control-shift

Jeff Wisniewski (University of Pittsburgh)

Mobile That Works for Your Library

Consider these telling statistics: It's estimated that by 2014 mobile internet usage will surpass desktop web usage. It's also predicted that in 2011, for the first time in history, sales of mobile devices will surpass sales of desktop computers. What does all this mean for libraries? It means that the future is mobile, and the future is now.

But what about other options? What about these "apps" the kids seem so crazy about? Besides mobile websites and mobile apps, are there other options? What are the advantages and disadvantages of each?

> "The future is mobile, and the future is now.

DEVELOPMENT PROCESS

Regardless of the specific method used to go mobile, the development process for all of them is roughly the same.

As always, begin with users. What do they need and want from your library, with respect to mobile? The best way to uncover this is to ask them. Survey your users for insights into their needs and wants. But don't stop there. Gather information about your community's mobile landscape. What devices are they using? This will help inform which mobile path is best suited for your library.

Since a reasonable number of libraries have already gone mobile, it's useful to look at other libraries' mobile offerings. You'll notice a pattern emerging in the content and services that routinely appear:

- Hours and locations
- Actionable contact information including email, instant messaging, text, and, of course, telephone
- Mobile catalog

Some also include links to mobile-optimized sources of content such as EBSCO*host* Mobile, Safari Books Online, or Nature Publishing Group's Scitable.

Once you decide what to include, commit functional specifications to paper. Yes, paper. The next step is designing the interaction and information architecture. Note that it's generally accepted that optimal architectures for mobile format are slim and deep. Instead of floating all of the content to a single mobile homepage, an optimal mobile landing page will have a link to action items. For hours, for example, it goes something like this: hours>branches>main branch> then the hours.

Next in the development process is testing. Paper prototyping works well here. Use index cards or one of the many printable mobile templates available for free on the web. Test with real users. Does your flow make sense? Are calls to action clear? Are labels understandable? Refine as necessary and proceed to visual design. Make it pretty and make it look like an app. Apps, for better or worse, have come to define mobile user interfaces (UI). All major mobile manufacturers have extensive UI documentation available online.

> “Instead of floating all of the content to a single mobile homepage, an optimal mobile landing page will have a link to action items.

OPTIONS FOR GOING MOBILE

Broadly, there are three approaches to going mobile:

1. Native app
2. Mobile web
3. Hybrids

Native apps live on the mobile device itself. They arrive on the device via download. Native apps are platform-specific—they are developed and coded with a specific operating system in mind.

Currently, the major platforms in the U.S. include Android, iOS (iPhone), and BlackBerry OS. To develop an app for any of these platforms requires the use of a software development kit (SDK), which is usually available for download at the companies' respective developer sites. Native apps are coded in programming languages such as Objective-C or Java.

NATIVE APPS: THE GOOD

There are a number of advantages to the native app approach.

From a usability and user experience perspective, native apps usually rate higher than mobile websites. First, native apps provide one-click access, unlike a mobile website that can take several clicks, and text entry, to access. Second, native apps are by default bookmarked and persistent. Once installed, there's a little icon that's always there, just asking to be clicked.

Native apps tend to be highly responsive as well. Since much, or all, of the data that populates the app is part of the app itself, there is less lag and latency than with the mobile web. In addition, all the major mobile platforms provide detailed documentation on functional and user interface convention for their platforms. If developers use the guidelines, they should end up with a usable and conventional (in the positive sense of the word) product. Native apps also have built-in marketing and discoverability support via the online app stores of the major platforms, such as iTunes for the iPhone.

While it can reasonably be assumed that, if someone has gone through the effort of finding and downloading your app, there is some level of commitment there, it turns out the level of commitment is more “going steady” than “till death do us part.” Not surprisingly, app usage is highest in the first 30 days after it's downloaded and from there, it drops off a cliff.

The most compelling argument in favor of native apps, besides the coolness factor, is that they can be used offline. A user does not need to be connected to a network to take advantage of them. They feature deep integration with the hardware and OS, so they can take advantage of the devices' camera and GPS. If your library plans to include QR code reading into your mobile presence, then an app is probably the way to go. For most libraries, however, these native app advantages aren't compelling enough to tip the scale one way or another.

However, don't discount the coolness factor. There are many considerations when choosing among methods, some legitimate and some not. But the fact remains that apps are undeniably cool and may encourage your users to continue to access your library's information via their mobile devices.

NATIVE APPS: THE BAD

The single most significant drawback to the native app model is the fact that they are platform-specific. Unless your community of users is remarkably homogeneous, that means you'll be developing for at least two or more platforms.

In addition to the diversity of platforms, there is now a diversity of form factors within each platform. Apple has the iPhone and the iPad. Android has Android phones and a slew of new tablets coming to market. Research in Motion, Ltd. (RIM), the maker of BlackBerry, has announced plans to launch a tablet as well. While apps technically written for an OS's smartphone will work on the same OS's tablets, the massive difference in screen real estate means that, in many cases, a smartphone app needs to be optimized for the tablet. Take each platform, multiply by two, square that, and … (insert headache here).

