

# Mobile Devices in 2011

## Abstract

*The exploding popularity of mobile devices has led to a market where new devices are being released, new programs are being developed, and new features are gaining and losing popularity very rapidly. This chapter of Libraries and Mobile Services helps to make sense of the current market in mobile devices while explaining some of the basic principles of how these devices work and are differentiated.*

Many of us in the library world have been monitoring developments in mobile technology for some time. We've recognized that these devices offered new and compelling means for access to information and have long sought to take advantage of them with, let's be honest, limited success. You could be forgiven for feeling that mobile technology for libraries is old news. You might be concerned that despite the dramatic shifts in device ownership and usage outlined in the previous chapter, our responses are limited to tools like SMS messaging.

Thankfully, the shifts in adoption and usage have been mirrored by dramatic evolution in mobile devices themselves. I agreed to write this issue of *Library Technology Reports* in September 2009. I'm completing it in late 2010 for publication in early 2011. The eighteen months that have passed between when I undertook this project and its publication are a lifetime in the mobile world (in the case of a few mobile companies, quite literally a lifetime).

In September 2009, Android released the software development kit (SDK) for its 1.6 update. Since then there have been no fewer than five more new SDKs released, and many Android phones are now running OS 2.2. The Palm Pre, running the proprietary

webOS operating system, had been on the market just a few months, and the GSM variant of the phone, which allowed it to run on AT&T's network, had just been released. Palm has since released two new versions of its phone, the Pre Plus and the Pixi. In 2010 the company was bought by Hewlett Packard, and in October, HP announced the Pre 2 and webOS 2.0. In the summer of 2009 Apple released the iPhone 3GS and celebrated the first birthday of the iPhone App Store, announcing that over 65,000 applications were available in the store and over 1.5 billion had been downloaded.<sup>1</sup> Two months later those numbers were 85,000 and two billion, respectively.<sup>2</sup> The most recent information available shows that by September 2010, the App Store had over 250,000 applications which had been downloaded over 6.5 billion times.<sup>3</sup>

If it seems that a lot has changed since I undertook this project, think of how different the mobile device landscape is from when Ellyssa Kroski wrote her LTR.<sup>4</sup> The devices available gratis with new mobile contracts have a hardware and software feature set the like of which was unthinkable in 2007. We used to talk about "smartphones" and "feature phones." Smartphones were those few devices that had PDA, e-mail, and rudimentary Web connectivity. Feature phones (sometimes referred to as "dumb phones") were those that were single-function devices, good primarily for making calls and texting. At best, feature phones sometimes had a very rudimentary WAP web browser, many of which were restricted to accessing a portal created by the wireless carrier, devoted to add-on services like paid ringtone downloads.

The era of the feature phone is at an end, and the dramatic difference between the devices Kroski addressed in her LTR and today's mobile devices can hardly be overstated. Today virtually every phone



**Figure 9**

A “clamshell” flip phone (Photo credit: aeiss, “Phone,” [www.flickr.com/photos/ohaihtar/4365544298](http://www.flickr.com/photos/ohaihtar/4365544298), licensed under the Creative Commons Attribution 2.0 Generic license, <http://creativecommons.org/licenses/by/2.0/deed.en>).

offered by the major U.S. wireless providers has a wide range of hardware and software capabilities, including unrestricted access to the open Web.

The opportunities for libraries have likewise expanded. Improved screen technology allows better reading experiences. Third-party application markets are hotbeds of new information access models. High-resolution cameras turn any mobile device into a barcode scanner. GPS chipsets allow for innovative location-based services.

The pages following are a crash course in the mobile device features that have so profoundly changed the capabilities of these devices, and by extension, our relationships with them. These are the hardware and software features that we need to keep at the forefront of our service delivery strategies as we enter the second decade of the millennium.

## Hardware

### Form Factors

Form factor refers to the hardware configuration of a device, including its shape and available physical methods of input. We’ve seen a number of trends in mobile form factor over the years, such as the “flip phones” many of us carried in the early to mid-2000s.

Flip phones, or “clamshell” devices (figure 9), are those that fold in half, typically showing a small auxiliary display on the outside and concealing a larger, higher-resolution screen and keypad within. If you owned one of the approximately 50 million RAZR phones sold by Motorola, you owned a flip phone. Another early trend in mobile devices was the so-called “candybar” form factor, which placed the keypad and screen on the same side of a slab-shaped device. The candybar seems like the inevitable product of the



**Figure 10**

Phone with a QWERTY keyboard (Photo credit: Cheon Fong Liew, “Blackberry Bold in hand,” [www.flickr.com/photos/liewcf/3741159735](http://www.flickr.com/photos/liewcf/3741159735), licensed under the Creative Commons Attribution-ShareAlike 2.0 Generic license, <http://creativecommons.org/licenses/by-sa/2.0/deed.en>).

ongoing miniaturization of the classic Zack Morris-style cellular phone. If you ever owned a Nokia phone that allowed you to play Snake on a black-and-white screen, you owned a candybar phone.

With the notable exceptions of some wild experiments on the part of Nokia and Samsung, the past few years have seen a consolidation in form factor centering on two key features: the QWERTY keyboard, and the touchscreen.

### QWERTY

Driven in large part by the innovations of RIM’s BlackBerry devices, various permutations of the full computer keyboard on mobile devices (figure 10) became quite popular, particularly in the corporate enterprise world. These QWERTY devices (named after the first six characters on a standard keyboard) wore their productivity lineage on their sleeves. They were optimized for rapid access to e-mail and instant messaging, and many an executive became as facile with the BlackBerry keyboard as with the computer’s. The first few iterations of the BlackBerry didn’t even have phone capabilities and were strictly e-mail devices that owed more to the alphanumeric pagers of the late 90s than to the mobile phones of the time. In many ways, we’ve now come full circle, and the phone has for many users ceased to be the central feature of their mobile device.

