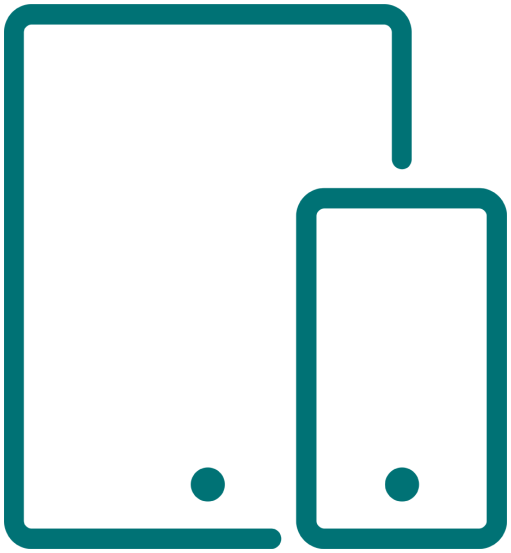
Choosing the Right Mobile Development Strategy for YOU



Vivek Agarwal
CTO and VP, Digital Experience Solutions
XTIVIA Inc.

A Publication of
XTIVIA

TABLE OF CONTENTS

About the Author

Vivek Agarwal is our CTO and VP of the Digital Experience Solutions practice at XTIVIA. Vivek and his team have implemented hundreds of successful web and mobile app solutions at enterprise and mid-sized clients since the year 2000, and he is considered a thought-leader in the digital experience space. His team provides consulting, architecture, design, implementation, and support services around digital experience platforms, mobility, integration, cloud, analytics, and other enterprise software. Prior to his current responsibilities, Vivek served as a Systems Architect, delivering multiple global projects using Enterprise Java, JavaScript, Portal, and CMS technologies. He was also a member of the prestigious IBM Gold Consultant program.

Introduction

This e-book is intended for the Product Owner, IT VP/Director/Manager, Enterprise Architect, or Mobile Developer who is tasked with trying to implement a mobile app, and is confused by the multitude of mobile app development approaches out there and the associated pros/cons. In that situation, you need objective, yet expert, information. Even more, you need it from an unbiased perspective. I have kept this in mind while putting this together, although where I have a personal preference or opinion, I share it with that disclaimer.

That said, the right choice for you is going to be dependent on your situation so any potential bias presented here will be largely irrelevant.

The way for you to get the most value for this information is to read through it, note the highlights that are most relevant to your organization, and then answer the questions throughout this piece to help you discern the best decision for your organization.

So grab a cup of coffee (or your favorite beverage) and read on...

Mobility and mobile strategy has been the focus of many CIOs, business owners, and enterprises for many years as a fundamental and significant architectural decision; this technology space has been evolving at a rapid pace with new mobile development tools and frameworks emerging each year. Replatforming your mobile app is quite the endeavor and not for the faint-hearted! Beyond the complexity of choosing one of the numerous mobile app development options available, what makes this decision even more challenging is the fact that what was in vogue yesterday may be frowned upon today by the elitists in the mobile community.

This article will walk you through the main mobile app development styles available to you today, then introduce criteria that are important while evaluating the development options, provide a comparison of the development options based on these criteria and, finally, summarize the key takeaways.

If you are short on time, or just eager to see the bottom-line, you can skip forward to see the full version of this comparison table. However, the full article will give you relevant background information and color commentary that will enable you to gain the most value.

Criteria	Mobile Apps		
	Fully Native Apps	Cross-Platform Native Apps	Hybrid Apps
Strategic Importance of the App	5	4	3
Your User Base	5	4	3
Your Competition	It Depends	It Depends	It Depends
Targeted Devices	5	3	2
Implementation Costs	1	4	5
Time to Market	1	4	5
Look and Feel	5	4	3
Feature Parity	1	4	5
Required Developer Skills	1	4	5
Developer Pool (For Hiring)	2	5	5
Application Security	5	4	2
Code Size and Reusability	1	4	5
Native Feature Support	5	4	3
Performance	5	4	3
Developer Community Interest	5	5	3
Hiring Company Competition	5	5	4
Technology stability	5	3	3
Development Agility	1	4	5
Interaction with Other Apps	5	3	2
Acceptance Into the App Store	5	5	4

The Options

Following is a description of the three primary mobile app development approaches, though there is some overlap between these approaches and some of the boundaries between them are not all that clear. However, first, let's cover a couple of mobile development options that don't involve developing and maintaining mobile apps.

