



How to Choose the Right Architecture For Your Mobile Application

Mobile Applications can Sell Products & Services, Raise Productivity, and Increase Awareness of Your Brand

Mobile technology is revolutionizing the way we shop, work and stay informed. In the U.S. nearly every adult now has a mobile phone, and almost one in three has a smart phone—devices almost as powerful as traditional laptops. This explosive growth is being repeated in many other countries.

The powerful capabilities of today's mobile devices provide fertile ground for countless exciting applications. Although demand is highest for consumer apps, new business applications are also emerging constantly to allow employees and partners to work more productively. For any enterprise that wants to engage, inform and assist its customers or employees, mobile applications create opportunities that are impossible to ignore.

While many companies would love to reach a mobile audience, they're often uncertain about how to proceed in this incredibly dynamic environment. This paper addresses the most important decision they'll have to make: the development architecture that provides the foundation for a mobile application. Do you choose native development, a cross-mobile platform, or a mobile web solution? Each approach has its strengths, weaknesses and appropriate use cases—how do you sort through the different parameters to make the right choice? There's also a variety of development frameworks and tools for each approach. How do you choose the right one?

As the paper guides you through these key issues, it draws upon RapidValue's experience helping companies succeed in the rapidly evolving mobile landscape. It examines the requirements you need to consider, the key characteristics of each approach, and what factors go into making a decision. The paper also examines some real-life examples that put the concepts described here into practice.