Indoctrinated in many cases by their employers, users soon began to demand the “push” e-mail and productivity features of RIM’s devices, and BlackBerry eventually made a play in the consumer market. By the late 00s, RIM’s QWERTY devices were available on nearly every carrier. These devices sat alongside

other QWERTY devices like Danger, Inc.'s Sidekick, an innovative device that was the first consumer-oriented phone with a true network focus, sporting an always-on data connection and excelling particularly in instant messaging capabilities. One of Danger's founders, Andy Rubin, later went on to create the Android mobile OS for Google.

Today there are many QWERTY devices on the market. Some retain the traditional BlackBerry form factor, with a large keyboard sitting below a squarish screen. Others conceal the QWERTY keyboard behind a large screen, allowing for basic functions when closed and sliding open when it's time to do some serious typing. However, even BlackBerry has been experimenting with form factors that eliminate the keypad entirely. The form factor that seems to rule the day is the slab-like touchscreen.

## Touchscreen

The flagship touchscreen device of the past few years has been Apple's iPhone (figure 11). This is the device that proved that less can be more when it comes to buttons. There was great skepticism at the iPhone's launch about its touchscreen interface, particularly the on-screen keyboard. Some users accustomed to clicky keyboards, scroll wheels, and trackballs felt there was no way they could maintain the same comfort and efficiency when entering information on a device that provided no physical feedback.

What many users found, however, was that the keyboard was pretty decent, and the autocorrect software was great. Combined, the two served to make text input on the iPhone good enough to inspire sales of around 75 million devices before the end of 2010. More importantly, many users found that the direct manipulation of on-screen elements using the intuitive tap, swipe, and pinch gestures foundational to the iPhone's operating system was shockingly intuitive. Online videos have proliferated showing infants and toddlers presented with the iPhone for the first time, and invariably show the children swiping through photos or interacting with games with ease.

The usability of the iPhone's touchscreen owes much to the underlying software, but there's a hardware story here as well. The iPhone was one of the first consumer devices to use what's called a capacitive touchscreen, one that senses the presence of a finger by its effect on an electric field. This was a leap forward from the resistive touchscreens commonly used on PDAs and other devices before the iPhone. Resistive touchscreens rely on actual physical pressure to record input. The move to capacitive touchscreens allowed for more accurate input without the use of a stylus and for the simultaneous tracking of multiple inputs, so-called multitouch.

Touchscreens have become common on many



**Figure 11**

The iPhone (Photo credit: Yutaka Tsutano, "iPhone 4 32GB Black," [www.flickr.com/photos/ivyfield/4731067532](http://www.flickr.com/photos/ivyfield/4731067532), licensed under the Creative Commons Attribution 2.0 Generic license, <http://creativecommons.org/licenses/by/2.0/deed.en>).

mobile devices. Google's Android OS is optimized for touchscreen devices, as is HP/Palm's webOS. As alluded to above, RIM has even released a series of touchscreen devices without its trademark keyboard, though the first few were hampered by resistive, rather than capacitive, screens.

Nokia, long the leader not only in phone sales, but in smartphone innovation, used to boast about the number of buttons on its high-end phones. Its phones had camera buttons, menu buttons, media playback buttons, number buttons, letter buttons, and many more. Some models topped out at more than 100 buttons, each dedicated to a single function. Compare this to the iPhone or Nexus One, Google's flagship Android device released in early 2010. Each has a number of buttons you can count on one hand. However, their touchscreens ensure that every application can implement as many custom controls as necessary, in effect an infinite number of buttons, customized for each application.

## Components

### Screens

It bears mentioning here that touch isn't the only evolution that's come to mobile device screens in the past few years. The phones available to most of us today when we initiate or renew a contract feature high-resolution color screens. These screens, as much as any other innovation, are what has enabled mobile devices to transcend simple "phonehood." Increasing the resolution of the display allows for more information per screen, more legible rendering of type, and quality display of photos that has likely relegated the wallet-sized print to the dustbin.

Screen resolution is described with two figures, pixel dimensions and pixel density. If you've ever fiddled around with hooking a laptop up to a projector, you're likely familiar with pixel dimensions. The projector in the conference room nearest my office maxes out at  $800 \times 600$ . These numbers refer to the number of individual pixels available on a screen to create an image. In the example above, the projector can use 600 horizontal lines of 800 pixels, each of which can be a single color at any given moment, to create the image on the screen. The pixel dimensions of mobile devices are rapidly approaching those of the computer monitors many of us used through the 2000s. One of the flagship Android devices of the summer of 2010, the HTC Evo, featured a  $480 \times 800$  screen.

Pixel density takes both the pixel dimensions and the actual physical dimensions of the screen into account to provide pixels per inch, or PPI, a measure of a given screen's ability to display a smooth, detailed picture. Apple branded the screen used in the iPhone 4 as the "Retina Display" because the pixel density supposedly approaches the point where it is physically impossible for the human eye to distinguish individual pixels. This screen has pixel dimensions of  $960 \times 640$  in a 3.5-inch display, for a pixel density of 326 PPI.

While you may still be able to find a phone featuring an old-fashioned black-and-gray LCD, the only monochrome devices I've seen on the market recently have been phones marketed to seniors, children, and others needing only a truly bare-bones device capable only of making phone calls, in some cases only to a predefined set of emergency numbers.

## Cameras

Speaking of photos, many of the fascinating developments in mobile device software and interactions that we'll look at later in this report rely on the ever-improving cameras that are ubiquitous on today's phones. I'm not kidding about ubiquitous; it's come to the point that the absence of a camera is a feature for some enterprise mobile consumers. If you're the guy buying BlackBerrys for a Department of Defense contractor, you need to specifically order the model with no camera to keep the nation's secrets from flying out over the airwaves.

