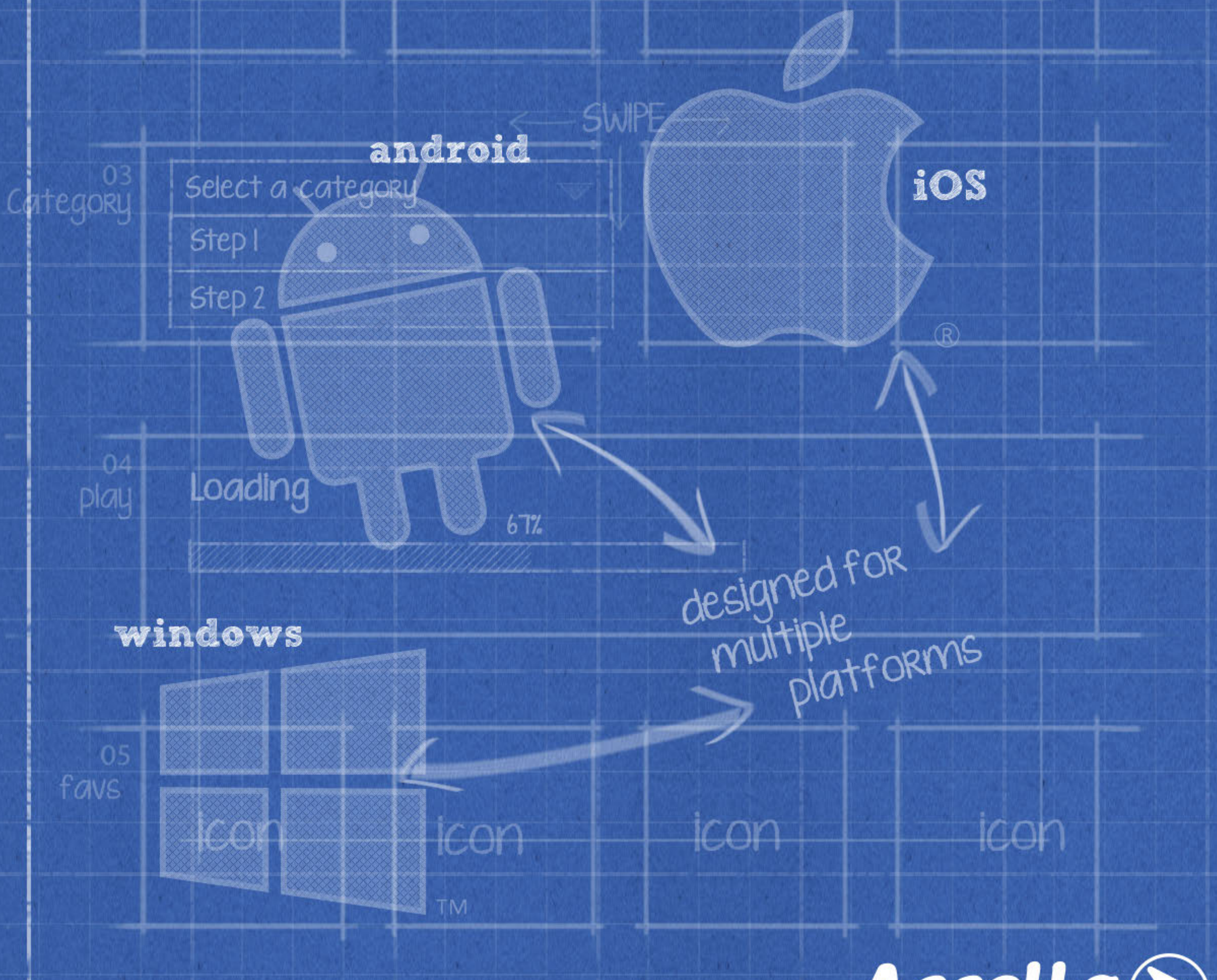
Cross-Platform Development

Understanding the difference between Native and Hybrid Development



Accella

Cross-Platform Development

The world of mobile applications has exploded over the past five years. Since 2007 the growth has been staggering with over 1 million apps available on both iOS and Android platforms. This extreme growth has many companies and organizations looking to create mobile applications to satisfy not only their client's needs, but also their own internal organizational demands.

