

March 2012

# Dealing with the Dilemma: Mobile App Development Approach & Decisions

By Mobility Practice  
Happiest Minds



## Overview

Today, mobile applications have become an integral part of nearly every organization's business strategy. With the right strategy, mobile applications can not only win new customers and revenue streams, but can also help in retaining existing customers and drive greater mobility adoption and penetration.

At the same time, due to extensive publicity, sales and branding undertaken through mobile channels, there is also a higher risk for a brand if a wrong mobility strategy or technology is adopted. Companies also have to cope with a fast-evolving environment characterized by rapid changes to operation platforms and releases along with a constant influx of new devices and form factors. Additionally, with multiple channels emerging for reaching out to mobile app users in terms of technology, platforms and User Experience (UX), companies are faced with the dilemma of making the right choice.

Since the mobile application user base is diverse, apart from various other business parameters, there are a few important questions that we need to answer before we decide on a Mobile Application Development strategy - *should the app be targeted across all available platforms? Do I need my app to access GPS, cameras and other device-specific functions? Do I need a uniform UX for the app across platforms? What is the EOL of the apps? Should I just stick to creating a mobile website?*

## **“One-Size Fits All” does not hold good for Mobile App Development. So what's the way out?**

Companies are trying to decide on what type of Mobile Applications they should deliver to their target users. This is a very critical decision that involves a trade-off between providing a rich, native user experience versus the portability of the application across platforms that will sacrifice the rich native UX. There are pros and cons to e-adopting both paths and each company have to evaluate this issue based on key parameters such as native experience, cost of development, time to market, ease of deployment, and managing the apps - to determine what the best choice is for them personally.

This paper discusses the following types of mobile app development and briefly outlines the advantages and disadvantages against each of them.

- Native mobile app development
- Mobile web app development
- Hybrid app development
- Cross compiled app development
- Virtual machine based app development

Based on the following evaluation criteria, organizations can make the best choice in adopting a mobile application development strategy.

### Criteria that can help you make the decision on choosing the right mobileapp development strategy

1. *Native UX*
2. *Portability*
3. *Time to market*
4. *Maintenance& management*
5. *Ease of deployment*

### Native Mobile App Development

Native mobile apps are developed using the native language of the mobile operating platform. These apps have access to all the device capabilities and functionalities as they use the native SDK for the app development. These applications provide the highest UX factor as compared to any other type of app development.

Advantages	Disadvantages
<ul style="list-style-type: none"> <li>✚ Maximum performance</li> <li>✚ Tighter integration with platforms</li> <li>✚ Ease of deployment due to app stores from native OS providers is very good</li> </ul>	<ul style="list-style-type: none"> <li>✚ No code reuse</li> <li>✚ Higher cost of development and maintenance</li> <li>✚ Increase in time to market</li> </ul>

### Mobile Web App Development

Mobile web apps are developed using mobile web framework technologies like HTML5, CSS3, and Java Scripts. These Apps are executed on the Web servers and accessible via Mobile Web browsers and are highly portable across multiple mobile platforms. However, they lack in creating a native, rich UX. Despite the fact that some of the device specific functions and offline stores can be accessed through HTML5, there are several constraints due to the dependencies of the sandbox nature of specific platforms and the extent of adoption of HTML5 specifications by native browser components that are still evolving.

A detailed Primer on HTML5 and its capabilities are provided in the Appendix.