Describing camera quality is a challenge. The easiest measure is the camera's resolution, the number of pixels that the image sensor uses to capture the image. However, the hardware and software in digital cameras, including those embedded in mobile devices, have advanced to the point where raw megapixel count no longer suffices to quantify image quality. A five megapixel sensor can theoretically capture an image detailed enough that you can make photo-quality  $8 \times 10$ s, but if the camera's lens distorts, or the image capture software can't intelligently adjust exposure to

avoid shadowed faces or a blown-out white sky, those  $8 \times 10$ s won't find a place on your mantel.

That said, it's increasingly common for mobile devices to ship with embedded cameras that are objectively quite good. A couple of years ago, I spent many hours researching before buying a digital camera that I felt hit the sweet spot between image quality, lens and optical quality, zoom, and compact form factor. That camera has gathered dust for months as my phone has become my exclusive camera. Actually, I did take my stand-alone camera on a recent trip to Mexico. I didn't want to risk my phone getting wet at the beach.

## Wi-Fi

While mobile carriers have been eager to push smartphone devices that require additional data service plans, thankfully, many of the Android, iOS, Palm, and BlackBerry devices available today also come equipped with Wi-Fi radios. With Wi-Fi, users can connect to their home or other Internet connections, often at speeds much greater than what's available through their data plan. iPhone users are well acquainted with the necessity of Wi-Fi, as AT&T won't allow users to download applications or media files larger than 20 MB over their data connection. Users running into this limitation are notified, hypothetically, that downloading *The Secret of Monkey Island* requires connection to a Wi-Fi network or syncing to a computer.

In smartphones with robust third-party software development communities, such as Android and iOS, the Wi-Fi radios have become a workaround source of connectivity for applications that appear to compete directly with the business of the mobile carriers. On iOS, for example, Skype developed a native application that allows users to make free voice calls to any Skype account holder, and to any phone number for a very small per-minute tariff. The application was approved through Apple's App Store, but it can connect only over Wi-Fi, not using AT&T's network.

## GPS

Add the standalone car-mounted GPS unit to the list of devices made obsolete by multifunction mobile devices (figure 12). Beginning in the mid-2000s, smartphones had very primitive geolocation applications that used the unique names of cell towers to pinpoint a user's location. Google, for instance, shipped a Google Maps application for BlackBerry that used this cell tower information to place the user on a map. The original iPhone shipped with a similar location service in the Maps application.

The capabilities and accuracy of mapping and navigation applications increased greatly with the inclusion of actual GPS chips in mobile devices. Current-generation devices use a sophisticated combination

of cell tower, Wi-Fi access point, and GPS satellite navigation to pinpoint a user's location with surprising rapidity and accuracy. An iPhone user can quickly perform Google Maps searches and get directions between locations optimized for driving, walking, or in many metropolitan areas, even public transit. Most current Android OS devices ship with true turn-by-turn navigation that can speak directions aloud.

The location services in mobile devices isn't limited to maps and navigation. Increasingly, users can grant permission to third-party applications to selectively access location. A Yelp.com application, for instance, can point out nearby restaurants. The World-Cat application can show you the library nearest to your current location. Various fitness applications allow users to track their cycling or running routes. More on applications later.

### Bluetooth

I'm sure we all still shudder when we recall the plague that afflicted the nation's yuppies in the mid-2000s. Previously competent and successful legions of suited businessfolk roamed our city streets and airports yelling loudly and gesticulating wildly, each sprouting a similar unsightly blinking growth from one ear or the other. They were the Bluetoothed, determined to infect every public space with loud one-sided conversations with what we can only assume were demonic tormentors.

Thankfully, the worst of the epidemic seems to have passed, and we can now focus on rebuilding the reputation of the otherwise benign radio standard known as Bluetooth. While Bluetooth radios have been common on phones for a number of years now, only recently have handset manufacturers been building in support for some of the more interesting features of the technology. No longer is it simply a means for connecting monaural headsets. Users can now use stereo Bluetooth headsets to listen to music without tangled headphone wires, and connect Bluetooth keyboards to turn their mobile device into a full-fledged productivity workstation.

### Flash Memory

Not to be confused with Adobe's Flash multimedia technology, flash memory refers to solid-state data storage technology. In a process analogous to Moore's Law, the price and capacity of flash memory have declined and increased respectively at an impressive rate. The first iPod to ship with solid-state flash memory instead of a traditional hard drive was 2005's iPod Nano, the cheapest version of which was \$199 for 2GB of storage. Today's iPod Shuffle offers the same storage for \$49.

I mention iPods not just because Apple is the

world's largest consumer of flash memory, but because the gap between mobile phones and multimedia players has narrowed to near-invisibility in recent years, largely because of the increased storage capacity of mobile phones. Where once phones shipped with a few megabytes of storage, suitable only for a handful of low-res camera phone photos or \$2.99 ringtones, now it's common for smartphones to carry up to 32 GB of storage. Many mobile devices also allow for expanded storage using SD cards. Apps, MP3s, videos, high-res photos, and videos all require ample storage space, and it's still easy to max out the available space on a device.

### Batteries

Many of us have become accustomed to re-upping our phone contracts every couple of years and receiving a shiny new device in return for our continued loyalty. With each upgrade we've gotten a vastly expanded and improved feature set—Bluetooth, color screen, multi-megapixel camera, gigabytes of multimedia storage, video calling, and so on. There's been one notable exception to this otherwise uninterrupted march of forward progress: battery life.