When companies look at building mobile applications, they can either use internal resources or outsource to a professional software development company. Both options incur costs, whether those are opportunity costs or direct costs to development firms, much of that is determined by what type of app they build, a native or hybrid application.

When developing an application which can be downloaded and installed on a user's device, there are two main avenues for writing code:

NATIVE DEVELOPMENT - Apps are separately coded for each respective platform in their native language (I.E. Java for Android, Objective-C for iOS)

HYBRID DEVELOPMENT - Apps are coded once, typically in a web based language and the code is adjusted to work on various platforms

This eBook is going to look at both approaches in an effort to give your organization insight as to which method will work best in creating an application.

Cross-platform development tools, such as PhoneGap or Titanium, can offer some up front time savings in the actual development of an application. However, the overall development process from start to finish may not see those same time savings. In fact, in our experience, when you look at the overall time it takes to build an app, taking in to account testing, bug fixing, and upkeep, using a hybrid approach will often cost a company more time and money in the long run.

WORLDWIDE STATISTICS

2013 Total App Downloads

Android	29B
Apple	27B
BlackBerry	2.4B
Windows	4.1B

Source: VisionMobile

What Developers Have Built with HTML5

87% - Have Built Desktop Websites & Web apps

53% - Have Built Mobile Websites

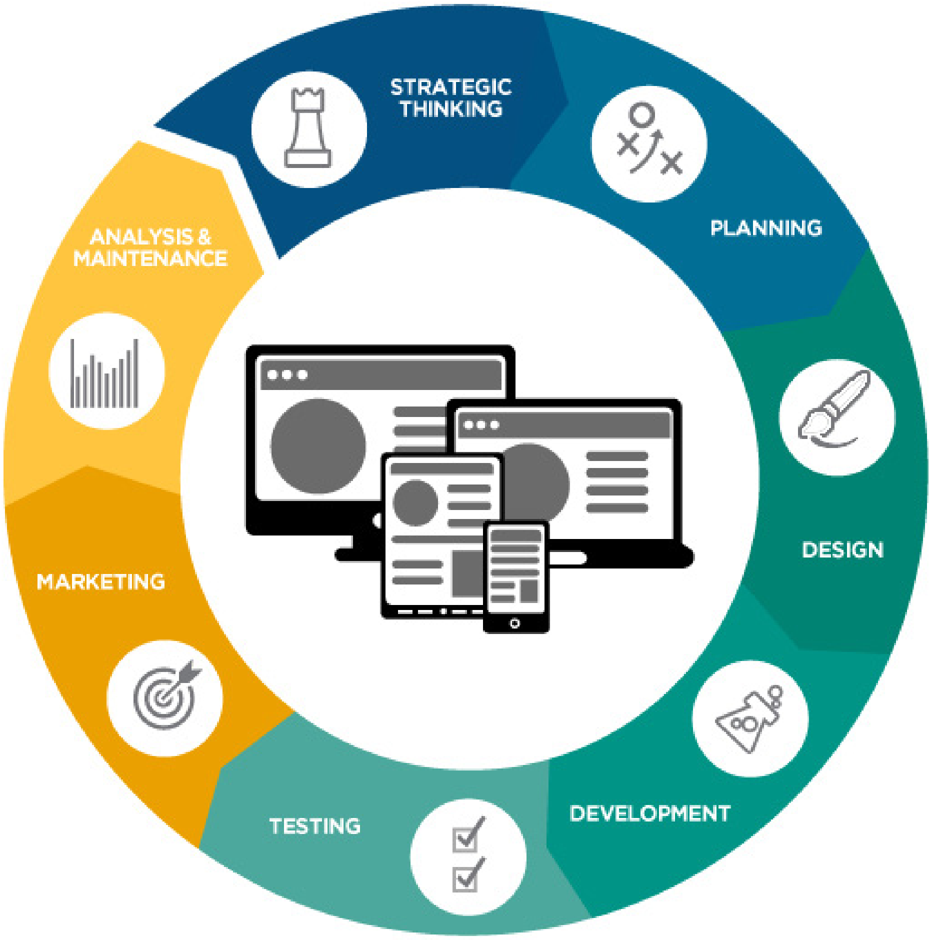
29% - Have Built Installable Mobile Apps (Hybrid Apps)

Source: VisionMobile

THE PROCESS OF BUILDING AN APPLICATION

Let's begin the discussion by looking at the process of building a mobile application. Creating an app extends well beyond simply writing lines of code in Java (for Android), Objective C (for iOS), or a combination of HTML5 & CSS (for web and hybrid applications).