Non-App Based Mobile Development Options

If you are looking to serve application functionality and/or content to mobile users, but don't necessarily care about having a mobile app published to the App Stores, then you can simply leverage Responsive Web Design (RWD) or Responsive Websites as your default option.

Alternately, you can leverage Progressive Web Apps (PWA) for rich functional websites that look and feel much like mobile apps without having apps published to the App Stores. Note: mobile-optimized websites that are developed specifically to serve mobile devices and are separate from the regular desktop websites no longer make sense with the advent of mobile responsive websites and progressive web apps.

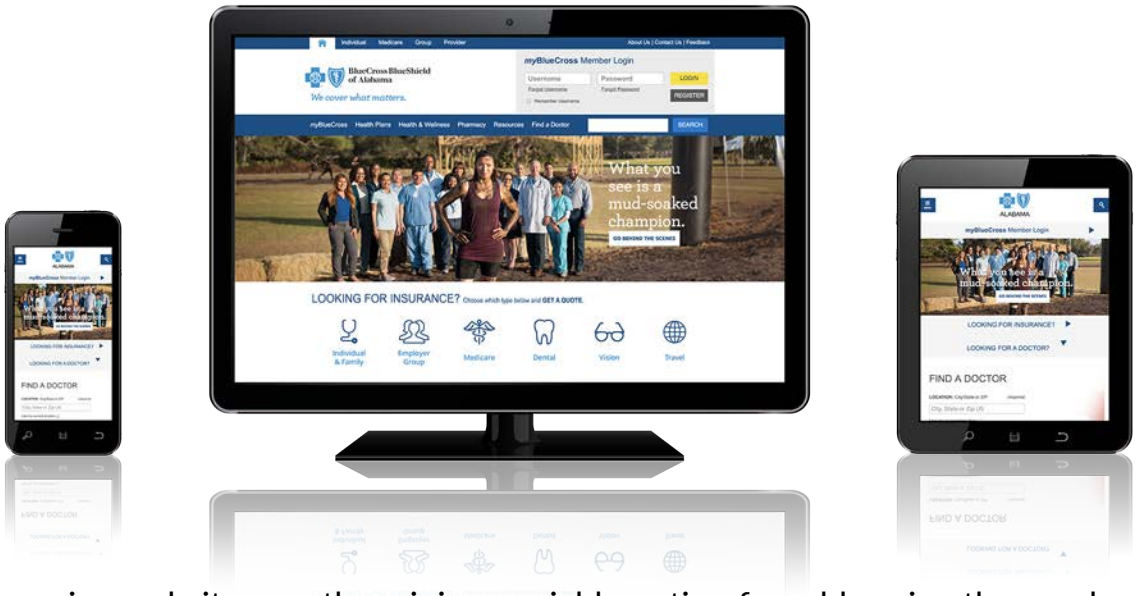
1.1 Responsive Websites

Per Wikipedia, Responsive Web Design (RWD) is an approach to web design which makes web pages render well on a variety of devices and window or screen sizes. In other words, a responsive website is ***responsive*** or ***adaptive*** to the user's access method (different device types, form factors) and adapts the layout to the viewing environment by using fluid, proportion-based grids, flexible images, and CSS3 media queries - an extension of the @media rule, in the following ways:

- The fluid grid concept calls for page element sizing to be in relative units like percentages, rather than absolute units such as pixels or points.
- Flexible images are also sized in relative units to prevent them from displaying outside their containing element.
- Media queries allow the page to use different CSS style rules based on the characteristics of the device the site is being displayed on, most commonly the width of the browser.