Where once all but the heaviest users of mobile phones could expect to go days per charge, we've now entered an era where users of most devices of recent vintage need to charge them daily. It may be that battery technology hasn't quite kept pace with the other device innovations, but it is also true that all of the features of our new phones come with a price. Big bright color screens require far more energy to power than did the black-and-white LCDs of our Snake-playing Nokias. Likewise, Wi-Fi connectivity, Bluetooth, and GPS each rely on new power-hungry radios. Our devices have also become steadily smaller, allowing less space for batteries. There's also the inevitable consequence of increased functionality (see figure 13).

### Software

Of course, what really differentiates today's smartphones from their predecessors isn't the hardware, which largely consists of predictable evolutionary



**Figure 12**  
Using GPS on a webOS smartphone.



**Figure 13**

A valid point regarding battery life in today's devices (Source: rands, July 22, 2008, <http://twitter.com/#!/rands/status/865671906> [accessed Jan. 4, 2011]).

steps from previous iterations, but the software, which has taken these devices into completely new realms of functionality. The software that's setting today's top-of-the-line phones apart from their ancestors comes in two broad categories: smartphone operating systems, which are developed or licensed by handset manufacturers, and applications that run within those operating systems, often developed by third parties.

Almost without exception, current-generation smartphone operating systems provide a suite of functions that until recently were available only on desktop and laptop computers, on specialized consumer electronics, or in PDAs that lacked always-on connectivity. Some of those features are as follows.

### E-mail

Every smartphone operating system today ships with its own sophisticated e-mail client built in. The vast majority will support multiple e-mail accounts, allowing users to access, for instance, both a personal and a work e-mail account from the same device. Many of these e-mail clients provide native access to popular Web-based e-mail services like Gmail, Hotmail, and Yahoo! Mail. Increasingly, these mobile e-mail clients support Microsoft Exchange server. Exchange is perhaps the most common e-mail server for the enterprise, and Exchange support means many users can access their work e-mail securely, including server-side folders. Exchange support also allows server administrators to remotely wipe data from mobile devices securely.

Note: Most smartphone e-mail clients do support HTML e-mail messages, but designing HTML messages that degrade appropriately for the smaller screens and higher pixels-per-inch on mobile devices may be even more challenging than developing HTML for mobile websites. If you typically send users HTML e-mails, watch your usage statistics for mobile browsers and

user agents. If you see many users viewing your messages on mobile devices, consider reverting to plain text messages. If your marketing strategy requires the analytics provided by HTML e-mails, simplify your message layout to allow the e-mail client to zoom and reflow text for legibility.

### Calendar

Exchange support also allows enterprise users to integrate Microsoft Outlook calendars with their mobile device calendars. The calendars built in to most mobile operating systems sync wirelessly with Exchange or CalDAV servers, allowing for real-time updates and notifications from services like Google Calendar or servers like Oracle Calendar (depending on your server configuration).

### Multimedia

In many ways, today's smartphones are directly evolved from both past mobile phones and past portable multimedia players. BlackBerry smartphones have moved from business-oriented messaging, calendaring, e-mail, and phone capabilities into the consumer multimedia realm, adding support for MP3 and other audio and video file formats. Apple's iPhone evolved in large part out of the company's iPod line of media players, adding a phone and PDA functions to what was arguably the best media player on the market.

These converged devices offer increasingly robust support for multimedia. Many devices allow users to load media files by mounting the device on a PC like a USB drive. Users can then drag and drop files into the mobile device's onboard storage and watch movies or listen to music or audiobooks on the go. Apple's iPhone syncs to the iTunes desktop application to load media, rather than mounting as a drive. This approach has pluses and minuses. Users are unable to directly manipulate the files on their device, and moving a single file onto or off of the iPhone requires a full sync. The syncing process does offer the benefit of allowing users to specify, for instance, that episodes of podcasts that they've already listened to be removed from the device upon syncing, automatically freeing space for additional media. Windows Phone 7 syncs media using the Zune desktop client. A number of third parties offer desktop software that allows for a similar type of syncing for BlackBerry, Android, and Palm devices.

At launch, the Palm Pre purported to offer media syncing through iTunes, a feature that would have made the Pre unique among non-Apple devices. This syncing capability was achieved without Apple's permission, and subsequent versions of iTunes disabled the feature.

The system-level support for many multimedia

codecs means that not only can smartphone users load audio and video files from their personal libraries, but they can often access media files embedded in webpages and otherwise available online.

## Web Browsers

Perhaps the most important, and likely one of the more underrated aspects of current mobile device software is the mobile web browser. As more and more of the applications we rely on in our work and personal lives have moved to the cloud, the reliable, speedy, standards-based web browser has done more to legitimize mobile devices as true productivity tools than just about anything else.

Today all of the major smartphone operating systems ship with a web browser that rivals major desktop platforms in its support for web standards. In addition, these browsers offer important affordances for the smaller screen sizes of mobile devices. They allow for intelligent zooming of webpages, identifying sections of the page and zooming to fit columns of text, and so on.

The signature evolution of this generation of mobile devices, however, is likely the rise of widely available high-quality third-party software applications. This is the age of the app. These programs range from bite-sized to multi-gigabyte, and from games to medical and automobile diagnostics. They take the bare essentials of the mobile device—a network connection, a camera, geolocation, and so on—and using the highly configurable touchscreen interfaces on most devices, transform your phone into something entirely other. Today apps are available for many smartphone platforms, with varying degrees of quality. In this report I'll focus on five operating systems: iOS, Android, webOS, BlackBerry OS, and Windows Phone 7.

## iOS

iOS, previously called iPhone OS, launched with the first iPhone in 2007. The operating system is derived from Apple's OS X operating system, which is used on its laptop and desktop computers, as well as Apple servers. The OS is optimized for the few pieces of hardware upon which it runs legitimately: the various models of the iPhone, iPod Touch, and iPad. As befits these devices, the user interface for iOS is touch-centric and app-centric (figure 14). Every function of the device is contained within an application. The Settings application contains general device settings and preferences.

