

Navigating the Mobile App Development Landscape

You keep hearing about user trends towards mobile devices; your 10-year old knows your iPad better than you, and so you figure that your business should also get on board and have its own mobile app. Of course, if your company primarily interacts with a niche audience mostly offline, it's quite possible that you don't really need a mobile app. If you make the caps that go on toothpaste tubes, I'm not sure an app would do all that much for you.

As more and more people start to access the internet from their mobile devices, many businesses find that having a mobile app is a great way of connecting them with their audience.

Case in point – when I make a reservation at my favorite restaurant, do I look up their number and call? Nope, I hop on their mobile app, see what times are available, and book my table. I can also see how many people they have on their wait-list. Now, if only the app allowed me to pre-order a glass of Merlot to be ready for me when I got there!

As a business considering hiring a company to build a mobile app, the landscape can be confusing, and sometimes even scary – it can often mean a huge investment of time and money. You hear talk of iOS, Android, Symbian, webOS, and people ask you if you're going native or with a Web App. Going native? No, it doesn't mean sporting a loincloth, facepaint, and a spear... it refers, rather, to a fundamental decision on which direction to take your mobile app development. And there is no right answer – rather pros and cons to each path which you will need to evaluate against your specific business needs.

Let Me Count the Ways...

It is often assumed that there are two ways to build for mobile devices: web apps or native apps. In fact there are more like four:

Native Apps

Native refers to building an app in a device's native programming language. For iOS devices this means Objective-C, and for Android it's Java. Native apps are typically fast, reliable, and can access all the the device's hardware (camera, accelerometer, compass, etc). Because of this increased performance, mobile games are usually built as native apps. It also means, though, that your app is tied to the platform it is built for. An iOS app, for example, won't run on an Android device without first re-coding the entire app to Java.

Web Apps

A web app is a website that you access from your device's browser, but the site is made to resemble an application rather than a traditional website. It's purpose is also more functional – it offers a utility or service rather than a straight-up website which is often more informational. A web app can be accessed by any mobile device that has a browser. Though being browser-based, it's typical that not all of the device's hardware features can be tapped into. To produce a more engaging and interactive experience. HTML5, CSS3 and JavaScript are increasing being used to take advantage of the advanced features offered by this new specification.

Platform-specific Web App

This is a web app which is specifically designed for certain mobile devices, such as smartphones running iOS or Android. Less focus is placed on how the app looks on all mobile devices, and emphasis is placed on making the web app look like a native app as much as possible.

Hybrid App

Some argue that a hybrid app is a native app, but it's important to make a distinction. A hybrid app is built using web technologies, and then wrapped in a platform-specific shell that allows the it to be installed just

like a native app. Thus it is sold/accessed through the device's app store.

PhoneGap is an example of a framework that allows you take a web app and turn it into a native app for iOS, Android, BlackBerry, Windows 7, WebOS, Symbian and more. The hybrid frameworks typically have APIs as well, that allow you to access the device's hardware and features that are locked out from the browser.

A No-Brainer?

At first glance you might be thinking hybrid is the way to go. After all, it seems to offer the best of both worlds, or at least the best compromise between development costs and distribution; a single foundational codebase across multiple platforms. Alas, it's not so simple. Let's break down some of the pros and cons for native, web, and hybrid apps to help you make the right choice. The direction you take should be more about matching the technology with your business requirements.

Native – When Speed Matters

Pros:

- Better performance (at least for now), snappier animations, transitions, and faster load times. The performance difference between native and web apps is far more pronounced on slower devices (e.g. iPhone 3G running iOS4)

- Can store more data offline
- Can be featured and searched for in the app store
- Full access to the device's hardware and OS features
- Implicit installation of an app to the device's home screen. On iOS devices you can add any web app to your home screen, but it's a manual process

- The App Store handles purchase transactions on your behalf

Cons:

- Typically more expensive to build, even for a single platform. Build costs increase significantly for each new platform.

