave incorrectly when emulated. Although the emulator is good for simple testing, it isn't foolproof.

Another common issue concerns Android phones with different hardware capabilities. Not all phones have forward-facing cameras, for example. Some phones don't have the "horsepower" for computationally expensive graphics. Every hardware manufacturer is free to add custom features and capabilities to its phone, which means not every phone provides every feature. Android provides the developer with ways to ensure that the phone has the features you need, but these sorts of differences between hardware platforms are one of the major challenges of programming for Android. The iPhone is a phone, but Android is a true operating system, supporting hundreds of unique devices from dozens of manufacturers.

As a developer, you should always think about how to write an app in the way that reaches the most users. Often, you are better off using an older version of Android than you prefer and avoiding the latest snazzy hardware features. This lets you reach more users. Android has an enormous installed base of users, but few of them are using the latest version.

If you get serious about Android development, you may want to buy a used Android phone or two in addition to your primary phone (which we assume is an Android phone!). Developing an app that runs on older versions of Android ensures that you can address the majority of the market. The only sure way to know whether your app works well on multiple versions is to test them. If you're committed to developing quality apps, you will want to test your app on at least a few different handsets.

Another option, if you can afford it, is to use a service like [www.perfectomobile.com/](http://www.perfectomobile.com/), which allows you to test against hundreds of real Android phones using a cloud-based interface.

## Porting Difficulties

For those who want to turn their iOS app into a full-fledged Android application (or vice versa), we want to let you know the process and pitfalls. If you're only focusing on the Android market, you can safely skip this section. However, many developers aim to develop for both Android and iOS in order to maximize their revenue. We will cover how to do that by the end of this chapter.

Let's say you've written an iOS app that is out right now or perhaps waiting in the app approval process. To turn it into an Android application, you have to adapt your software so that an executable program can be created for a computing environment that is different from the one that it was originally designed for. This is known as *porting*.

iOS apps are generally written in Objective-C, while Android apps are generally written in Java. Although the logic of these programming languages is quite similar because they are both linear, procedural, and use notions of object orientation (OO), they are very different with respect to OS support, GUI objects, and application life cycle. Sadly, Objective-C is not supported on Android.

As far as we know, there isn't any magical program that will allow you to put iPhone apps in and get Android applications out (unless you use development tools that have this in mind from the beginning). We will talk about cross-platform development tools later, but first we'll explain what you can do for both Android and iOS.

Although your iOS coding can't be reused for Android coding, you don't have to rewrite all your Android code from scratch. For example, you can completely reuse the icons and images, as well as any SQLite database code. Additionally, certain types of C code in an iPhone app (for example, code for image processing or digital signal processing) may be directly utilized in Android using the Android Native Development Kit (NDK). Although you might think that the user interface (UI) design would be reusable, iOS and Android have different UI elements, and forcing an Android app to behave like an iPhone app can be a time-consuming (and costly) endeavor. In some situations, you may be able to reuse use cases that were written for iOS apps, but you should consider the impact that the Android UI has on your use cases. Also, the Android ecosystem tends to favor free apps with advertising or in-app purchasing more than iOS does. In other words, a change to the overall business plan might be in order when porting to Android.

It usually takes nearly the same amount of effort to port an iOS app to Android as it does to create it. It really depends on how big the application is, as well as the complexity of the code, reliance on GUI tools, and ability of the developer.

By the way, there are people who make it their business to program apps, and that could lighten your workload. Since your app already exists in one form, it's easy to outsource application porting; the developer can always use the original app as a reference.

If you have written your iOS app in ANSI C or C++, perhaps using one of the many gaming engines that are designed for that purpose, then your task may be simplified. Android has the NDK, which allows for ANSI C or C++ code to be called from the Android Java code and vice versa.

## Examples of Cross-Platform Development Tools

Chances are you want your application to be downloaded as much as possible, which means you probably want it to be on as many devices as possible. If you want to have your application on both iOS and Android as well as other mobile platforms, you may want to use one of the cross-platform development kits.

Although we believe that the application market is headed toward a universal solution, we are not quite there yet. This book focuses on the Android Software Development Kit (Android SDK) for constructing Android applications. For completeness, however, we will discuss a few cross-platform development kits, just so you know there are alternatives.

## LiveCode

LiveCode is the work of RunRev, a company that creates development tools. In the words of Ben Beaumont, product manager for RunRev, LiveCode is “a multi-platform element environment that [has] now been moved to the mobile space.” LiveCode was originally made for Mac, Windows, and Linux, and it boasts “compile-free coding.” Compile-free coding means that when you make a change to your program, you will see it as you are programming. This is different from the usual method of editing, compiling, running, and debugging.

LiveCode also has a visual development environment in which the user can drag and drop the objects and images that will make up the final interface. The user can then attach scripts to these objects to really bring them to life as well as lend them speed. LiveCode uses a very high-level language, which allows the user to write in code that is very close to English. This allows users to write in this code easily, and the code will be easier to read. All this enables the creation of live prototypes that actually run on the device, and promises to make it easy to quickly iterate and improve your application, because you can immediately see the results of your work.

## Appcelerator

Titanium has created a free and open source application development platform that allows the user to create native mobile, tablet, and desktop application experiences. Its Appcelerator program allows the user to build applications full of features, as if they were written in Objective-C or Java. The end results are native apps that are customizable with a lot of features, all built with the web technology of JavaScript.

Appcelerator allows developers to concentrate on building the application and provides a toolset for many platforms.

## appMobi XDK

The appMobi mobile app development XDK is made for web developers, and it claims that if you can build an application for the web using HTML5, CSS3, and JavaScript, then you can build it as an application on the iPhone, the iPad, and Android smartphones and pads. According to appMobi, developers can develop robust, 100 percent native API-compliant mobile applications in hours using preferred editors, and write once and deploy to all target platforms.

The XDK includes an onscreen emulator with simple, approachable tool palettes to simulate user interaction with a testing device. It also allows you to send your application project over a local Wi-Fi connection or upload it to the cloud to test it from anywhere.

**Note** You will need Java 6 and Google Chrome 6.0 to run the appMobi mobile app development XDK.

appMobi also offers a service known as MobiUs, which allows any app publisher to offer its app from anywhere on the Web. This could mean the end of frustrating and complicated processes of submission and approval to traditional app distributors (and also the end of developers needing to share their profits with these distributors). It is also cloud-based, which means it is possible to create iPhone apps on a Windows PC and Android applications on a Mac.

## PhoneGap

According to its web site, PhoneGap allows users to build apps with web standards based on HTML 5.0. PhoneGap users can also access native APIs to create applications for multiple platforms, including iOS, Android, Windows, BlackBerry, webOS, and more. PhoneGap is currently in version 1.0.0.

## Summary

The Android application market has grown rapidly in the past few years. In the near future, it is likely to eclipse the Apple App Store and become the largest source of revenue for app developers. However, there are so many Android applications on the market that a developer really has to have something very different and outstanding to achieve significant revenue. Beyond that, merely having a great app isn't enough. You have to make sure your app connects with the right users, and you have to make sure your business case is sound.

Let's see how to make your app stand out from the crowd and reach the right users with a compelling business case.