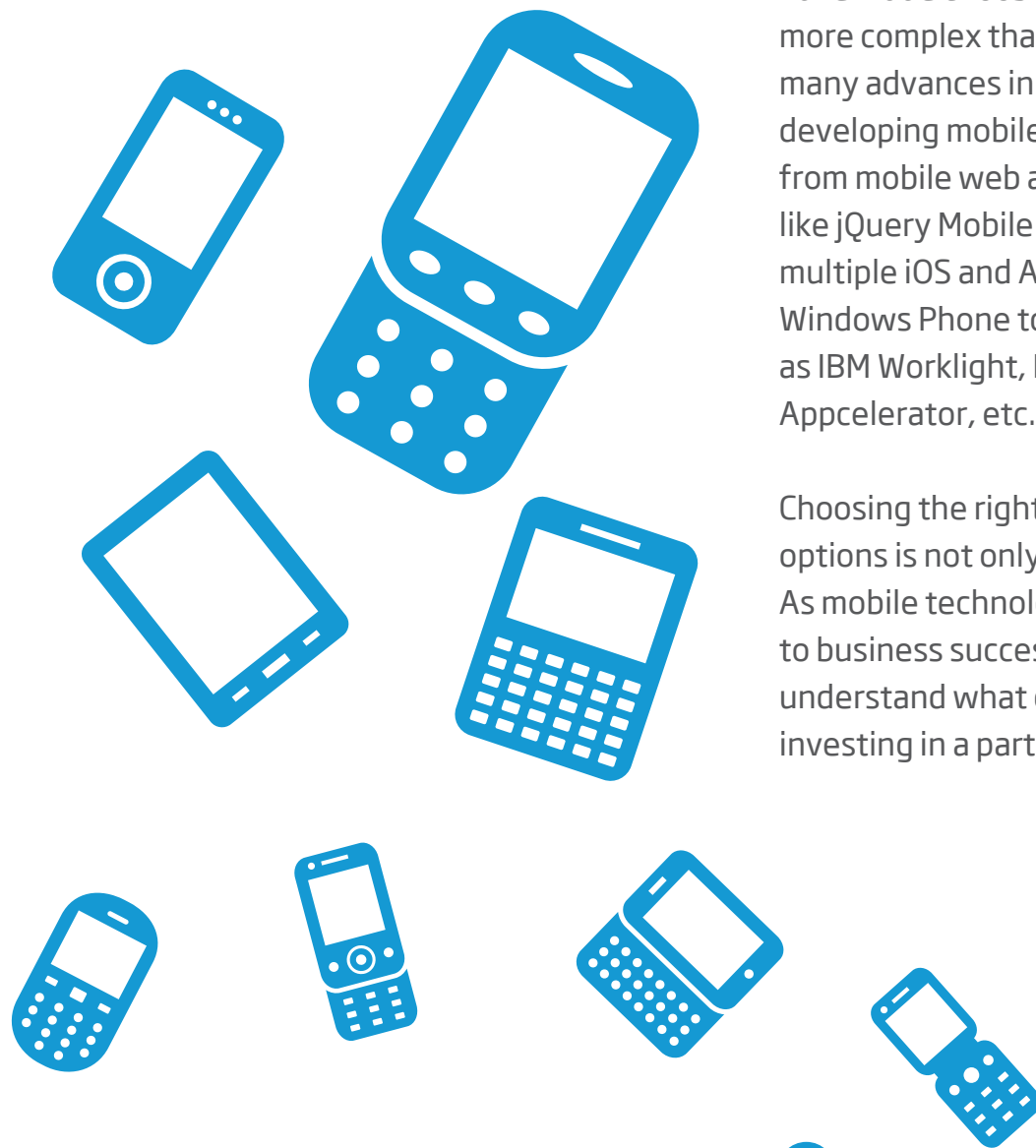
Navigating a Complex Landscape

Unlike the standard desktop environment of Windows PCs, the mobile marketplace is characterized by a variety of operating systems and device types, all evolving at different rates.

Within this volatile environment, success depends on quickly delivering applications with the quality and features that users demand, across a wide range of screen sizes, presentation formats and device capabilities. How to navigate through this varied landscape has long been a challenge for mobile developers as well as CIOs and the people who advise them.

The rapid proliferation of smart phones and release of new tablets (including iPad3, iPad Mini, Amazon Kindle Fire, Google Nexus 7, and Microsoft Surface) have made choosing the right mobility solution more complex than ever. There have also been many advances in the technologies used for developing mobile applications. Choices now range from mobile web apps using HTML5 or frameworks like jQuery Mobile or Sencha to native apps for multiple iOS and Android versions, BlackBerry, and Windows Phone to cross-platform frameworks such as IBM Worklight, PhoneGap, RhoMobile, Titanium Appcelerator, etc.

Choosing the right mobile architecture from all the options is not only difficult; the stakes can be huge. As mobile technology becomes increasingly crucial to business success, decision makers need to understand what capabilities are important before investing in a particular approach.



The Key Technology Choices for Mobile Applications



While many technology options are available for mobile applications development, it's possible to simplify the lay of the land. Mobile applications can be developed in three fundamental ways:

Native Apps

Build the application in its native code to run on a specific device and operating system. The app is downloaded from an app store and resides on the device.



Cross-Platform Apps

Build the application once using a cross-platform framework and run it on multiple mobile platforms (after fine-tuning for each platform). It's also downloaded from an app store and resides on the device.



Mobile Web Apps

Build the application to run over the internet from a central server, from where it can be accessed by any device with a web browser.



Due to differences in underlying technology, each approach has inherent advantages and drawbacks, development frameworks, and appropriate use cases. Careful analysis is required to ensure that an application is built using the right technology for the functionality required. The following table summarizes the different approaches:

Development Approach	Native	Cross-Mobile Platforms	Mobile Web
Definition and Tools	Build the app using native frameworks: <ul style="list-style-type: none"> - iPhone SDK - Android SDK - Windows Phone SDK 	Build once, deploy on multiple platforms as native apps: <ul style="list-style-type: none"> - RhoMobile - Titanium Appcelerator - PhoneGap - Worklight - Etc. 	Build using web technologies: <ul style="list-style-type: none"> - HTML5 - Sencha - JQuery Mobile - Etc.
Underlying Technology	<ul style="list-style-type: none"> - iPhone: Objective C - Android: Java - Windows Phone: .NET 	<ul style="list-style-type: none"> - RhoMobile: Ruby on Rails - Appcelerator: Javascript, HTML - PhoneGap: Javascript, HTML - Worklight: Javascript, HTML 	<ul style="list-style-type: none"> - Javascript, HTML
Deployment	App stores	App stores	Over the web
Key Use Cases	<ul style="list-style-type: none"> - Apps requiring high-end user experience, more transactional in nature - Large user base on one device (e.g. physicians with iPad) - Offline usage - Apps requiring extensive device and/or OS functions 	<ul style="list-style-type: none"> - Simpler apps, more informational in nature - Offline Usage - Multiple device types distributed across key users - Works well for a number of enterprise applications that do not require heavy device functions 	<ul style="list-style-type: none"> - Generic user experience, performance depends on Internet connection - Distributed user base across smart phone platforms - Need to maintain single code base - Moderate amount of device functions