Advantages	Disadvantages
<ul style="list-style-type: none"><li>✚ Better manageability due to web server based deployments</li><li>✚ Very good portability across platforms</li><li>✚ Higher ease of code maintainability and reuse</li><li>✚ Decrease in total cost of ownership</li><li>✚ Better time to market</li></ul>	<ul style="list-style-type: none"><li>✚ Lack of native UX</li><li>✚ Lower performance due to browser based dependencies</li><li>✚ Highly dependent on native browser implementation for access to device capabilities</li><li>✚ Unpredictable performance due to higher dependency on Internet connection</li></ul>

## Hybrid App Development

As the name suggests, Hybrid app development platforms use combination of both web based and native app development. Applications are built using web technologies like HTML5, CSS3, and Java Scripts and involve the native APIs for access to device specific capabilities. These apps are resident on the devices as they are packaged within the browser control of the platform and compiled using the native SDK. This type of app provides better portability across platforms as compared to native apps and they rely on the platform provider for hybrid platform development. Some of the hybrid platforms also provide flexibility to extend and customize the platform by adding additional wrapper plug-ins so that the apps can leverage plug-in extensions to access more specific native device capabilities to a limited extent.

Advantages	Disadvantages
<ul style="list-style-type: none"><li>✚ Access to limited native device capabilities</li><li>✚ Very good portability across platforms</li><li>✚ Good ease of deployment like Native apps</li><li>✚ Decrease in total cost of ownership</li><li>✚ Better time to market</li></ul>	<ul style="list-style-type: none"><li>✚ UX better than mobile web apps, but not on par with native UX</li><li>✚ Possible lower performance due to browser based dependencies and the extent of the web dependency</li><li>✚ Dependency on the hybrid platform provider capability for native API extensions</li></ul>

## Cross Compiled App Development

Applications created through the cross compiled app framework are deployed directly on devices. These apps are developed by leveraging cross compiler frameworks that transform the source code compiled into the native device specific platforms. These frameworks provide specific APIs that leverage deep native integration capabilities and the apps developed through this method have a rich user experience that closely mirrors the native UX. On the flip side, the portability of such apps is much lower when compared to hybrid or web based apps.

Advantages	Disadvantages
<ul style="list-style-type: none"><li>Richer UX like native apps</li><li>Leverage platform specific capabilities</li><li>Can be ported to multiple platforms though development in a single language</li><li>Deployment on the popular app stores ecosystems is high</li></ul>	<ul style="list-style-type: none"><li>The ability to develop the app is governed by the limitations of the platform APIs.</li><li>Certain native capability usage inhibits seamless portability of the app</li><li>Cross platform compilers are not available for all platforms and are limited to certain few</li></ul>

## Virtual Machine based App Development

Virtual machine base apps are developed with custom languages and they run on a virtual machine for each target devices or operating systems. App developers may not be required to write lot of code but have to make configuration changes that will allow the app to be executed over a virtual machine. The portability of these apps is much easier and the app success largely depends on the quality of the virtualization platform and the tooling that it exposes for app development.

Advantages	Disadvantages
<ul style="list-style-type: none"><li>Apps time to market is better due to sophisticated tooling for development</li><li>Apps can be upgraded through OTA(Over The Air) due to the virtual machine concept</li><li>Less maintenance of code and high code reuse</li></ul>	<ul style="list-style-type: none"><li>Poor native UX</li><li>Virtual machines may not be available across all platforms</li><li>May take a hit on performance due to runtime engine</li><li>No native customization possible</li></ul>

## An overall comparison of approaches

Criteria	Native App Development	Mobile Web App Development	Hybrid App Development	Cross-Complied App Development	Virtual Machine based App Development
User Experience					
Portability					
Time to Market					
Maintain & Manage					
Ease of Deployment					

## Making your decision on HTML5 vs. Hybrid vs. Native

While it is difficult to make a straight forward decision in selecting a tool, the following parameters should be taking into consideration along with business requirements:

- **Capabilities in handing use cases** – *What would be the application capabilities across other platforms? Will it have similar capabilities across platforms?* Unfortunately all platforms cannot handle application capabilities in a similar fashion. For example, iOS hands user notification and application states are different from those on Android.
- **User Experience** – *How do other platforms handle application views?* Views that are implemented in iOS may not be possible to implement in Android and/or Windows Mobile phone.
- **Security** – *Will uniform security features be implemented across all mobile platforms?* For example, platform security implementation in iOS is different from Android and RIM OS.
- **Synchronization** – *Some smartphone application need to contact the server to synchronize the data. What would be the application's data synchronization approach across all platforms?*

- **Performance** – *Will the application performance be compromised when implementing the solution across other platforms?*
- **Storage** – All smartphone applications provide APIs to store data in the device. The storage capability and storage style (locale, file system etc) should be analyzed thoroughly before building the application across other platform.
- **Marketing and Distribution** – *How will the application be distributed when available across multiple platforms?* While some applications are distributed via side-load, others can be distributed via the enterprise app store.