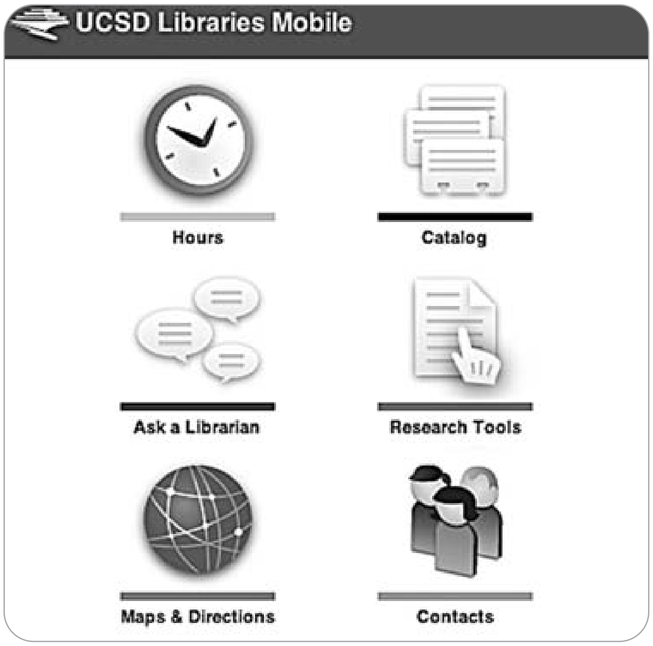
In addition to this platform proliferation, the programming skills needed to code native apps aren't familiar to many library staff members. Even when these skills are available in-house, they tend to be skills that are in high demand. Thus, mobile development can fall somewhere lower in the development queue than is preferable.

Finally, native apps aren't simply launched and forgotten. They need to be updated. Given the fact that the app lives on the mobile device, updates are pushed to them cyclically.

This is in contrast to a mobile website that, like its desktop cousin, can be updated instantly. If one chooses, it can also be updated fairly continuously.

MOBILE WEB

Mobile websites are, well, websites. They live on a web server, not on the device itself. While they need to be designed and optimized for mobile use, they are developed using an infrastructure and languages that are already familiar in most libraries. If you have a desktop website, which most likely is written using HTML, CSS, and JavaScript, you can have a mobile website, such as the example shown here from the University of California–San Diego.



The University of California–San Diego's mobile website

MOBILE WEBSITES: THE GOOD

Libraries have been in the website game for many years and usually possess staff members who are proficient in one or more of these web languages. The issue of developer mindshare is less of an issue with mobile web than it is with native apps.

Because mobile websites are websites, they can be tested for functionality and usability more easily than native apps. While not an exact replica of real-world mobile context, mobile websites can be tested on a desktop computer using tools and techniques that are already familiar, to uncover at least the most pressing issues.

There are two compelling arguments for mobile websites. First, libraries can develop them once, using familiar languages, and they'll be accessible across the widest possible array of devices. Second, no child is left behind here. Users of devices on any platform, be it iOS, Android, BlackBerry, Symbian, or Windows Phone can access your content, as long as they have a mobile browser and a data connection.

MOBILE WEBSITES: THE BAD

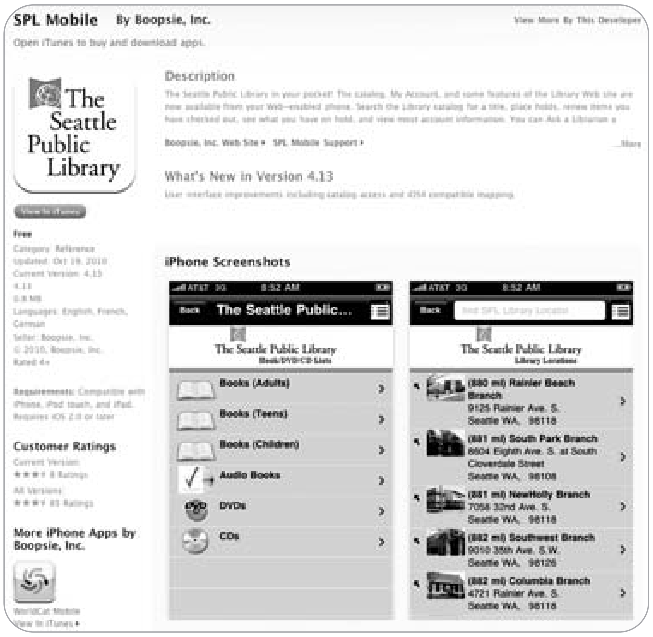
Mobile websites tend to score lower in usability and user experience. Indeed, usability expert Jakob Nielsen recently wrote (www.useit.com/alertbox/mobile-apps-initial-use.html), "App users … suffered much less misery than users in our mobile website tests." There are a couple of reasons for this, some of which are out of the control of the developer. First, mobile websites require several actions to access. Open the mobile browser, type in the address (if you remember it), and wait for the site to load. Unlike a native app, a user needs to make this click investment just to access your mobile content.