1.1 Responsive Websites (cont.)

While building responsive websites, it is strongly advisable to optimize image delivery by leveraging advanced image strategies such as vector graphics and responsive images - the Mozilla site is a good resource for understanding some of the issues with RWD and how “responsive images” help you address them.



Responsive websites are the minimum viable option for addressing the needs of your mobile users. Compared to mobile apps, this approach has the following pros and cons:

- Pros
  - Least expensive from an initial implementation and ongoing maintenance perspective (assumes that you need to provide your mobile functionality via a website and not just mobile apps)
  - Runs from a mobile web browser on any device (iOS, Android, Windows, and others) resulting in the broadest device support
  - Built using standard web technologies - HTML5/JavaScript/CSS, making it easier to find developers and share them across web/mobile initiatives
  - No submission, approval, or distribution needed by an app store
  - Run across multiple screen sizes
- Cons
  - Lowest performance
  - Limited access to native device functionality
  - Slowest ability to adopt new features

1.2 Progressive Web Apps

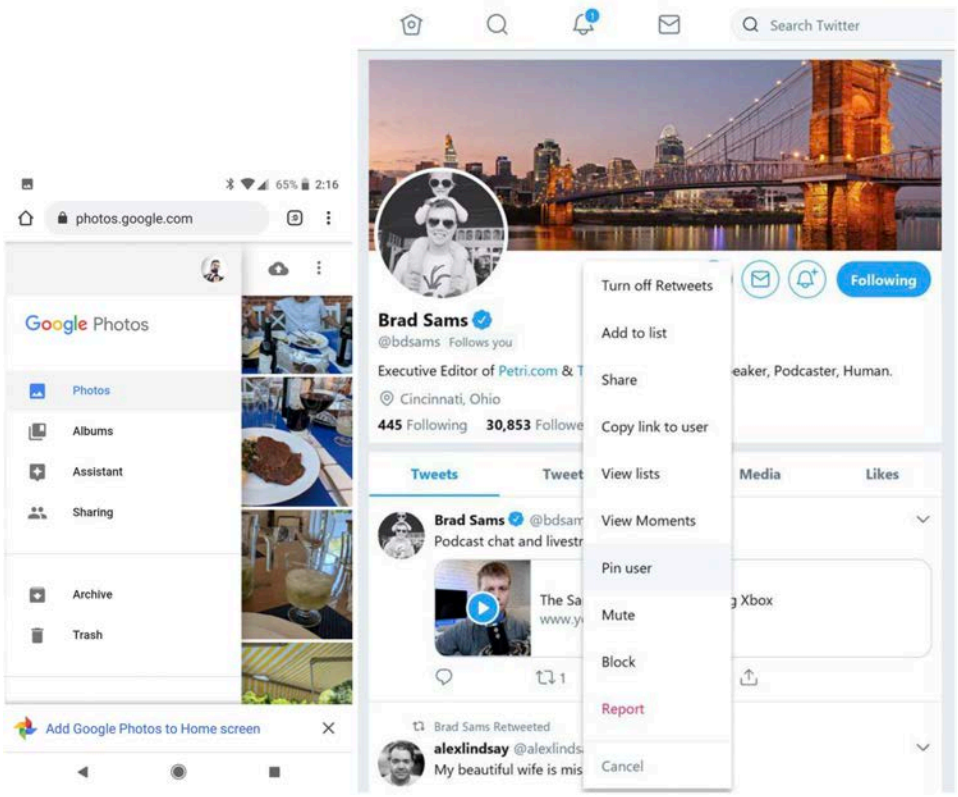
Per Google, Progressive Web Apps (PWAs) use modern web capabilities to deliver fast, engaging, and reliable mobile web experiences that mimic capabilities typically expected of mobile apps - for example, they can support offline experiences, background synchronization, and push notifications on supported browsers. A caveat is that, on unsupported browsers, the user experience degrades such that it resembles a responsive website.

