WHITE PAPER

Will Mobile Application Development Follow the Path of Web Development?

Remember, not so many years ago, when supporting a web application across the handful of common browsers was unlikely? For an enterprise environment, there was a strong tendency to standardize on one browser to simplify development of internal web applications. Externally-oriented applications often performed poorly on one or more browsers even if, in theory at least, those browsers were supported. Putting aside for the moment all the old web apps out there, when was the last time you saw a newly developed web app that didn't support Internet Explorer, Firefox, and Chrome – usually even Safari is included. Why? What changed? More importantly, are there lessons we can learn from the evolution of web development that can inform the challenges many companies are facing in the development of mobile applications?

As demand has grown to extend enterprise applications to the myriad of mobile devices in use today, IT departments have been faced with a far more complex environment than they ever had to worry about during web adoption. Initially, companies tried to enforce device standards across the company. Whether it was Blackberry, iOS, or Android, maintaining a monoculture of devices quickly met the rapidly changing world of mobile technology and the, often very senior, employees who wanted to adopt the latest new devices. A corporate IT decision to support only a particular device and/or operating system became untenable in most cases. Even if a monoculture was being maintained, many employees were using different devices and could not realize the value of the company's investment in mobile applications.

In many ways, this is a far more difficult situation for corporate development teams than even the worst days of the browser wars, when multiple – partly or mostly incompatible –standards were being pushed by various tech giants. Eventually, a relative peace was reached in the promotion of incompatible standards. As important as greater adherence to common standards was to simplifying development for multiple browsers, the availability and adoption of web development tools was important to simplifying the situation. Tools could easily support idiosyncratic differences between browsers that had previously depended on the memory of highly knowledgeable developers. Furthermore, tools could automatically generate some of the code needed to support various features on specific platforms.

The current situation with mobile applications is in many ways far more complex than the browser wars ever were. Browsers were, for the most part, based on common standards and had limitations placed on them by the basic user expectation that the websites work properly in the browser they were using. If they didn't, users simply switched to using another browser. Mobile devices often have different hardware, different screen sizes, and different operating systems. Although mobile browsers are slowly growing in importance, most users are still looking for the speed and offline capability that comes only from an installed application. Of course, device manufacturers are happy to encourage development of custom, device-specific applications since these allow them to better differentiate their products. HTML5 is slowly making headway, but for good technical reasons and for reasons of self-interest on the part of device makers, it will likely stay well behind the capabilities of native applications on most devices for the near future.

Some enterprise needs can certainly be met by HTML5, but few companies will find that they can meet all the needs of the business with HTML5 alone. So, development of hybrid and native applications will remain an enterprise requirement. Unfortunately for many enterprises, they have found themselves supporting multiple device platforms and multiple application technologies. They have often arrived at this juncture without the sort of careful planning for future technology requirements that most companies and IT teams would like. Market forces and employee demands have conspired to create a complex environment that is not easily simplified by declaring and enforcing standards – and we've not yet even considered the effects of bring-your-own-device policies.

Fortunately, just as web development tools eventually simplified building web applications for multiple browsers, mobile application development platforms are rapidly reducing complexity for those who have adopted them. Of course, many enterprise development teams reached this situation incrementally. They probably started with requests for modest functionality on an iPhone. Then iPad entered the picture and brought with it requests to make better use of the larger display and more flexible interface. Android couldn't be avoided – there was just too much demand. The development effort grew and many companies could not afford to keep up with the demand, even if the requests did seem like pretty good ideas. Too many enterprise development shops have found themselves devoting ever more resources to the increasingly complex device environment while providing less new functionality. Mobile application development tools should be the answer to that problem. Creating one set

"Too many enterprise development shops have found themselves devoting ever more resources to the increasingly complex device environment while providing less new functionality. Mobile application development tools should be the answer to that problem."

of code that can run on all the major platforms would dramatically increase productivity and lower the cost of supporting enterprise apps on the still-growing menagerie of devices. Of course, a plethora of mobile application development tools have appeared - each claiming to make developers' lives easier.