HTML5 will be the best fit to start the development process if performance and UI is compromised. This will allow organizations to create a quick GTM strategy and market presence. While this approach does away with the need install the application on the device, it does not enable consumers gain full access to application features without a data connection. However native applications are a best fit when handling offline application features like field force capturing sales orders, etc. For organizations that are looking to package websites and web applications as native apps, solutions like PhoneGap may be a good fit. This open source framework wraps existing applications to create a native app that leverages a web viewer component. Organizations can also take the web application route, where they can build an HTML5-driven application that I runs on most systems and devices. This can work if the application is very simple and leverages only the common denominator functionality of HTML5 that is supported across all browsers. The best path for an organization really depends on the team's expertise, budget and time-to-market requirements. While native development offers the most functionality and options, alternatives in the market like Titanium, QT and OpenPlug provide massive advantages in leveraging a team's existing skills to deliver an application that is faster-to-market and ultimately cheaper to build.

## Migrating a HTML5 web application to Mobile

Mobile web applications have become increasingly prevalent because of their ability to deliver a “personalized web experience”. By leveraging device features like location and cameras, it delivers relevant information users, thus enabling them to perform operations immediately.

A question asked often is - *how can we leverage existing desktop based websites and convert these into mobile versions in a quick and cost efficient manner?* Ideally organization should be able to provide a mobile version of their existing websites without changing a line of code.

However in reality, a website designed specifically for mobile will always provide a much better user experience to mobile users.

The two most popular approaches for mobilizing existing website are:

1. Screen scraping – This technique basically reads through the HTML and converts it into a mobile version by replacing tags. However this does not deliver an optimum user experience as this technique cannot leverage the device capabilities to provide true context.
2. Miniaturization – This technique uses style sheets to adjust existing web pages to a mobile form factors. Again this technique does not provide context and results in a less than optimum user browsing experience.

The advantages of the above approaches include:

1. Quick deployment to mobile web –these can be achieved within a 3-4 week timeframe
2. Lower costs
3. Availability of proven off-the-shelf tools which can enable quick transformation

Taking the above factors into consideration, the recommended approach would be to **think mobile** and design web applications that are intended to be used on mobile as a separate exercise. A major consideration to be kept in mind is - “never design for the lowest common denominator” while using the following design principles:

1. Detect device capability and adapt the content to be delivered
2. Simplicity – An effective mobile layout needs to be simple. The information presented needs to be structured into a layout that minimizes scrolling and the need for zooming. . For example, a single column layout is generally preferred to avoid cluttering and horizontal scrolling.
3. Navigation – Mobile web applications need to provide an easy way of navigating between web pages and make optimum use of the device capabilities. For example, the navigation mechanisms in touch based smartphone are primarily icon and tab driven, while the navigation is menu driven for devices with keyboards. Reducing the number of click to perform an action is necessary to enhance user experience.
4. Content – The content needs to be adapted based on the device.
  - a. Hide unnecessary text and links to optimize use of screen real estate
  - b. Use different graphics for different resolutions



- c. Video formats need to be adapted to the device supported formats
  - d. Interaction mechanisms need to be customized based on device capabilities
5. Online/offline operation – The mobile web experience would be badly affected if it depends on persistent internet connection. Mobile web apps need to leverage local caching mechanism to deliver a smooth user experience while negotiating “dead-zones” and network signal variance.
6. Speed – Mobile web apps need to provide fast response times to the user. This can be achieved by keeping their size small, local caching and usage of background processing to provide a seamless browsing experience.