Native Development: The best experience, but at higher cost

Despite the maturing of other approaches, applications developed in a device's native code continue to provide the richest, most compelling user experience. Native apps are optimized for the operating environment of the device they run on, and can fully leverage device hardware (GPS, camera, graphics) as well as software (calendar, contacts, file system, etc.). Another key advantage is the ability to run offline since the application remains installed on the device after the original download. Data transfers can resume (and re-sync with back-end apps) when the connection is restored.

Access to some device features may require an app to be written in the native environment. While apps built using multi-platform frameworks or web apps can access some device features, native apps inherently provide full access and performance. This is particularly true on the latest versions of a device equipped with new technology that only native apps can support.

The chief disadvantage to native development is cost:

- The app typically has to be re-built for each native operating system. About 75-80% of the cost to develop for the first platform needs to be budgeted to build the second platform¹. The 20-25% savings comes from leveraging the user experience (UX) design and integration work on the original platform that can be carried over to additional platforms. (Developing BlackBerry apps can incur up to 150% of the development effort of other platforms due to the number of operating system versions and devices involved.)
- Development in multiple languages requires different skill sets (for example, iOS is Objective C-based while Android is Java-based). This makes development using in-house resources extremely challenging. People either need to be cross-trained on multiple technologies (which takes time and rarely achieves the quality required) or additional staff need to be hired specializing in different platforms.
- Keeping the application up to date means additional work, requiring development, testing, and distribution each time a new OS release or device model becomes available. For example, when iOS6 was released in September 2012, developers had to scramble to update application code to ensure their apps would work properly on the new version.

Cross-platform Frameworks: Economies of scale, but at the expense of user experience

Cross-platform frameworks like Appcelerator, PhoneGap, RhoMobile and IBM Worklight² are designed to save time and costs by allowing developers to write an application once in a language they know and use the framework to adapt it for multiple platforms. Most of these frameworks use web-based technologies like Javascript and HTML which in-house developers are often already familiar with. (An exception is RhoMobile, which is based on Ruby on Rails, which also has a large developer base.)

¹ This figure applies to development on the client side; any development on the server side can generally be re-used.

² This is by no means a comprehensive list of cross-platform frameworks; however, they are the market leaders and for the sake of simplicity we have only addressed four in this document.

Cross-platform frameworks have different strengths and weaknesses, with the right choice depending on the needs of the application and the business. The most important requirement is obviously whether the framework supports target devices and operating systems. However, you also need to consider how well the framework will adapt your application to the characteristics of each platform so that it runs at its best. Other criteria include what native or device features the framework supports, how much work is required to implement them, services and tools the framework comes with, and of course cost.

A few other things should be noted when using a cross-platform mobile development framework:

- **User experience:** Most cross-platforms frameworks cannot provide the same user experience as native applications. An exception is Appcelerator, which creates native apps for each supported platform once the base code is written. The typical user experience for apps built using PhoneGap, RhoMobile and others is closer to web apps than native apps even though the app is packaged in a native wrapper and deployed in an app store like a native app.
- **Development effort:** Even though these frameworks in theory facilitate “Build Once and Deploy on Multiple Devices”, in practice apps require tweaking for each platform. Typically, plan on 20-25% additional effort on each platform once the base app is built. Some device functions may already be supported by the cross-platform framework, so a careful analysis is needed to determine what additional effort is required when developing on multiple platforms. Most cross-platform frameworks let you build plug-ins for device features not supported by the platform itself. However, plug-ins need to be written for each platform, which adds to the overall effort.
- **Potential vendor lock-in:** Cross-platform frameworks take time to support new functionality that becomes available when a new OS version or device model is released. (There’s also no guarantee that it will be supported.) In contrast, native apps can leverage the new capabilities immediately.