The iPhone launched in 2007 with just a small handful of native applications, all of which were developed by Apple. In addition to the core Phone, Messages (for SMS and MMS text messaging), Mail, and iPod applications, the iPhone included the Calculator



**Figure 14**

The home screen on the iPhone 4. (Photo credit: William Hook, "iPhone 4 - Retina Display - Home Screen," [www.flickr.com/photos/williamhook/4742863198](http://www.flickr.com/photos/williamhook/4742863198), licensed under the Creative Commons Attribution-ShareAlike 2.0 Generic license, <http://creativecommons.org/licenses/by-sa/2.0/deed.en>).

app, Maps app, YouTube app, Notes, and so on. While many suspected that it might eventually be possible for third-party software developers to write native iPhone applications, initially Apple did not offer any means to do so, and instead touted the ability for developers to write robust web applications.

Unsatisfied with the limited capability of web applications, third-party developers soon discovered ways to break in to the iPhone's closed file system and load their own applications onto the device. The process of circumventing the security measures in the iPhone's software and hardware is known as "jailbreaking." The popular iPhone applications Twitterrific and Tap Tap Revenge began as samizdat software for jailbroken iPhones before it was possible to develop legitimately and before any documentation was available for developers. Late in 2007 Apple announced the forthcoming release of an official software development kit (SDK) for the iPhone, which would allow developers to create native iPhone applications and to submit them for sale through the forthcoming App Store.

Applications for iOS devices are written using a language called Objective C, the same language used for applications in Mac OS X. Prospective iOS application developers who register with Apple can download for free the software development kit (SDK) for developing iOS applications. The (hefty) download includes Apple's Xcode application, an integrated development environment used for creating both iOS and Mac OS X applications. The SDK also includes iPhone and iPad simulator applications, which allow developers to test their applications on screen. These simulators, like Xcode, are OS X-native applications, and will not run on Linux or Windows. Also included in the SDK is Dashcode, an application for developing

iOS-optimized web applications.

iOS applications are available only through the iTunes App Store, either in the iTunes desktop application, available for Windows or Mac OS, or directly on iOS devices themselves. Dating back to the days before the App Store existed, there has always been an appetite among some developers and users to sidestep the App Store in order to install directly on the device software that wouldn't pass Apple's rigorous approval process. Installation of unapproved third-party software requires the process euphemistically referred to as jailbreaking. This process also allows users to bypass the so-called "carrier lock" that keeps the iOS device tied to AT&T's network, enabling its use on any compatible network.

Though jailbreaking has become quite popular as a way to customize one's iOS device or to run an iPhone on T-Mobile's GSM network rather than AT&T's, it is also a risky proposition. Jailbreaking relies on exploiting holes in iOS's security model, holes which could also be used to compromise users' personal data on the device. The jailbreaking community catalogs known exploits, but takes care to use only one at a time, knowing that Apple will always patch whatever vulnerability has been exposed by the latest jailbreaking method. This means that at any given time there are a number of outstanding exploits awaiting future use as jailbreaking methods that could in the meantime be put to nefarious use.

## Android

The roots of the Android OS (figure 15) lie in the early development of another smartphone, the Sidekick. Created by a company called Danger, Inc., the Sidekick was notable as one of the first consumer-oriented device with a full QWERTY keyboard and with so-called "push" technology, which allowed for near-instantaneous delivery of data to the device. Previously available only to enterprise customers who could deploy a BlackBerry server, push data service provided not only e-mail delivery, but always-on instant messenger capability. The Sidekick's implementation of AOL's AIM instant message protocol made it popular not only with its target teen and twenty-something demographic, but with hearing-impaired users.

Under Danger, Inc.'s direction, the Sidekick went through a number of hardware revisions over several years on T-Mobile's network. In 2008, Danger was purchased by Microsoft. Shortly thereafter, Danger's former CEO, Andy Rubin left Microsoft to found a company called Android. Android was subsequently purchased by Google, which further developed and released its mobile operating system as Android OS. Andy Rubin has stayed on at Google, where he oversees Android development.

## Free as in Android

Android is unique among the major smartphone OSes in that it is ostensibly free and open source, though in Android's case, both of these terms require some caveats. Beginning with open source, Android is not strictly a community-driven open source project like many of the standard bearers of free/libre/open source software (FLOSS). Key aspects of Android OS development are completely contained within Google. As stated

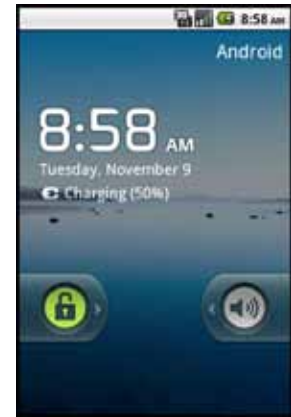
on the Android Open Source Project page, "Some parts of Android are developed in the open, so that source code is always available. Other parts are developed first in a private tree, and that source code is released when the next platform version is ready."<sup>5</sup>

Google has released the source code of Android under the Apache software license. The Apache license differs from the GNU Public License (GPL), which uses a more traditional open source model and requires that derivative software also be released as open source. Under the Apache license, developers and handset manufacturers can add proprietary code to their implementations of Android without having to release it back to the open source community.

So, Android is open source and thus free as in freedom. For handset manufacturers and mobile carriers, it is also ostensibly free as in beer, but in practice that may not actually be the case. While Google will not charge anyone for Android, there are a number of companies that assert that Android implementations infringe on their intellectual property, and they have begun filing lawsuits demanding license fees.

