

Native App or Web Site? Deciding Your Next Step in Mobile

MOBILE COMPUTING

**Software & Information
Industry Association**
Software Division



DEVELOPED BY THE SOFTWARE DIVISION OF THE
SOFTWARE & INFORMATION INDUSTRY ASSOCIATION (SIIA)

Copyright © 2011. All rights reserved.

Table of Contents

Introduction	3
Understanding Your Business and Customers / Users	4
Considerations	6
Market / user segmentation	7
Development and maintenance cost	7
Distribution and channel support	7
User experience and expectations	8
Off-line operation	9
Time to market	9
Development skills and capabilities	9
Data security	10
Technical product management	10
Usage analytics	11
Use Cases	12
Summary and What the Future Holds	13
Company Descriptions	14
Additional Resources	15

Native App or Web Site? Deciding Your Next Step in Mobile

Authored by
Paul Moceris, Deloitte
David Smud, Deloitte
Daniel Vitulich, Deloitte
Nolan Wright, Appcelerator

Introduction

Mobile is top-of-mind for companies of all sizes these days. In Q4 2010, IDC¹ reported that the number of mobile devices sold had exceeded the number of PCs sold for the first time in history. Today's users are already accessing web services through native apps and mobile browsers, and competition amongst free services and applications is heating up. Nearly everything available through desktop environments, from banking to games to CRM, has begun the transition. Understanding your users' wide variety of behaviors, preferences and capabilities with their mobile devices is necessary to build a solution designed for your market. Factors such as functionality, storage, distribution, security and others must play a role in deciding your route to market. While bridging these factors with your options for deployment may seem like an impossible task, there is a method to this mobile dilemma.

Mobile Web Sites

Browsers and data plans are ubiquitous with today's smartphones, and users are browsing the web on mobile devices more than ever before. In order to meet this demand, companies are creating mobile web sites optimized for smaller screens using their existing web content and development teams. This single mobile web site can be used by most devices, regardless of the brand or form factor of the phone.

The App Tidalwave

Thanks to Apple, many of your users and employees have come to expect "an app for that." By developing a native mobile app using the software development kits (SDKs) for iOS or Android, for example, you can create a feature-rich, engaging experience that utilizes the native features of your users' devices. In order to create this "native experience" the app must be downloaded to the device from an app store like iTunes or the Android Market, or from an enterprise app store behind a company's firewall. Native apps are written for a specific type of handset and mobile OS in order to take advantage of a phone's specific functions like the camera and GPS.

Development Options

When Apple launched their App Store in June 2008, there was only one option for getting in on the app game – write a native app in Objective-C using Apple's iOS SDK. This made iPhone development comparatively more difficult and likely more expensive than traditional web site development.

The arrival of Android devices made the development scenario even more complicated. Although Android adopted a more popular programming language, Java, developers had to create apps in two different languages if they wanted to reach users on both iPhone and Android.

By 2009, a new category of mobile development frameworks began to emerge to address the skill set and cross-device development issues. Companies like Appcelerator², PhoneGap³, and Rhomobile⁴ abstracted the native development languages (Objective-C and Java) and enabled developers to primarily use web development languages (JavaScript and HTML) to create native mobile applications. This opened up mobile app development up to millions of web developers, creating a huge and active ecosystem for mobile apps.

These frameworks also solve many of the problems associated with developing for multiple platforms by allowing a single code base to be published to multiple mobile platforms. Rather than having to write new code for each mobile OS, 80-90% of the code can be re-used from one device to another.

Understanding your Business and Customers / Users

Asking the question “Do I need a mobile web site or a native app?” is like asking, “Should I use email or IM?” It depends on the situation, and both probably have a place in your mobile strategy.

- **Do you engage your users heavily on the web today?** Companies for whom the web plays a strategic role in their business will want to have a mobile web site as a complement to their existing web site, since their users will most likely want to engage with them on their mobile devices as well as on their PCs.
- **What is your competition doing?** If they’re delivering a compelling native mobile app, you may need to have an app to continue to grow your business.
- **What are your executives demanding?** As the app tidal wave gains momentum, the corner office is increasingly asking, “Why isn’t THIS on my smartphone (or tablet)?”
- **Are there things that native mobile apps can do that could add value to your user engagement?** As you look to bring existing web experiences to mobile, explore if native features like geo-targeting or push notifications can add real value and productivity to the app.
- **Do you want to create a “stickier” mobile relationship with your user?** Users are now spending more time using mobile apps than they are surfing the web – and that’s surfing the web on a PC and mobile browser combined.