Mobile web apps: widest reach, but at the expense of user experience and performance

The key advantage of mobile web apps over other development approaches is platform independence. With the ability to run on any smart phone or tablet equipped with a web browser, web apps are able to reach the largest audience with the least effort (although some device-specific customization is usually required). But mobile web apps have long had two distinct limitations: they can’t use the hardware and software on the device, and they require an Internet connection in order to work.

HTML5, the latest version of the HTML language used to deliver content over the web, is starting to overcome these limitations. By supporting on-board functionality, HTML5 will enable a user experience that's much closer to the look and feel of a native app. HTML5 web apps will also be able to run offline, just like a native application. However, HTML5 is a work in progress, and functionality is still somewhat limited compared to native and cross-platform apps³. Frameworks like Sencha and JQuery Mobile that leverage HTML5 enable developers to make their web application richer by providing the look and feel (and functionality at times) of native apps.

Advances in HTML5 technology are on-going, supporting more and more device functions to enable richer, more powerful web applications. Mobile web apps based on HTML5 are now mainstream—for example, the Financial Times launched an HTML5 app that reached a subscription of 1 million users within four months. The need for HTML5 has further increased higher now that Adobe has removed Flash support on mobile browsers, requiring a number of applications originally written in Flash to be re-written in HTML5.

There are several other things to keep in mind when developing mobile web apps:

- HTML5 currently lacks full support on Internet Explorer and most other browsers, which means that some features and functions of this browser may not work as expected.
- Even though it holds the promise of platform independence, apps built using HTML5 need to be optimized for different platforms. Typically, an additional effort of roughly 20-25% needs to be budgeted for each additional platform.
- HTML5 may need to be combined with frameworks like JQuery Mobile or Sencha in order to achieve the best results.

³A case in point: Facebook originally used HTML5 as the basis for its mobile web experience and its iOS and Android apps, thinking that it would be an easy way to develop for multiple platforms and screen sizes, and that the user experience would be similar to a native app. Unfortunately, the benefits of cross-platform development weren't enough to outweigh HTML5's current limitations and lack of stability. As a result, Facebook decided to jettison its HTML5-based offerings in favor of native apps.

A Decision Framework for Selecting the Right Technology

Key questions for identifying the right technology framework:

- Who will use the app?
- What type of experience do users expect?
- What functionality is required?
- How important is multi-platform compatibility?
- Does the app need to run offline or when the connection is bad?
- Are there industry-specific requirements?
- How much time do you have to build the app?
- Does your staff have the necessary skill sets?
- What's the total cost of ownership?
- How soon do you need new device or OS features?

Depending on your industry and business requirements, you need to consider a number of factors in order to identify the right technology framework. You need to understand the purpose of the application, who the users are, what capabilities they need, and what type of experience they expect. Other factors include security, portability to run the application across multiple platforms, and of course cost. Working through the following questions will guide you to the architecture most appropriate for your application:

Who will use the app?

Identifying the audience that your application will cater to is the first step in charting your mobile technology roadmap. Audience analysis will reveal what platforms need to be supported as well as the type of user experience you need to provide.



Platform requirements will be a key input into the technology framework selection. For example, an application for physicians might be built on the iPad as the primary platform, as about 75% of U.S. physicians use this device. On the other hand, an education application targeted towards working adults may be better suited for Android devices, as these are popular with this demographic. And general consumers primarily use iPhones and Androids, so these are the platforms you'll probably want to target first.

What type of experience do users expect?

The user experience is a key factor in determining the right technology framework. For consumer apps where “stickiness” is key, native apps are the logical choice because they can take full advantage of the capabilities of the mobile device to deliver the richest, most impressive experience. (Since there’s nothing to stop consumer users from looking elsewhere, you want to make sure you capture their attention.) On the other hand, for employee applications that need to run on multiple platforms but don’t necessarily require a compelling user experience, a cross-mobile platform or web app may be the right choice.

What functionality is required?

Identify the key functionality requirements, bucketed into informational, transactional and device function categories:

- A higher amount of informational requirements (where the user primarily consumes content) would point to cross-mobile or mobile-web based solutions.
- A higher amount of transactional requirements (where the user primarily interacts with the application to accomplish a task).
- Device-specific requirements (e.g. offline transaction entry or file access) will point towards native-based solutions or cross-mobile.