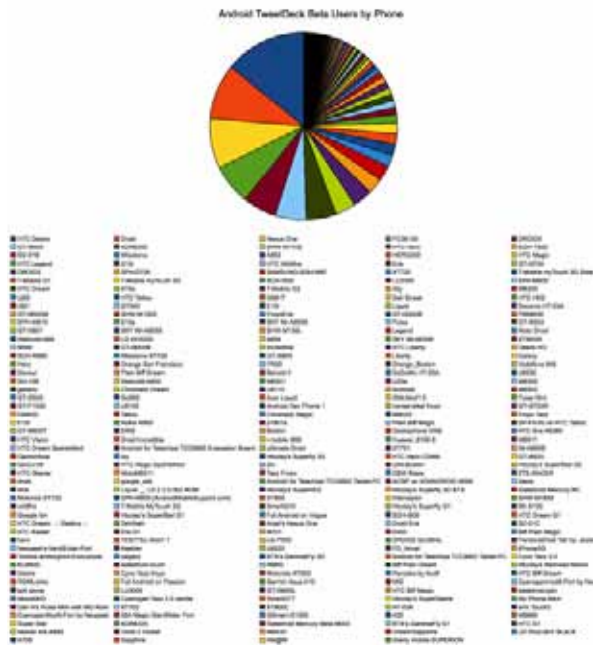
One of the companies that is targeting Android devices with licensing suits is Microsoft, which has successfully extracted license fees from phone manufacturer HTC for its Android handsets.<sup>6</sup> HTC has long been a partner with Microsoft on Windows Mobile devices, and so the matter of patent licenses for HTC's Android phones was handled quietly and out of court, meaning we don't know much about what intellectual property was at stake or what fees HTC paid.

Microsoft CEO Steve Ballmer, seemingly very confident, asserted in a recent Wall Street Journal interview that implementing Android would not be free for handset manufacturers, "Android has a patent fee. It's not like Android's free. You do have to license patents. HTC's signed a license with us and you're going to see license fees clearly for Android."<sup>7</sup>



**Figure 15**  
The Android operating system.





**Figure 16**  
 Android has become popular and its usage diverse (Source: Toby Padilla, “Android Ecosystem Infoporn Overload,” TweetDeck blog, Oct. 12, 2010, <http://blog.tweetdeck.com/android-ecosystem> [accessed Jan. 4, 2011]).

**Android Adoption, Implementation, Fragmentation**

So far, the threat of patent license fees doesn’t seem to be slowing the rate at which handset manufacturers are adopting Android and turning out handsets. Over the past year and a half or so, best-of-breed Android handsets have been released at a staggering rate (figure 16). Between the summer and fall of 2010, Android surpassed iOS in the number of phones shipped per month (note that iOS also ships on nonphone devices). A recent study released by the developer of the popular crossplatform Twitter client TweetDeck showed that among 36,427 Android users, there were nearly 250 distinct phone models in use.<sup>8</sup> Every major U.S. mobile carrier has at least one Android handset on offer, and many have a half-dozen or more.

While this explosion in Android adoption and handset models bodes well for the future of the platform and for prospects of development targeted at Android, it also speaks to some challenges. In order to provide the best possible user experience, app developers must take hardware into account as they develop software, and the sheer number and variety of Android handsets on the market makes this task very difficult.

Likewise, because of Android’s liberal software license, it’s not necessarily even appropriate to speak of it as a single platform. Handset manufacturers like HTC and Motorola, as well as carriers like Verizon, have been shipping handsets with versions of Android



**Figure 17**  
 Another illustration of the diversity of Android usage.

so highly customized that they are incompatible with future updates to the Android OS, which instead have to be similarly customized before being shipped to users. The same survey by TweetDeck found over 100 different variations on the Android OS were running on their users’ phones (figure 17).<sup>9</sup> While a significant majority were using recent mainline Android releases (2.2 and 2.1), a still-significant number were using outdated or splinter versions of the OS that would likely be incompatible with applications developed for the current Android release.

**Android App Development**

Android development is very similar to Java development—so similar, in fact, that Oracle, the owner of key Java patents, has sued Google for infringement. Regardless, developers who are accustomed to developing Java applications should be able to transition to Android applications with little difficulty. Google makes a number of tools available to developers in its Android Developers site. Here developers can contribute to the open portions of the Android source code, as well as read documentation and best practices for developing native Android apps. The Android SDK is also available and includes a desktop emulator developers can use to test applications. Google’s recommended method for writing applications is to use the crossplatform Java-based Eclipse IDE, for which it provides an Android Development Tools plugin.

*Android Developers website*  
<http://developer.android.com>

Developers wishing to publish or sell their Android application through Google's Android Marketplace can pay a \$25 fee to register as a Marketplace Developer, after which they can submit applications. The aforementioned hardware and software fragmentation in Android extends to app stores, however. First, not all Android devices are compatible with the Android Marketplace. Google limits Marketplace access to devices that meet a stricter set of guidelines, among which at the time of writing was the need for a persistent network connection from a mobile carrier. This requirement has so far hampered the emergence of a contract-free iPod Touch equivalent running Android. Google's Android Marketplace is not the only game in town, however. Both Verizon and Amazon.com have announced intentions to open Android app stores, and it's speculated that handset manufacturer HTC will do the same.

### webOS

The launch of the Palm Pre (figure 18) at the 2009 Consumer Electronics Show was seen as a welcome return to form for the company that defined the PDA in the 90s and pioneered the PDA/phone hybrid in the 00s with the Treo. Industry pundits, watching Palm's balance sheets carefully, predicted that the Pre was a do-or-die move for the company. In the years leading up to the Pre's launch, Palm had become a haven for a number of ex-Apple employees, including Jon Rubinstein, who lead Apple's iPod division. It came as little surprise then, that the Pre featured both hardware and software worthy of an Apple device, polished, thoughtful, and innovative.