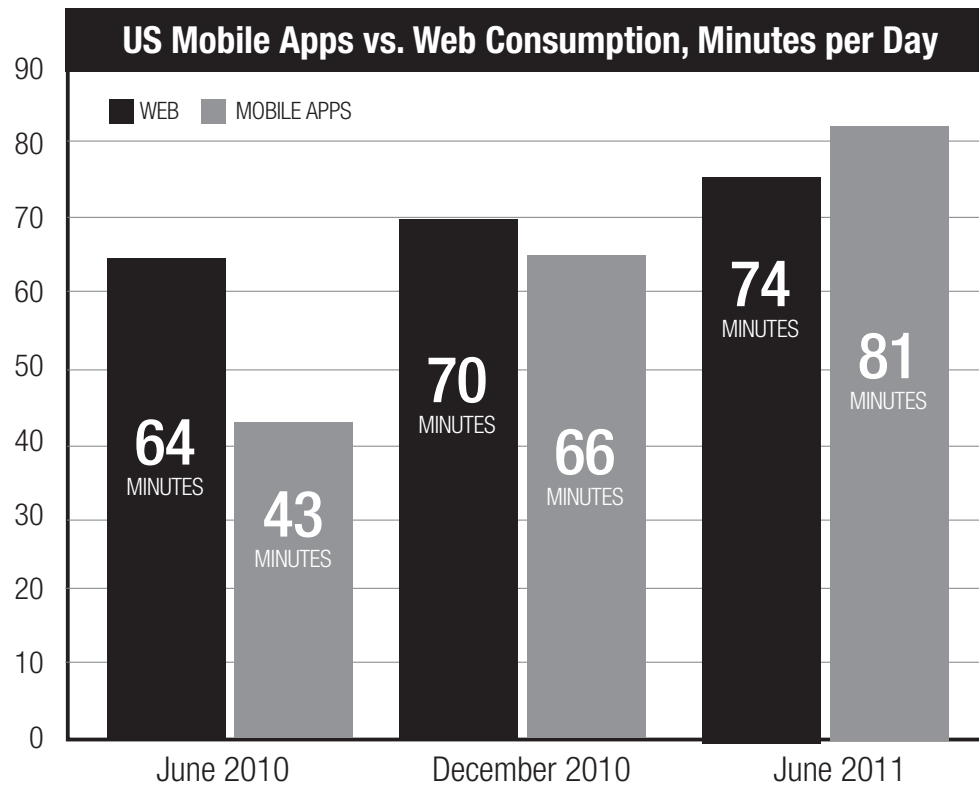


Figure 1⁵ Mobile analytics firm Flurry recently reported that users spend more time per day engaging with mobile apps than they do browsing the web on either a desktop or mobile device. This stat is even more remarkable if you consider that it took less than three years for native mobile apps to achieve this level of usage, driven primarily by the popularity of iOS and Android platforms.

Considerations

When looking to go mobile, organizations have different options for creating a mobile presence. The following table summarizes various considerations for making the decision between a native app and a mobile web approach.

Table 1: Considerations for deciding between native and Web mobile apps

	MOBILE WEB	NATIVE APP
Market / User Segmentation	Less tech-savvy users More diverse target platforms	App-centric users Specific platforms
Development and maintenance cost	Consolidated effort across platforms, driving lower overall cost Higher overall cost if developing with multiple native SDKs	Higher overall cost if developing with multiple native SDKs Similar cost to Web if using a cross-platform framework
Distribution and channel support	Over the web Updates automatically available	Through app stores Updates available upon installation by the user
User experience and expectations	Re-use of web-type services experience	“App” experience à la iPhone Ability to access native features of the phone
Off-line operation	Network connectivity required unless using HTML5	Can work either online or offline
Time to market	Fast time to market	Slower time to market if using multiple native SDKs (can be improved with cross-platform development tools)
Development skills and capabilities	High reuse of established web skills	If using native SDKs, specialized skills are required (Objective-C, Java, etc.)
Data security	Can leverage existing web security	May be able to leverage existing web security Requires a strategy and tools to secure data stored on the device
Technical product management	Similar to Web product management	If using native SDKs, each app on each platform will have to be treated like a separate product line If using a cross-device platform, it will be similar to web product management
Usage analytics	Limited to existing web analytics solutions	Can combine web analytics with native app analytics solutions

Market / user segmentation

When setting out to decide between a native app and mobile web site, one of the first considerations should be your target audience. It is important to understand the demographics of your users, how they use their mobile devices, and the relative share of the mobile platforms they use.

Not all individuals use their devices in the same way. Some user groups may prefer to download and use apps extensively while others primarily use their mobile devices for traditional wireless services and web browsing.