Our team has identified 7 key steps in building a mobile application. As you can see below, development is just one of the seven main steps in creating an app. Though it is one of the most important and time-consuming steps, there are six other phases which need to be executed well in order to successfully launch an application.



WHAT IS NATIVE APPLICATION DEVELOPMENT?

Native phone apps are written in object-oriented languages like Java (Android) and Objective C (iOS) – the same languages that power a device's mobile operating system. It's similar to writing a computer program for Windows vs. Mac. The application is designed from the ground up to take advantage of the features and benefits each platform's hardware provides. These apps are designed to interact with the controls and software connections for each operating system.

Conversely, cross-platform apps rely on basic web programming languages such as HTML, CSS, and JavaScript. Mobile apps written in these web-programming languages are then compiled into the native language of the mobile device's platform (iOS, Android, Windows, etc.)

We find organizations go with the native approach because they are looking for a robust application that performs as expected, is easy to update & maintain, and takes complete advantage of each device's feature-set.

That's not to say that native development doesn't come with its own set of challenges.

Benefits of Native App Development

Speed of Applications

Native applications are faster than hybrid apps as all of the code is natively written. For cross-platform development (CPD) only a fraction, such as Titanium, are compiled into native code. The majority are reliant on web technologies causing a lag in responsiveness.

Feature Set

Some of the most critical Smartphone functions rely on the device's native capabilities – features like the camera, geo-location, gestures, and push notifications. Native apps benefit from being able to easily access and integrate these features.

User Experience

Native apps are designed to work on a relatively narrow range of devices, which means that it is that much easier to ensure that the look and feel of the app is consistent with the device's operating system. Users aren't confronted with any buttons or menu styles that they aren't familiar with, enhancing the overall experience.

Security

There are a variety of strategies for handling security in your application. What route you take not only depends on your device, but also your security parameters and functional requirements.

Web Applications

Another option is creating a Web-based application.

These apps are run via a web-browser, so they cannot be installed directly on to the device, and they also don't allow you to access a device's features such as the camera, GPS, or push notifications.

While these apps can work on any device with a web-browser, they have many drawbacks vs. an app which is installed on a smartphone or tablet

Basically, native app development will always be able to tackle any of those security strategies and meet any requirements head on. Cross-platform tools may or may not have access to the same strategy out of the box.

> “Currently there are **2** MILLION apps available via the major app stores.
>
> Less than **5%** of those apps were developed with the help of CP Tools.”
>
> Source: Research2 Guidance

So What Drawbacks Come with Native Development?

Development Requires Specialized Skills

iOS is written Objective C, while Android uses Java. Native development isn’t a walk in the park, and a company’s in-house development team may not have the prerequisite experience needed to properly build the app.

Resources are Limited

Because building a native app requires specialized skills, the amount of resources available are limited. That’s not to say there aren’t plenty of developers out there, but finding one that understands HTML is much easier than finding one that can program in Objective C and Java.

Time to Deployment

Because native apps need to be coded for each platform separately, many organizations develop for one platform first and then start developing for the next platform once the app is finished (a process we agree with). This can mean that you have a functioning app on one platform for three to six months before the equivalent hits the other’s store.

Budgetary Concerns

With a hybrid approach, company’s can often use existing employees to build their applications and their salary comes out of the operating budget. When companies use a native approach, they often need to either hire new employees or work with a qualified development team. This can add to your operating budget by bringing on a new employee, or funds will need to come out of an IT or marketing budget for contracting out development.

HYBRID APPLICATION DEVELOPMENT

Companies are lured to the idea of using cross-platform development because it can save you time and money in development costs, that’s the theory at least. In reality, those savings may be minimal at best.

How is Cross-Platform Development Beneficial?