## Advantage - HTML5?

There is no easy answer with regard to selecting HTML5 or native applications due to multiple factors that need to be considered in the decision making process.

### 1. End user experience

The type of content or service that best fits on web or apps is dictated by the end user needs. Obviously, games works best as native apps as they often use heavy graphics and benefit from not having to access resources via a browser. However, shopping (mCommerce) and services apps are better suited for the mobile web.

### 2. Marketing

Native applications are great for marketing. Once you have an application, you can attract potential interest since the “application phenomenon” is hyped and in demand these days.

### 3. Customer priorities

Customer priorities play a big role in deciding the technology approach. If the customer wants the best user experience irrespective of cost, then the native approach would be right way to go ahead. On the other hand, HTML5 is a better option for customers looking for a good user experience with better code and app manageability.

### 4. Application features

The features that need to be supported in an application will decide whether HTML5 is a right fit. If the application needs access to device features like contacts, calendar, etc, then HTML5 cannot be used.

## 5. Technology landscape

The changing technology landscape dictates the technology of choice. Network coverage, bandwidth availability and device capabilities play a big role in deciding the app development platform. For example, SMS based apps are the most convenient and the biggest hit in developing countries because of various technology constraints.

Although there have been significant improvements in the mobile web technology in recent times, there are still areas where native applications are the best fit. There is no right answer in the debate on choosing between mobile web and native application or going for both. A detailed analysis of the parameters needs to be done to arrive at the right approach best suited for your organization.

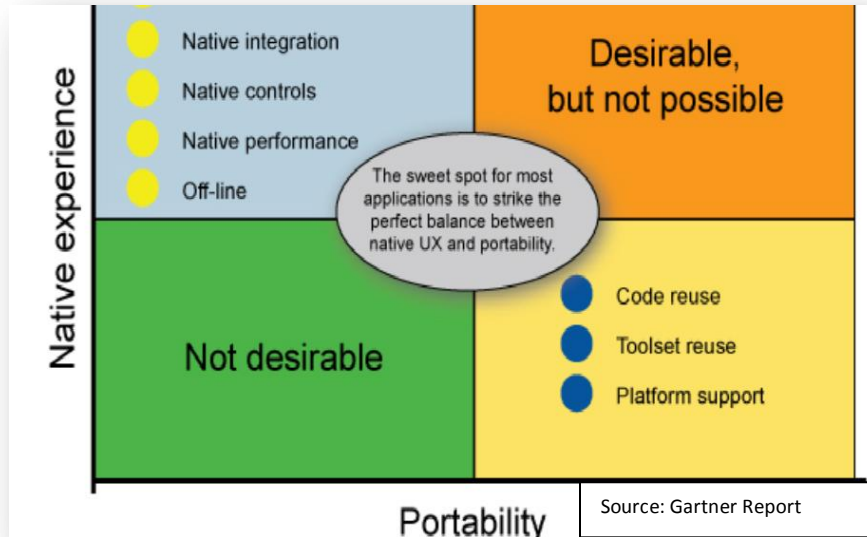
## Doing the Balancing Act & Conclusions

Based on the above criteria, it is clearly seen that organizations need to prioritize their requirements through a mobile strategy in order to come up with a decision-making map on the type of mobile applications that needs to be built.

Some of the questions that need to be addressed while deciding on the strategy are -

1. Are we building applications that are graphics intensive?
2. Are we building applications that need a lot of algorithmic computation?
3. Does the application need integration to device specifics?
4. Is app security a high priority?
5. Do we need to deploy the app on multiple platforms?
6. What changes do we anticipate in the app and can this be addressed on a chosen development model?
7. Is there an immediate need to deliver the app to the market?
8. What is the EOL for the app being developed?
9. Are the chosen tools mature enough to provide sufficient support for development?
10. Does the app need to sync with back end systems and data sources?
11. How frequently do we need to upgrade the app and does the app storefront deployment plan support this requirement?