In addition, applications that require offline access of large files such as PDF catalogs are better built leveraging native.

How important is multi-platform compatibility?



Is there a primary platform you want to focus on for your application, or are they all equally important? Depending on the answer, it sometimes makes sense to follow different strategies in parallel. For example, if iOS is the primary platform requirement, it may make sense to build a native app for iPhone/iPad (providing the best experience) and an HTML5 app for all other platforms (providing a good but not great experience).

Does the app need to run offline or when the connection is bad?

This is another key part of the user experience. For consumers, it may not have to be the full application, but at least some basic capabilities should live on the user device so you can maintain some type of engagement with users while the device is offline. This also applies to enterprise users, who can't stop working just because the connection is down. Data captured on the mobile device can re-synchronize with the central server once the connection is restored.

Are there industry-specific requirements?



Sometimes industry requirements point you toward a particular solution. For example, in retail, discoverability is important, which will point to mobile web apps because they're much more SEO-friendly than native apps. Sometimes it makes sense to use both approaches in order to capture and retain an audience. For example, RapidValue is building a native app and a mobile web app for a large fashion company. The native app is aimed at existing customers, while new customers searching for "fashion apparel" on their mobile browser will have a much greater chance of finding a link to the web app.

Industry-specific technology choices that RapidValue sees regularly include:

- Retail: Native iPhone and Android apps provide the best experience for current customers; mobile web apps ensure visibility in web searches for new customers.
- Healthcare: Native iOS, as most physicians use iPads.
- Enterprise apps: HTML5 or PhoneGap, since the need is to reach the largest number of employee devices but not necessarily provide the greatest experience. Apps for top executives and those requiring a better user experience (such as enterprise dashboard apps) are developed in iOS.

How much time do you have to build the app?

If time is of the essence, a web app is the fastest way to reach the widest audience. You can then follow on later with native apps to provide the best experience on key platforms.

Does your staff have the necessary skill sets?



You also have to look at what skills your development team currently possesses and what skills will be needed to reach your user community. You then have to weigh the time and cost to train current staff vs. recruiting outside resources with the expertise you need. Native development requires greater knowledge of proprietary technologies such as Objective C for iOS. In contrast, most cross-platform and mobile web frameworks are based on Javascript and HTML which most enterprise developers already know.

What's the total cost of ownership?

The cost to build and manage a mobile application will vary widely depending on the development architecture. Mobile web apps are the most affordable approach since the same app can run on any device with a web browser (even if it may look rather “clunky”). They are also the least expensive to maintain.













Building native apps on multiple platforms normally costs the most, with each new platform adding 75-150% of the cost to develop for the first platform. In contrast, cross-platform and mobile web development add only about 20-25% of the original cost. Native apps that run on multiple platforms also add significantly to support and maintenance costs, as different skill sets are required on different platforms. Of course, the benefits of a compelling native application (customer loyalty, increased sales, etc.) may make the higher development and maintenance costs a very attractive investment!

How soon do you need new device or OS features?

One of the advantages of a native development environment is that it can immediately leverage new capabilities and features provided by OS upgrades or new generations of mobile devices. With cross-platform frameworks, you usually have to wait for the framework vendor to make the new features available. This lag time can range from days to months depending on the complexity of the upgrade and the priorities of the vendor. (Of course, if the application already fully meets your requirements, an urgent need to support the latest features is less of an issue.)

Comparison Summary

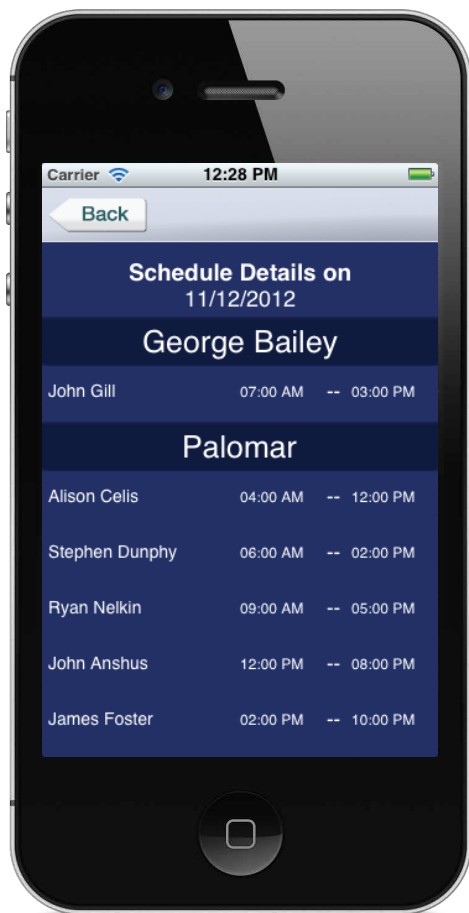
The following table summarizes the strengths and weaknesses of each development approach.