Companies Can Use Existing Resources

With native application development, developers need to use special development languages in order to program applications. This is a skill that existing development teams quite often don’t have, so companies have four general options to build a mobile app. They either hire a new, full-time developer that has these skills, have one of their existing developers learn the new language, contract with an outside development firm to do the programming with them, or use a cross-platform development tool that allows their developers to use existing programming knowledge to build an app.

Cross-platform development allows a company to build apps using a base set of code that is then deployed across multiple platforms. In cases of mobile applications, cross-platform development streamlines the development process for apps that will be simultaneously released for iOS and Android, though they can also be built for Windows Phone, and the mobile web.

There is a Reduced Development Time Frame

By using your company's existing resources and concepts that your organization is already familiar with, you don't need to spend time trying to recruit a qualified and capable development team for each of your app's platforms.

Useful for Proof of Concepts

Companies can use internal resources to produce their app to show to stakeholders before proceeding with native development.

In 2012 Mark Zuckerberg, Facebook CEO, was speaking on stage at a TechCrunch event. He was speaking about Facebook building an HTML5, web-based mobile app. The app, which he himself admitted was "slow" and "clunky" was also "the biggest strategic mistake [Facebook] ever made."

Facebook spent 6 to 8 months working on that hybrid app which resulted in a slow and unstable application.

Facebook scrapped that entire project and instead started fresh, building a completely native iPhone application that resulted in a much faster, more reliable app.

What Challenges Does One Face When Using Cross-Platform Development?

Opportunity Costs Need to be Factored In

When organizations attempt to use cross-platform development, they typically take one of their existing developers or development teams and have them learn the new tool. This pulls them off of existing projects as they are needed to learn new skills and tackle new projects.

Developers Have a Hard Time Debugging Apps

Anytime you code something, whether it be a mobile app or a website, there are bound to be bugs, or problems that arise. This could stem from forgetting to put a close bracket ">" at the end of a command or missing a letter here or there. With cross-platform apps, there are now multiple places to look for these bugs. You're not

Continued: Page 8

Cross-Platform Development Tools

There are many different tools to choose from, the ones listed below are some of the more popular, widely used options.

ADOBE PHONEGAP BUILD

www.phonegap.com

PhoneGap allows for development using HTML, CSS, and JavaScript. App builds are compiled in the cloud and subsequently deployed to the corresponding app store. Key features of this service include built-in debugging and options for collaborative development between users.

CORDOVA

cordova.apache.org

Made by Apache, Cordova makes use of HTML, CSS, and JavaScript allowing users to make use of a device's native functions. This one is free and open-source, and will remain so forever, though with that, you may not find the support resources you need should your team encounter serious problems.

HTML WITH WRAPPER

www.appwrapper.org

AppWrapper simplifies developing for Android and Windows Mobile devices; at the present moment, the tool does not support iOS development. The SDK takes your HTML or JavaScript application and packages it for a mobile-friendly app experience. Features include ad network integration, Flurry and Google Analytics, bug tracking, and MobileAppTracking metrics.

MOTOROLA'S RHO MOBILE

www.rhomobile.com

The Rho Mobile suite builds iPhone, Android, Windows Mobile, and web-based applications using Ruby. This software suite is particularly notable in that it includes almost everything you need to build, test, and deploy the app. In addition to the Rhodes development tool, RhoMobile contains device emulators, design frameworks, and app management solutions.

TITANIUM

www.appcelerator.com

Appcelerator's Titanium allows for multi-platform app development using JavaScript. It features over 5,000 APIs, built-in analytics, and a device emulator that allows for rigorous app testing. Although the SDK itself is free, a paid version allows for automated testing and code optimization.

XAMARIN

www.xamarin.com

Xamarin is a unique tool in that it allows cross-platform apps to operate with all the benefits and features of native apps. But only a small portion of the base app code can be used between each platform. You will need to have someone on your development team who can write for each of the target platforms. Additionally, Xamarin apps are written in C#, which is not as common as JavaScript or HTML5.

just looking at one set of code, you need to check to see if the bugs occurred when you wrote the app in the development tool, or did it happen when you converted the app to the native code. Then in order to test it, you'll need to fix the bug, recompile the code, then test it. This process can become tedious and cumbersome, especially for developers that don't fully understand the operating system's native language.