The Pre featured a multitouch touchscreen interface, the first commercially available phone other than the iPhone to have one, a feature that provoked talk of lawsuits from Apple. Unlike the iPhone, it also featured a slide-out hardware keyboard and a small trackball, both of which were entirely optional, as all device features could be operated with the touchscreen. Palm later introduced the Pixi, a slimmer version of the Pre that sported a fixed keyboard below the screen. The Pre was the first smartphone to cleverly integrate a number of Web-based services, allowing users to sign in to Facebook, Twitter, and Gmail, and then see contacts from all three services seamlessly integrated into the phone's contact list. The Pre also wisely included an application that emulated the earlier Palm OS, allowing longtime Palm users to load dearly loved applications that no longer run on any other device.

The operating system running both the Pre and Pixi is known as webOS, an appropriate name since even native applications for the device are largely written using web standards and programming languages. webOS applications can be written entirely in HTML, CSS, and JavaScript, using the webOS Mojo



**Figure 18**  
webOS on the Palm Pre.

JavaScript framework, which provides access to the native device functions. Applications can be submitted for distribution through the webOS App Catalog, where they are available on users' devices or through RSS feeds.

The barrier to entry for writing native webOS applications is by far the lowest of all the major smartphone platforms. Yet, a recent survey of application developers by Millennial Media, a mobile advertising firm, found

that only 4 percent intend to target webOS in 2011.<sup>10</sup>

The Pre may not have been a do-or-die device for Palm. While it was generally a critical success, it saw only modest consumer adoption, perhaps in part because it launched exclusively on Sprint's network, one of the smallest of U.S. carriers. Just over a year after the launch of the Pre, Palm was acquired by Hewlett-Packard, which now markets the Pre as the HP Palm Pre, and webOS as HP webOS. There is every indication that HP intends to throw considerable corporate weight behind the development of webOS devices. webOS 2.0 was launched in late 2010 alongside the Pre 2. Industry watchers speculate that HP's interest in Palm and webOS had as much to do with future developments of tablet computers as with the smartphone business.

### BlackBerry

If there is a granddaddy of mobile information access, it's the BlackBerry. Developed by Canada's Research in Motion in the late '90s as a device that had more in common with a pager than a phone, the BlackBerry was for many years the only option for users who needed access to e-mail at all conceivable times. The BlackBerry has been synonymous with harried executives and thumb-cramped attorneys for many years, and only in the past several years has it made inroads into the consumer market.

The first BlackBerry devices were small, with a monochrome screen allowing users to read or compose just a few lines of a message at a time. There was no phone, no calendar, no games, just e-mail. But what e-mail! Using a BlackBerry required installation of the BlackBerry Enterprise Server on a company's e-mail server. Once it was installed, messages would be delivered to BlackBerry devices in real time, allowing for near-instantaneous e-mail communication anywhere. The device feature set expanded,



**Figure 19**  
Using e-mail with a QWERTY keyboard on a BlackBerry device.

eventually incorporating all the features one would expect from a phone and PDA. By the mid-00s, RIM developed the BlackBerry Internet Service, which allowed cellular providers to run a BlackBerry server and offer to consumers the same lightning-speed e-mail delivery previously available only to corporate clients.

BlackBerry's run at the consumer market was remarkably successful. Users who'd become accustomed to the BlackBerry service with phones issued by their employers snapped up BlackBerries for personal use, and devices like the Pearl, with a more pocketable form factor than the traditional BlackBerry slab, sold quite well. Unfortunately,

while BlackBerries excel in e-mail and instant message delivery, they have largely failed to offer quality Web access, using an antiquated and substandard browser. Likewise, the BlackBerry interface, no doubt a model of efficiency, has lacked the polish and design of competing products. Whether for these or other reasons, BlackBerry sales have suffered as Android and iPhone sales have soared, particularly in the enterprise.

RIM has made steps to compete, some more successful than others. In response to the iPhone, RIM released its first touchscreen BlackBerry, the first BlackBerry without a physical keyboard. Unfortunately, the device was roundly panned for having an unresponsive touchscreen that inhibited the extremely rapid input that BlackBerry users craved. Subsequent iterations of BlackBerry touchscreens have been better received. Likewise, with the release of BlackBerry OS 6.0 in late 2010, RIM has implemented a new web browser based on the industry standard WebKit, allowing for web rendering parity with other platforms.

BlackBerry offers a great deal of documentation for potential developers on its developer portal. Native BlackBerry application development is accomplished using Java, and the only supported method is to use the Eclipse IDE on a Windows machine. The BlackBerry SDK also provides a Windows-based emulator, allowing developers to preview applications on their desktop machines. Developers can submit applications to BlackBerry App World, a marketplace for



**Figure 20**  
Social media applications on the touchscreen of a BlackBerry device.

BlackBerry applications that is accessible both on BlackBerry devices and via the Web.

*BlackBerry Developer Zone*  
<http://us.blackberry.com/developers>

*BlackBerry App World*  
<http://appworld.blackberry.com/webstore>

## Windows Phone 7

Microsoft's Windows has quite a legacy in the smartphone world, and yet is also a complete newcomer. After years of development on the Windows Mobile platform, in 2010 Microsoft launched Windows Phone 7 (figure 21), which represented a complete clean break from its earlier mobile operating system. The business model for Windows Phone 7 is strictly as a software platform, much like the Windows desktop OS. Microsoft does not manufacture any of the Windows Phone 7 handsets, but instead licenses its operating system to manufacturers like Dell and HTC, which then build phones according to a fairly strict set of specifications determined by Microsoft. The handset manufacturers then sell their phones to mobile carriers.

The initial rollout of Windows Phone 7 devices was fairly well received by tech reviewers and consumers, though it remains to be seen if it will gain a foothold in the already crowded smartphone OS market. Like Android and BlackBerry, Windows Phone 7 had the advantage of having handsets available on multiple carriers at launch. In many ways, Windows Phone 7 devices mirror the first-generation iPhones: they lack meaningful multitasking and copy-and-paste clipboard functionality. However, they do have access to a third-party app marketplace at launch.