The various platforms that your target demographic uses, and the relative popularity of those platforms, will also go a long way in determining the approach you will use in developing a mobile presence. A diverse user and platform base lends itself more naturally to a cross-platform web solution while a more select user base more easily embraces apps that are built for specific platforms.

Development and maintenance cost

At a minimum, the costs that should be considered include all of the tools and resources to develop your app for each target platform. This includes development environments, licensing fees, labor, training, and ongoing maintenance. Web development represents a familiar cost for many organizations. However, when developing native apps, especially for multiple platforms, many companies will face a learning curve for not only the technical skills necessary but also the ability to accurately forecast costs.

Cross-platform development frameworks can help bridge the cost gap between developing web and native apps since the app is developed once but then deployed across multiple platforms.

Distribution also has an impact on the total cost. When offering a native app through a public app store, the app store often bears the cost of hosting and distributing the app. If using a private distribution method, companies must factor in the cost of that capability. For mobile web sites, distribution costs come in the form of on-going web hosting for the site in addition to hosting of any back-end server components that can be common to both approaches. When using a cross-platform development framework, companies still need to consider the distribution cost for each platform even though the app is only developed once.

Finally, you should consider long-term maintenance. You should plan for fixes and enhancements for your app or site across all platforms that it runs on and the development, licensing, and distribution costs associated with that.

Distribution and channel support

New mobile apps must be distributed to end users and installed on their mobile devices, and -as with any software product- apps too need to get periodical updates and bug fixes. Understanding your target audience and segmentation, and the key implications of the different distribution and support strategies can play an important role in selecting the right app type.

When considering native apps, it may be necessary to utilize commercial “app stores” to reach end users. This is a digital storefront facility controlled by carriers or mobile device manufacturers where users browse, buy, and download apps, to then install them on their phones. Under this scenario, enterprises work with app stores to get their apps tested, approved and posted online, and then support them for the entire app lifecycle. Popular app stores may be very appealing for showcasing new apps, but the time it could take to make critical updates available through this channel to the end user may represent a serious problem.

While public app stores allow for broad distribution of apps, an internal enterprise distribution strategy may be required for internal or sensitive apps. This can be an internal Web server, or even a more complex mobile device management (MDM) or mobile application management (MAM) platform. These solutions provide full control to the enterprise on app and updates release, but it is important to understand that not all of them may take the responsibility away from the user to download and install these apps and updates. A distribution strategy and solution should therefore be carefully evaluated.

Finally, mobile web sites could be considered the simplest option from a distribution and support channel perspective. The end user is totally agnostic to app updates and bug fixes, as these are automatically available to the entire user population once the enterprise takes care of applying them in their application servers. Discovery of new sites, however, may lack the promotional benefits that the popular app stores present to a brand name.

User experience and expectations

The surge in mobile app use and development over the last few years has set high user expectations of app features and usability. Through the use of native software development kits (SDK), mobile apps are able to make full use of the features and hardware of the mobile device they run on. This includes OS user interface elements, gestures, various sensors and local storage. Native application SDKs also provide a more advanced way to process and display push notifications compared to traditional WAP Push or SMS as well as allow apps to access device accessories and external hardware.

Although upcoming standards such as HTML5 aim to provide richer feature sets for mobile web sites, in practice it is generally more difficult for web sites to take full advantage of advanced OS features and hardware components. Broader acceptance, wider implementation and decreased fragmentation of the draft HTML5 standard will reduce this gap over time.

When making the choice between a web site and a native app, it is important to define the features and functionality that your app will use both in the near-term and long-term: what features do your users expect; what features do your users need? For many enterprise uses, a mobile web site will provide sufficient functionality. Even if your long-term vision includes leveraging features only a native app has access to, it is worth exploring the idea of starting out with a mobile web site and then moving to a native app later, increasing the app’s functional richness along the way.

Off-line operation

Due to the nature of wireless signals, mobile devices are subject to experiencing connectivity instabilities such as speed variations and connection drops depending on their location. This is particularly important when considering “where” and “how” the user will be using the app.

While mobile web sites provide good cross-platform support and reduce the risks associated with storing sensitive data on the mobile device, unless using emerging technologies such as HTML5 for local data caching, they usually require always-on connectivity through a wireless network in order to be of any use. When experiencing bandwidth and throughput drops, the app transaction latency increases dramatically, sometimes causing it to time out. If a user is in the process of downloading / uploading data or completing an online transaction, the data will frequently be lost.

Data buffering and synchronization should be properly handled by the app in order to support off-line operation when wireless connectivity is unreliable or unavailable. Native apps are well-suited for supporting this type of operation.