A key aspect that should be followed during the decision making process is the consideration of **native UX versus portability** since this has broad ramifications on the application manageability, time to market and cost.



As the application portability increases, the richness of its native UX decreases. At the same time when the app has a rich native UX, its code reuse decreases while the cost of maintenance goes up. Developing apps that have native UX will also increase its time to market significantly.

# Appendix

## Embracing HTML5 standards for Mobile environment

HTML5 is the new evolving standard for browser based application development. This is driven by W3C and WHATWG bodies with notable industry participation. The specification of HTML5 is in the working draft and several vendors have already adopted many of these standards.

Some of the major capability areas of HTML5 are as follows:

- a. Several new elements that are introduced
- b. Embedded audio and video interactions
- c. 2D drawing through built-in canvas
- d. Offline storage
- e. Geo location

This section provides a briefing on the HTML5 features and capabilities.

### HTML5 Features

#### Elements

The following are the high level elements classification of HTML5. These new elements enable clearer and simpler markup that makes pages more obvious on mobile devices.

Elements of HTML5	Description
<b>Structural</b>	Headers, footers, articles, navigation and sections
<b>Block semantic</b>	Dialog, figure, aside
<b>Inline semantic</b>	Progress, meter, time
<b>Interactive</b>	Command, menu, datagrid, details

#### Storage on the device

Below are the methods that can be utilized for client-side storage.

Storage of HTML5	Description
<b>Sessions</b>	Storing in browser based sessions
<b>Local</b>	This is a persistent storage. The scope of local storage can also be controlled
<b>Database</b>	Storage ability to use SQL database and can be programmable

## Canvas drawing

Canvas in HTML5 is used to draw graphs on the fly usually through java scripting. The canvas has several methods for drawing paths, boxes, circles, characters and adding images.

## Embedded audio and video

HTML5 defines a new element that specifies a standard way to embed audio files on a web page. They are also designed to give users a faster experience by doing away with browser plug-ins such as Adobe Flash.

## Offline access

This feature is used to improve an application's performance by storing data in the cache or to make data persistent between user sessions and when reloading and restoring pages.

## JavaScript APIs

HTML5 provides JavaScript APIs to exploit several browsing features. This also aids in improving the user interactions on the browser.

- ✚ WebWorker API helps create JavaScript based worker threads that can work in the background without blocking the user interactions.
- ✚ WebSocket API establishes connection between clients and web server.
- ✚ GeoLocation API will extract the user's longitude and latitude coordinates.
- ✚ History API will provide ability to add to the history list.
- ✚ Notification API helps send notifications to the client.

## Support for HTML5 in mobile browsers and devices

The advent of Webkit browser engines is further driving HTML5 penetration into the mobile devices.

Some of the early adopter of HTML5

- ✚ RIM Blackberry Torch
- ✚ iPhone and iPad ( 4.0 and above)
- ✚ Android ( 2.0 and above)
- ✚ IE9 browsers

## About Happiest Minds Technologies:

Happiest Minds, the Mindful IT Company, applies agile methodologies to enable digital transformation for enterprises and technology providers by delivering seamless customer experience, business efficiency and actionable insights. We leverage a spectrum of disruptive technologies such as: Big Data Analytics, AI & Cognitive Computing, Internet of Things, Cloud, Security, SDN-NFV, RPA, Blockchain, etc. Positioned as “Born Digital . Born Agile”, our capabilities spans across product engineering, digital business solutions, infrastructure management and security services. We deliver these services across industry sectors such as retail, consumer packaged goods, edutech, e-commerce, banking, insurance, hi-tech, engineering R&D, manufacturing, automotive and travel/transportation/hospitality.

Headquartered in Bangalore, India; Happiest Minds has operations in USA, UK, The Netherlands, Australia and Middle East.

**Business Contact:** [business@happiestminds.com](mailto:business@happiestminds.com)

**Media Contact:** [media@happiestminds.com](mailto:media@happiestminds.com)