- Because the codebase needs to be re-worked for each OS, the time

to build an app for multiple devices can also be quite involved.

- Your app must be accessed through the device's app store, which has two important considerations: your app must go through an approval process, which can be lengthy and arbitrary, and if your app generates revenue you must share a percentage with the store (30% for Apple's App Store, including in-app purchases). App updates must go through a new approval process each time.

Web – Write Once, Run Everywhere

Pros:

- A single codebase which can be accessed by any browser-enabled mobile device.
- Uses web technologies (HTML/CSS/Javascript), which are arguably easier to learn than native languages like Objective-C or Java.
- Performance issues are becoming less of an issue as mobile browsers become faster and their Javascript engines keep improving
- No approval process needed, and updates to the app can happen instantaneously
- No revenue sharing with an app store

Cons:

- Using web technologies means interpreted code (as opposed to compiled code for native apps), which some people believe means web apps will always be slower than native apps.
- Don't have full access to all the methods exposed by the device's operating system, meaning you are limited to the APIs made available by the browser. As it stands now in Mobile Safari, this means no camera, compass, video capture, microphone, user contacts, file uploading or push/local notifications.
- Can't be found on the app store. If you're lucky enough to be a featured app in Apple's store, for example, it is a huge marketing boost.
- If you are looking to generate revenue, it's up to you to build a commerce model.

Hybrid – Fully Loaded with Better Fuel Economy

Pros:

- Allows you to capitalize on the single codebase offered by web technologies yet still market your killer app in each of the major mobile app stores.
- Gives you APIs to access some, if not all, of the features locked out of the browser, such as camera, compass, contacts. For reference, here's a list of [PhoneGap's supported features](#)
- Purchases are managed by the App Store.

Cons:

- Still subject to the store's approval process and revenue sharing. No instant updating.
- The app's performance is still dependant on the device's browser capabilities.

An Informed Decision

As you can see from the above list of pros and cons, the decision of which technology to go with is often dependent on what your app will need to do. Be clear on what your business objectives are, and see which method serves them best. Right now, given the state of web app performance, if you are building a game with fancy graphics, you're probably going to need to go native. If your app is on the simpler, like an app that reminds people of their anniversary (and scolds you if you forget), then perhaps a web app is what you need.

Back to the Future

In my line of work, I'm always looking towards the future, and my crystal ball is telling me very good things about web apps and the expanding capabilities of HTML5 technologies. There are already some great examples of HTML5 based games which produce experiences that were previously only achievable using Flash, see [Pirates Love Daisies](#). The Financial Times recently pulled their native app from Apple's store

and opted instead for an **HTML5-based app**. HTML5 brings things like video, audio, transitions, transforms, animations, offline storage, and drawing APIs direct to the browser... things that previously were only possible with plugin technologies like Flash and Silverlight.

Building web apps isn't easy though. It still requires great knowledge of HTML, CSS and Javascript. Javascript is a very robust, efficient, and capable language, and does most of the heavy lifting in web apps (e.g. code logic, animation, database integration, interactivity, etc). And to make developing sophisticated web apps easier, a number of Javascript frameworks have cropped up recently, some of which even employ an MVC (model-view-controller) architecture to really make you feel like a software engineer.

After evaluating a number of these new frameworks, I have recently gotten behind one called **SproutCore**. SproutCore is an open-source framework that leverages HTML5, uses a nice and clean MVC architecture, and is designed around creating fast and responsive web apps that rival the performance found in native applications.

SproutCore achieves this by moving all of the business logic to the browser, thus avoiding the bandwidth bottlenecks and network latency that can happen in more traditional web apps that are built around back and forth communication between the client and the server.

Mobile Apps for Everyone!

This is an exciting time for mobile and web developers. While I don't foresee native apps being completely replaced by web apps, I believe that we will continue to see an increasing number of businesses look to web apps for a more cost and time effective solution when considering mobile app development. And who knows, maybe your toothpaste cap manufacturing company could use a mobile app after all!