When evaluating whether off-line operation is required, or possible, ask yourself where the user will be located; whether wireless connectivity service is reliable; how much data needs to be uploaded or downloaded; how frequently; what is the minimum required connection speed; if data can be buffered locally without causing data integrity problems; whether operation can be supported with locally stored data, etc. Asking the right questions will help you set a solid framework to support the decision process.

Time to market

A mobile presence has become table stakes for companies in many industries. Those who are not yet in the game look for a quick time to market when jumping in.

For many, a mobile web site will be the quicker way to launch a mobile presence. Many organizations have existing web sites and web capabilities to leverage in order to deploy a mobile web site quickly. Building native apps can be more time-consuming, especially if building for multiple platforms due to the learning curve for new technologies.

Leveraging cross-platform development frameworks can help cut down on the time to market when developing for multiple native platforms because the app is developed once. However, as with any approach for developing native apps, post-development steps such as distribution still need to be performed for each platform separately.

Development skills and capabilities

Organizations should examine existing capabilities when deciding what type of app to build. By drawing on existing skills, an organization can likely reduce costs and accelerate implementation. Even though the user experience (UX) design of a mobile web site can be fundamentally different from that of traditional web site, and a skill that not all companies will possess, many will find that the technical skills for developing mobile web sites are very similar to existing web skills. This typically includes the necessary know-how to overcome subtle differences in browsers’ implementations of web standards.

Several of the cross-platform mobile development frameworks also allow you to leverage web technologies to write apps for multiple platforms. Apps are written using web languages and are then compiled into multiple native, standalone apps that can run on specific platforms.

In contrast, the development skills for native apps tend to be more specialized – requiring development in languages such as Objective-C or Java or unique Integrated Development Environments (IDE) such as Xcode or Eclipse. Some organizations will have these capabilities in-house. However, it becomes less likely that you will already have all the necessary skills as your number of target platforms increases. The good news is that the leading platforms provide substantial support and tools to help developers accelerate their learning curve.

Data security

Mobile devices are vulnerable to loss or theft, and as with any device connected to the network, they are also vulnerable to malicious code and other attacks. Both native apps and mobile web sites should implement authentication, access control and data encryption to protect sensitive business and personal data stored on or accessed from the device.

Due to their nature, mobile web sites which extend existing Internet facing services to support mobile devices can typically leverage new and existing security mechanisms such as SSL VPN access, multi-factor authentication and single sign-on (SSO) to internal and third-party web services.

While supporting off-line operation, mobile devices running native business apps will inevitably contain sensitive or critical data locally stored on the device. Securing this information is the responsibility of the app. Mechanisms such as password protection, data encryption, administrative rights controls (system settings, permissions to directly install apps) and managing access to back-end services should therefore be part of the app security framework. This can typically be complemented with the use of mobile device management tools (MDM) that allow for device tracking, usage reporting and remote locks or wipes for lost or stolen devices.

Extending new data, transactions, and services to mobile devices must be carefully managed from a security perspective. Consider the following questions when selecting the app type that's right for you: What data does the mobile device need to access? Is this information sensitive? Is it required for this data to be stored locally? What threats is the data being transmitted exposed to? Will existing browser-based security mechanisms protect it or will others be required? What access control and authentication mechanisms should be provided to secure local and remote data?

Technical product management

Product management policies become a requirement as organizations introduce apps to the market. Users will expect to receive frequent updates and patches or bug fixes to their existing apps, so the organization has to consider the implications brought along by the different app architectures and the tools to support their lifecycle.

Those familiar with web product management disciplines will find it is relatively easy to follow a similar approach to support mobile web sites, which are in nature very similar from a technology point of view. However, expectations are for continuously evolving features and capabilities, not just content. This can be a stretch for even the best existing web teams.

Native apps, however, are subject to very different considerations. For example, when supporting multiple device platforms, and using multiple native SDKs, each app needs to be treated as a separate product. This implies that each product might have its own feature roadmap, probably a different distribution channel and support strategy. However, some cross-platform development offerings currently available in the market may help narrow this gap. This option should be taken into account when considering native apps.

Usage analytics

Gathering key information from your users will help you learn what they like, what features they prefer, and where they usually go. How your app performs, and what results your marketing campaigns are bringing should be at the top of your mind when considering the right type of app for your company.