Criteria	Fully Native	Cross-Platform Frameworks	Mobile Web Apps	Comments
Cost of Development				<ul style="list-style-type: none"> - Cross-platform frameworks and mobile web apps reduce development costs, as the application can be built for multiple platforms. - Native apps require a separate development effort on each platform, with minimal reuse.
Multi-Platform Compatibility				<ul style="list-style-type: none"> - Mobile web apps provide maximum reach, running on any phone with a web browser. - Cross-platform frameworks reach all major platforms (but not always the same ones). Native features not included can be coded as plug-ins. - Native apps run only on the operating system they are developed for.
User Experience				<ul style="list-style-type: none"> - Native apps provide the best user experience. - Cross-platform frameworks may not give the best experience on all phones, with web-app look & feel in some cases. Scrolling and touch response are slower than native apps. - Web apps provide a generic experience that may look rather clumsy on some browsers (although this is changing with HTML5).
Features & Performance				<ul style="list-style-type: none"> - Native apps provide access to all device features and provide the best performance. - Apps developed using cross-platform frameworks can provide features comparable to native apps, but somewhat slower performance. - Mobile web apps have more limited capabilities and performance. For example, Push Notification alerts are supported in native/cross-platform apps but not mobile web apps.

Choosing a Mobile Development Strategy–Three Case Studies

The following case studies illustrate how RapidValue helps clients choose a development strategy for their mobile applications, and some of the thinking that goes into making an effective decision.

Case Study 1: Physician Communication Application on Native iOS and Android Platforms



RapidValue built a physician communication application for one of the leading physician practice management organizations in the United States. As part of the engagement, RapidValue conducted a technology strategy assessment, which included identifying users of the application and functionality required. The key findings:

- Primary users were physicians and other healthcare staff, who were accustomed to a rich experience on their mobile device. Engaging users was essential, and hence building the application on native platforms made the most sense. Since the application was built for phones, it was felt that the Android user base was too large to leave out—which pointed to native applications for iPhone and Android devices.
- Required functionality involved multiple communication channels including chat and VoIP phone calls, which were also more suited to a native framework.

RapidValue built the application on both device platforms within 18 weeks, including creating the user experience and graphics in coordination with physician users. The application was deployed in the next 4 weeks across a large user base of over 1000 physicians.

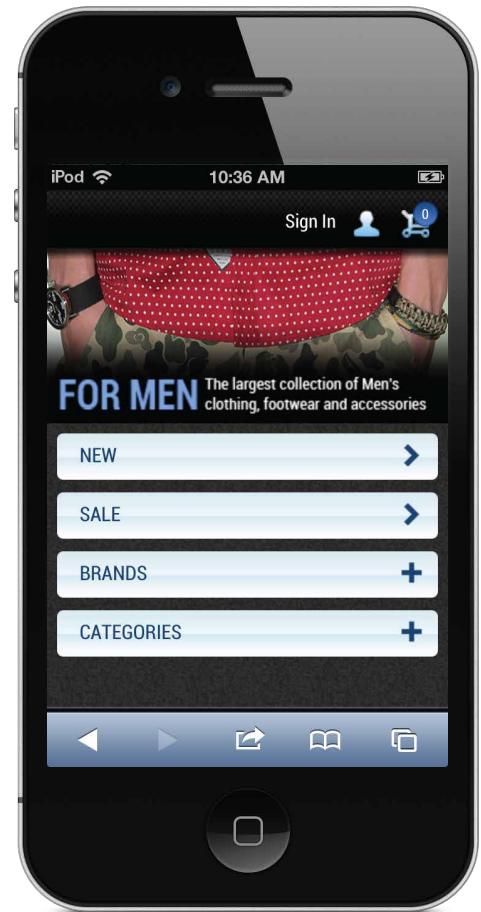
Case Study 2: Mobile Web-Based mCommerce Solution

RapidValue launched a Mobile Web-based mCommerce solution for the world's largest online retailer of street-wear. The Mobile Web service allows consumers to shop online using their mobile devices

The client wanted to add a new channel for their consumers. To meet client requirements RapidValue had to identify the optimum solution (responsive web vs. regular mobile web).