Additionally, Windows Phone 7 devices feature robust integration with other Microsoft tools and



**Figure 21**

Windows Phone, one of the newest mobile devices (Photo credit: Luca Viscardi, "windows phone 7 - 1," [www.flickr.com/photos/viskas/5104024799](http://www.flickr.com/photos/viskas/5104024799), licensed under the Creative Commons Attribution-ShareAlike 2.0 Generic license, <http://creativecommons.org/licenses/by-sa/2.0/deed.en>).

services. Each handset ships with a native Office suite that integrates deeply into the OS, allowing users to open and edit Office documents from their e-mail or from the Web. Users of Microsoft's Zune subscription service will be able to access their music and videos seamlessly on the device either by syncing files from a PC or Mac or by streaming. The device also integrates with the Xbox LIVE online gaming service, providing access to users' Xbox friends list and achievement points, which can also be earned by playing games on Windows Phone 7 (figure 22).

The stakes are high for Microsoft. Every indication is that mobile devices are poised to be as revolutionary as were desktop PCs. Microsoft has prospered for the past 20-odd years based on its dominating position in the PC software marketplace. You can argue that mobile is still in early days, but Microsoft is to date playing also-ran. It's also had some embarrassing missteps. Microsoft's tardiness in developing a state-of-the-art smartphone OS saw its most faithful handset partner, HTC, move strongly into the development of Android phones. An earlier attempt at a consumer-friendly, social-networking-focused phone, the Kin, was a dramatic failure. Released to great marketing fanfare, the Kin One and Kin Two were abject sales failures, prompting *NYTimes.com* to title a page on its website "A Youthful Market Spurns the Wares of Microsoft."<sup>11</sup> Only six weeks after the Kin launched, Microsoft killed the product altogether.

Windows Phone 7 is in its infancy, but there is every indication that Microsoft is making a serious run at the consumer smartphone space with these devices, and that it's doing things right so far. You could argue that Microsoft is late to the app-enabled smartphone



**Figure 22**

Gaming on a Windows 7 phone (Photo credit: Jeff Wilcox, "Windows Phone 7 Series," [www.flickr.com/photos/jeffwilcox/4435367457](http://www.flickr.com/photos/jeffwilcox/4435367457), licensed under the Creative Commons Attribution 2.0 Generic license, <http://creativecommons.org/licenses/by/2.0/deed.en>).

OS party, but seemingly it has taken the past couple of years to learn from some of its peers' mistakes.

Unlike the BlackBerry or webOS app stores, which have a very limited selection of available applications, or the Android Marketplace, which can seem overrun with worthless or even scam applications, Microsoft has taken steps to ensure that its store is front-loaded with quality applications. Despite the fact that Apple's hands-on and sometimes overly restrictive App Store approval process raised the ire of some developers, Microsoft has instituted a very similar process for its store. Additionally, Microsoft reportedly approached the developers of a number of popular iOS and Android applications with very generous bonuses for developing Windows Phone 7 applications for launch.

## Notes

1. Apple, "Apple's App Store Downloads Top 1.5 Billion in First Year" (press release), July 14, 2009, Apple Press Release Library, [www.apple.com/pr/library/2009/07/14apps.html](http://www.apple.com/pr/library/2009/07/14apps.html) (accessed Jan. 5, 2011).

2. Apple, "Apple's App Store Downloads Top Two Billion" (press release), Sept. 28, 2009, Apple Press Release Library, [www.apple.com/pr/library/2009/09/28appstore.html](http://www.apple.com/pr/library/2009/09/28appstore.html) (accessed Jan. 5, 2011).
3. Apple, "Statement by Apple on App Store Review Guidelines" (press release), Sept. 9, 2010, Apple Press Release Library, [www.apple.com/pr/library/2010/09/09statement.html](http://www.apple.com/pr/library/2010/09/09statement.html) (accessed Jan. 5, 2011).
4. Ellyssa Kroski, "On the Move with the Mobile Web: Libraries and Mobile Technologies," *Library Technology Reports* 44, no. 5 (July 2008).
5. "Frequently Asked Questions," Android Open Source Project website, <http://source.android.com/faqs.html> (accessed Jan. 4, 2011).
6. Microsoft News Center, "Microsoft Announces Patent Agreement with HTC" (press release), April 27, 2010, [www.microsoft.com/Presspass/press/2010/apr10/04-27MSHTCPR.msp](http://www.microsoft.com/Presspass/press/2010/apr10/04-27MSHTCPR.msp) (accessed Jan. 4, 2011).
7. Steve Ballmer, quoted in Nick Wingfield, "Ballmer Aims to Overcome Mobile Missteps," Oct. 3, 2010, *WSJ.com*, <http://online.wsj.com/article/SB10001424052748703466104575529861668829040.html> (accessed Jan. 4, 2011).
8. Toby Padilla, "Android Ecosystem Infoporn Overload," *TweetDeck* blog, Oct. 12, 2010, <http://blog.tweetdeck.com/android-ecosystem> (accessed Jan. 4, 2011).
9. *Ibid.*
10. Millennial Media, *State of the Apps Industry Snapshot* (Baltimore, MD: Millennial Media, Oct. 2010), 2, [www.millennialmedia.com/research/stateoftheapps](http://www.millennialmedia.com/research/stateoftheapps) (accessed Jan. 4, 2011).
11. Ashlee Vance, "Microsoft Calling. Anyone There?" July 4, 2010, *NYTimes.com*, [www.nytimes.com/2010/07/05/technology/05soft.html](http://www.nytimes.com/2010/07/05/technology/05soft.html) (accessed Jan. 4, 2011).