So what would the characteristics of a good mobile development platform be? Well, first of all, it would need to enable building apps to support the major mobile platforms with very little additional work for each additional platform. Surprisingly, this one requirement eliminates many of the tools on the market. A number of them are made by the device/OS manufacturers. Those generally are only useful for a single OS or brand of device. Another group of mobile development platforms is from large software companies, concerned that somehow the mobile device wave would damage their core businesses. As a result, these products are not really general development platforms, they simply make it relatively easy to build apps that function in the maker's software ecosystem but not at all outside of it.

Evaluating only those products that are truly general purpose mobile application development platforms can still be a confusing challenge as there are at least a couple of dozen that claim the capability. A few further, modest requirements, however, rapidly shrink the list. If you are going to use a development platform, shouldn't it make development across platforms easier? More than a few mobile application development platforms require so much change and customization across different devices that they do little to reduce the amount of work required to support those platforms. Of course, a good mobile application development platform should go further. It should also accelerate the development process itself. Users should benefit from it, even if they are developing an application for only one platform. Development should be faster and have fewer errors than other approaches, and, of course, building the application for multiple platforms should require minimal additional effort.

Faster cross-platform and easier single-platform development are some of the goals Nexaweb had in mind when it launched Nexacro. Nexacro provides a comprehensive ability to write once and deploy to the web as well as popular mobile device platforms. It supports current standards for HTML5 but can also support creation of hybrid and native apps, allowing full use of device capacities inaccessible through HTML5. Developers create the app through the flexible, what-you-see-is-what-you-get (WYSIWYG) development environment of Nexacro Studio and can then deploy to multiple environments.

Nexacro's WYSIWYG development environment is central to maximizing developer productivity

"More than a few mobile application development platforms require so much change and customization across different devices that they do little to reduce the amount of work required to support those platforms."

"Nexacro provides a comprehensive ability to write once and deploy to the web as well as popular mobile device platforms. It supports current standards for HTML5 but can also support creation of hybrid and native apps, allowing full use of device capacities inaccessible through HTML5."

and minimizing bugs. One of the biggest challenges of designing mobile applications is the wide variety of screen sizes available. Providing interfaces that work for PC's, tablets, mini-tablets, fablets, phones, and even watches can be the most complex and challenging part of application development. Nexacro makes creating interfaces that work across the full range of screens sizes straightforward. For more controlled environments, Nexacro allows you to specify the particular devices in use. For less controlled environments, interfaces that perform well over wide ranges of screen sizes can be developed with the app automatically adapting to the device in use.

The current device environment is complicated and, unlike in the days of the browser wars, based on disparate hardware and software which are very unlikely to harmonize in the foreseeable future. With browsers, as the deliberate incompatibilities were eliminated, development tools that could reliably enable support across the popular browsers with modest additional effort emerged. If you accept that a heterogeneous device environment will continue even as demand for enterprise software capabilities on those devices increase, it's obvious that enterprise IT departments will need to identify and adopt tools that will enable development across the device environment. Cost effectively developing applications that work well across the range of popular devices will remain an ongoing challenge, but tools do exist that can dramatically simplify the development effort and reduce the cost. However, that device environment continues to change rapidly and keeping up while meeting the business needs remains an essential problem for most IT development managers. Increasingly, many of us are beginning to look back wistfully for the days when the challenging problem of the day was supporting the common browsers.

To learn more about Nexaweb mobile application development software and services, visit www.nexaweb.com and access our white papers, customer case studies and free software downloads.

WORLDWIDE HEADQUARTERS
Nexaweb, Inc.
1 New England Executive Park Drive,
Suite 205 Burlington, MA 01803

NEXAWEB JAPAN
Nihon Nexaweb KK
10F Akihabara OS Bldg., 5-3-1 Sotokanda, Chiyoda-ku,
Tokyo 101-0021, Japan
www.nexaweb.co.jp | For more information: info@nexaweb.co.jp