The client already had Android and iPhone native applications, however it was important to build the mobile web solution, in order to reach the largest number of consumers while providing service across multiple device types. Furthermore, native applications are not SEO- friendly and hence it was important to be present on the mobile web.

RapidValue designed the complete user interface, built and deployed the mobile service, all within 3.5 months.



Case Study 3: Cross-Platform Mobile CRM Application using PhoneGap

RapidValue engaged with a large industrial supplier to build a CRM and quoting application for its sales force. The requirement was to build a quoting application that would extend the client's Siebel CRM system and run on multiple smart-phones (iPhones, Androids and BlackBerry devices). Most regular CRM functionality (e.g. retrieving pricing information and entering line items on a quote) could be achieved using mobile web functionality, but there were requirements like GPS that could only be achieved with native device capabilities.

A decision was made to build the app using the PhoneGap framework because mobile application was required for multiple device platforms and functionalities were not mobile OS dependent. Although not familiar with native technologies, the in-house IT team was experienced with Javascript and CSS, two base technologies of PhoneGap. This made it easy to transition the app to internal support once development and deployment were complete.

Conclusion

Here are some final thoughts on choosing the right architecture for your mobile application:

- If cost is not a constraint, native apps provide the best features and performance.
- “Develop once and run anywhere” is an important concept, but there are different approaches for achieving it. A mobile web app provides almost universal coverage for a relatively modest development effort. Cross-platform frameworks can provide a richer experience while reaching the most strategically important mobile devices.
- If the requirement is to build iOS and Android apps while providing the best user experience, RapidValue recommends native apps. When development is limited to these two platforms, there may not be a cost advantage in using a cross-platform framework. If the requirement is to build for all platforms including BlackBerry and Windows Mobile, cross-platform development can provide cost benefits, although at the expense of usability.
- Native apps and web apps are not mutually exclusive; they can complement each other to help an organization engage its mobile customers and employees. For example, in retail, native apps provide current customers with the best possible experience while mobile web apps ensure visibility in web searches for new customers. While reaching the most strategically important mobile devices.
- Native and web apps will coexist for the foreseeable future. As HTML5 matures, mobile web apps will continue to take on richer, more native-like capabilities. For many apps, the economics of software development will increasingly favor HTML5 and related tools. However, native apps will remain the choice for engaging users with the richest experiences and the latest device functionality (at least for the next few years).
- Will there be much further consolidation? Probably not, at least for the next few years. Customer loyalties, fierce competition, unique technologies and lack of standards mean that multiple platform architectures and the need to develop for them will remain a fact of life for the foreseeable future.

RapidValue’s innovative thinking and industry knowledge make us a trusted resource for many of the world’s most dynamic and fastest-growing companies. We can help you identify a mobile strategy that best fits your requirements, and we have the skill sets and experience to bring it to life.

Our approach combines technical resources, design talent and understanding of your business to deliver mobile solutions that are not just compelling but also extremely cost-effective. Whether your goal is a rich, memorable consumer app or an application that improves the productivity of thousand of employees, give us a call and discover how RapidValue can make mobile work for you.

About RapidValue

RapidValue is a leading provider of mobility solutions to enterprises worldwide. Armed with a team of 150+ experts in mobility consulting and application development, along with experience delivering over 200 mobility projects, we offer a range of mobility services across industry verticals. RapidValue delivers its services to the world's top brands and Fortune 1000 companies, and has offices in the United States and India.



www.rapidvaluesolutions.com



www.rapidvaluesolutions.com/blog



+1 877.690.4844



contactus@rapidvaluesolutions.com