Both native apps and mobile web sites offer the possibility to gather this type of information. However, while mobile web sites can easily leverage existing web server analytics services, they do not necessarily offer the same richness of information that native apps can provide. There are solutions available in the market that, by simply adding a few lines of code to the native app, take analytics one step beyond traditional web analytics. By leveraging device built-in capabilities such as GPS, the platform can gather information about the user's location, roaming patterns, and interaction with your app. The possibility to gather information while the user is offline is another important point where native apps present a clear advantage.

When assessing your options, consider what usage information is important for your organization and how simple or complex it would be to gather it.

Use Cases

Native applications and mobile web sites can provide similar features and experiences, but there are some key differences in the benefits and challenges to each approach. From our experience and observations, the use cases defined below illustrate the reasons why you would choose one over the other based on your solution's purpose and your development capabilities.

OVERALL	ADVANTAGES	DISADVANTAGES
Mobile Web Sites: Quicker development with broadest reach Better for consumption of information (text feeds, YouTube, etc.) Light user interaction Not device-specific experience	Broad reach with less development effort Leverage web developers and existing investment in web infrastructure Fast time to market Leverage web security	Inferior user experience Limited access to device- specific capabilities Requires network access for usage (no off-line use available)
Native Applications: Deeper, stickier, better user experience Richer user interface Access to native capabilities of the device (camera, user interaction, geo-location, etc.)	Better user experience Deeper level of cuser engagement Huge momentum and user demand Online and offline capabilities	Specialized development resources needed (Objective-C and Java developers) Longer time to market due to development requirements No code reuse if from one OS to another

Summary and What the Future Holds

Many technology executives recognize their enterprise has operational problems that demand mobile solutions, but they are reluctant to build in-house capabilities or engage vendors to deploy native applications for each type of mobile device used across the company. Mobile web sites are simpler and quicker to create, but current technology doesn't provide the rich functionality that's possible with today's customized native applications. Many are stymied by their indecision.

Rather than focusing on the technology, consider the problem that needs to be solved or the opportunity that's there for the taking. If you can do the job with a web-based app, by all means, go for it. But if there's a strong business case for investing in the rich functionality and user experience only available in native apps, go that route. It's totally acceptable for an enterprise to have a mix of web-based and native mobile applications.

Ultimately, regardless of approach, you should think through personas and usage scenarios in the design of your app. Attempting to port or even optimize what historically was a point-and-click experience on a desktop or laptop screen will almost always underserve. Take advantage of underlying web content and services, but design specifically with a mobile mindset – considering the form factor, context of how interaction will be occurring, and potential for how business could be done untethered from the enterprise. From the user down, not the system up. See Twitter and Pandora as examples – where public web sites were significantly redesigned to approximate the usability and features of their native app counterparts.

Meanwhile, keep an eye on the new web standards that are being developed. Some, like HTML5, combine the best of both worlds – increased functionality that can be easily accessed by all mobile devices. It is possible that web-based applications will eventually overthrow native applications, but we're not there yet. Meanwhile, focus on creating business value by solving today's problems with today's tools – using whatever gets the job done.

Company Descriptions

Appcelerator

Appcelerator is the leading enterprise-grade, cross-platform development solution on the market today, with over 1.5 million developers using its software to power more than 21,000 cloud-connected mobile, desktop, and web applications used by tens of millions of users every day. The company's flagship offering, Appcelerator Titanium, is the only open source platform to enable fully native, cross-platform development, from a single codebase for iOS, Android, and Blackberry. Appcelerator's customers include NBC, PayPal, eBay, Merck, Medtronic, Harrah's, ZipCar, LA Times, and Cisco. Try Titanium for free at www.appcelerator.com.

Deloitte

In the United States, Deloitte LLP and its subsidiaries have 45,000 professionals with a single focus: serving our clients and helping them solve their toughest problems. We work in four key business areas — audit, financial advisory, tax and consulting — but our real strength comes from combining the talents of those groups to address clients' needs. Fortune and BusinessWeek consistently rank our organization among the best places to work, which is good news for our talent and our clients alike. When the best people tackle the most compelling challenges, everyone wins.

Additional Resources

Deloitte Mobility

<http://www.deloitte.com/us/mobileapps>

“The Mobile Dilemma: Web vs. Native Application Development”

http://sii.net/index.php?option=com_docman&task=doc_download&gid=2758&Itemid=318

Endnotes

1. <http://www.idc.com/about/viewpressrelease.jsp?containerId=prUS22689111§ionId=null&elementId=null&pageType=SYNOPSIS>
2. <http://www.appcelerator.com/>
3. <http://www.phonegap.com/>
4. <http://rhomobile.com/>
5. <http://blog.flurry.com/bid/63907/Mobile-Apps-Put-the-Web-in-Their-Rear-view-Mirror>.