A recent survey of mobile users revealed that a majority indicated they had had problems with mobile websites. The problem reported the most was the site's loading speed. Here's the rub: As developers we can code our mobile sites to be as lean and mean as possible, but beyond that there's not much more we can do. The device, browser, and network of a user all impact the user experience of a mobile website, so a significant degree of the user experience is out of our hands.

HYBRIDS APPS

Hybrid apps are a cross between mobile web and native apps. Hybrid apps are installed on the mobile device, like native apps. But unlike native apps, they are developed using web languages that are more familiar to libraries than those used to code native apps. The app, developed using HTML, CSS, and JavaScript, is then "wrapped" in a native app wrapper so that it can be installed to a mobile device like a true native app.

Because these apps are built using HTML, CSS, and JavaScript, many libraries already have the skill set needed



SPL Mobile By Boopsie, Inc.

View More By This Developer

Open iTunes to buy and download apps.

Description

The Seattle Public Library in your pocket! The catalog, My Account, and some features of the Library Web site are now available from your Web-enabled phone. Search the Library catalog for a title, place holds, renew items you have checked out, see what you have on hold, and view most account information. You can Ask a Librarian a

Boopsie, Inc. Web Site › SPL Mobile Support ›

...More

What's New in Version 4.13

User interface improvements including catalog access and iOS4 compatible mapping.

View In iTunes

Free
Category: Reference
Updated: Oct 19, 2010
Current Version: 4.13
4.13
0.8 MB
Languages: English, French, German
Seller: Boopsie, Inc.
© 2010, Boopsie, Inc.
Rated 4+

Requirements: Compatible with iPhone, iPod touch, and iPad. Requires iOS 2.0 or later

Customer Ratings
Current Version:
★★★ 8 Ratings
All Versions:
★★★ 85 Ratings

More iPhone Apps by Boopsie, Inc.

WorldCat Mobile
View In iTunes ›

iPhone Screenshots

AT&T 3G 8:52 AM
Back The Seattle Public...
The Seattle Public Library
Book/DVD/CD Lists
Books (Adults) ›
Books (Teens) ›
Books (Children) ›
Audio Books ›
DVDs ›
CDs ›

AT&T 3G 8:52 AM
Back Find SPL Library Locatio
The Seattle Public Library
Library Locations
(880 mi) Rainier Beach Branch
9125 Rainier Ave. S.
Seattle WA, 98118 ›
(881 mi) South Park Branch
8604 Eighth Ave. S. at South Cloverdale Street
Seattle WA, 98108 ›
(881 mi) NewHolly Branch
7058 32nd Ave. S.
Seattle WA, 98118 ›
(882 mi) Southwest Branch
9010 35th Ave. S.W.
Seattle WA, 98126 ›
(882 mi) Columbia Branch
4721 Rainier Ave. S.
Seattle WA, 98118 ›

The Seattle Public Library uses a hybrid app, available for iOS, Android, Windows Mobile, Palm, and BlackBerry.

to create them. Often the services that assist in the creation of the hybrid app have graphical drag-and-drop interfaces that minimize even this coding.

Notable hybrid app developers include the following:

- Nitobi Software: www.phonegap.com
- Appcelerator, Inc. (Titanium): www.appcelerator.com
- Rhomobile, Inc.: http://rhomobile.com
- MotherApp: www.motherapp.com
- Boopsie, Inc.: www.boopsie2.com
- Netbiscuits, GmbH: www.netbiscuits.com

HYBRID APPS: THE GOOD

Hybrids apps offer the persistence (think, icon staring you in the face) of a native app with the easier development activity of using user-friendly tools and familiar programming languages.

Some hybrid apps can make calls to the mobile web, meaning that the user at some point is taken out of the app itself and taken to a mobile webpage. The advantage here is that, like a full mobile site, mobile webpages are easier and faster to update than native apps.

HYBRID APPS: THE BAD

Despite being built with web languages, in the end hybrid apps are still apps in the sense that their app "wrapper" is platform-specific. So while you can develop once, you can't, in the truest sense, deploy anywhere—you deploy to specific platforms. Not all hybrid app generators support all platforms, so based on the data you collected from the mobile survey you surely did, pick an app generator that wraps in the platform(s) you're targeting.

A brief note on metrics: Tracking usage is as important in our mobile presence as it is in our desktop sites. Most analytics packages, including Google Analytics, can separately track mobile usage stats. In addition, Google recently released a free tool to track activity in your mobile apps, for Android and iOS only (http://code.google.com/mobile/analytics).

WHAT TO DO?

Which mobile path is right for you depends on many factors: your constituency, your budget, your goals, and your skill set, to name a few. There's a mobile path out there that's a good fit for your organization. Regardless of the path you choose, know that now's the time to get mobile. See you on-the-go.

__Jeff Wisniewski__ (jeffw@pitt.edu) is web services librarian, University Library System, University of Pittsburgh.

Comments? Email the editor (marydee@xmission.com).