The User Experience Suffers

When you create one app and deploy it on multiple devices, often times designers are forced to make one design, which will be deployed on both platforms. This results in a less than ideal user experience.

Maintenance Can Become Difficult

Cross-platform development tools compile apps to work with a certain version of the target operating system. Consequently, this means that all hybrid apps work at the mercy of the platform owner (e.g. Apple and Google). All it takes is one small update to a smartphone's operating system in order to render the app buggy and unstable.

Your App May Lag on Some of the Device's Features

When considering a hybrid solution, it's essential to remember that Titanium, Cordova, and others learn about new features and additions to the operating systems from Apple and Google at the same time all developers do. When you use a CPD tool, you then have to wait for that tool to create a script allowing you to use these new features.

Time Savings May Be Smaller Than you Think

Overall, there are a lot of steps in creating a mobile application, not only developing it, but planning, designing, marketing, and maintaining the app. When you use a cross-platform tool, the only step where you're saving significant amounts of time is in development. With many of the cross-platform tools available today, there is still a significant portion that needs to be coded independently for each mobile platform. Actual time-savings may only be 10-20% of the overall development time.

Cross-Platform Apps do Not Run As Fast

This is caused by a result of the apps being built in HTML and not in an app's native code. They are web-based applications, so when a page is loading, they are loading HTML pages instead of using native controls.

You May Need Native Developers After All

Some cross-platform development tools can write the entire app in HTML or JavaScript and then compile that code, others will do a certain percentage in HTML and then require the additional portions to be written in Java or Objective C.

This can also come in to play if you compile code and have bugs in it, you may need to hire developers that know Java or Objective C to help find where bugs are.

How often do Apple and Google update their OS?

Android

1.0 (No Name)
9/23/2008

1.5 Cupcake
4/27/2009

1.6 Donut
9/15/2009

2.0 Froyo
5/20/2010

2.3 Gingerbread
12/6/2010

3.0 Honeycomb
2/22/2011

4.0 Ice Cream
Sandwich
10/19/2011

4.1 Jelly Bean
7/13/2012

4.4 KitKat
9/3/2013

iOS

1.0 - 3/6/2008

2.0 - 7/11/2008

3.0 - 6/17/2009

4.0 - 6/21/2010

5.0 - 10/2/2011

6.0 - 9/19/2012

7.0 - 9/18/2013

Kiran Prasad, LinkedIn's senior director for mobile engineering on why they went from a hybrid app to native:

"If you look at Android and iOS, there are two very large corporations that are focused on building tools to give a lot of detailed information when things go wrong in production. On the mobile web side, getting those desktop tools to work for mobile devices is really difficult."

SO WHICH IS BETTER, A NATIVE OR HYBRID APPROACH?

We've taken a look at the differences between native and hybrid development, and each offers their own advantages and challenges. We have been developing mobile apps since the advent of the iTunes store and we have seen our share of successes and failures with each approach, as well as projects that came to us long after they were supposed to be completed. It's our history and experience that leads us to choose native development nearly every time.

The speed, resources, and maintenance benefits of mobile apps are just too much to overcome for a hybrid's approach of saving time in development. What time that may be saved in development is often quickly eaten up in debugging apps and fixing problems in the code.

Hybrid development may be a good choice for simple, static applications, or proof of concepts that we've mentioned. But until we see vast advancements in HTML5 techniques and cross-platform development tools, we can't truly endorse a hybrid approach over native for the vast majority of our clients.

About Accella

At Accella, we are designers, developers, marketers, and problem solvers that help our clients create award-winning websites and mobile applications. Our team works in tangent with our clients' marketing, IT, and digital strategists to produce websites and mobile applications that are delivered on time, in budget, and within scope.

Based around a process of user-centric design, Accella digs deep into the needs of our clients to produce websites and mobile applications with a purpose. Communication, problem solving and creativity are the hallmark of Accella projects and that is why our clients keep coming back to us year after year with their most important projects.

Accella is a team that you can trust. We Create Happy Clients.

To see examples of our work, visit our portfolio:
www.accella.net/mobile-portfolio.