Some characteristics of PWAs (source: Wikipedia) are:

- Progressive - Work for every user, regardless of browser choice because they're built with progressive enhancement as a core tenet.
- Responsive - Fit any form factor: desktop, mobile, tablet, or other forms yet to emerge.
- Connectivity independent - Service workers allow work offline or on low quality networks.
- App-like - Feel like an app to the user with app-style interactions and navigation.
- Fresh - Always up-to-date thanks to the service worker update process.
- Safe - Served via HTTPS to prevent snooping and ensure content hasn't been tampered with.
- Discoverable - Are identifiable as "applications" thanks to W3C manifests[6] and service worker registration scope allowing search engines to find them.
- Re-engageable - Make re-engagement easy through features like push notifications.
- Installable - Allow users to "keep" apps they find most useful on their home screen without the hassle of an app store.
- Linkable - Easily shared via a URL and do not require complex installation.

1.2 Progressive Web Apps (cont.)

Microsoft made a big splash in February with its support for Progressive Web Apps on Windows and iOS has also recently added support for PWAs to Safari in the 11.3 update. That means you can now make a Progressive Web App and ship it to Android, iOS, Chrome OS, and Windows.



Sample PWAs: Google Photos and Twitter Lite

PWAs have similar pros and cons as responsive websites do compared to mobile apps. However, they are clearly a huge step up from responsive websites in terms of user experience though they are obviously more expensive to develop and maintain.

App Based Mobile Development Options

If the need of the hour is a mobile app, and nothing less will do, then you are essentially looking at three options - fully native apps, cross-platform apps, and hybrid apps. This section describes these options at a high level but we will compare them at a deeper level in a later section.

2.1 Native Mobile Apps

A fully native application is an app developed for a certain mobile platform using the programming languages, software development kits (SDKs), APIs and tools specific to that platform, and requires your app developers to learn the toolchains for iOS and Android platforms (with the consolidation that has happened in the mobile industry you can concentrate on these two platforms and have excellent coverage of your users).

Typically, you have two separate sets of developers - one focused on Android using Android Studio and Java (or Kotlin) and another that is focused on iOS using XCode and Swift.

The fully native mobile apps approach is the Ferrari of all the options available to you - gets you the highest performance and functionality but also costs you the most.

2.2 Cross-Platform Native Mobile Apps (no WebView)

A Cross-Platform Native Mobile Application is an app developed using a technique where you write the application code once, and run it ***natively*** on multiple platforms (Android & iOS at the bare minimum and possibly others). These apps do **not** run inside a WebView (embedded web browser) which eliminates an abstraction layer that slows hybrid mobile apps (more on them in the next section) and this is what defines these apps as **native**.

There are a number of Cross-Platform Native App development tools but the most popular ones (as of the date of this article) are React Native (JavaScript) from Facebook and Xamarin (C#) from Microsoft. Flutter is a relatively new Cross-Platform Native App development tool that looks promising and, further, has the backing of Google.

Cross-platform iOS and Android Instagram Apps
(using React Native)
Screenshot source: Instagram Engineering Blog

The cross-platform native mobile app development approach is a great blend of performance and relatively low development cost (compared to fully native apps); the caveat with them is that they may lag behind the platform-specific fully native tools in terms of new feature compatibility and may not include support for all native features.

!

Cross-Platform Native Mobile Apps using tools such as React Native and Xamarin present a very enticing value proposition, but you need to evaluate this option thoroughly for your current and long-term needs. If you need help with your mobile app project, please check out our Mobile Apps Offerings and talk with us TODAY.

2.3 Hybrid Mobile Apps

Hybrid Apps are created using web technologies (HTML5, CSS3 and JavaScript) and a native shell application that leverages WebViews (embedded web browser) for rendering the web content and provides plugins that enable access to native functionality. Hybrid Mobile Apps are similar to Cross-Platform Apps in the sense that they share a single code base and support multiple device platforms. It is sometimes said that Hybrid development combines the best (or worst) of both the native and HTML5 worlds (source: Salesforce Developer Library), and I, personally, couldn't agree more with this statement.

For the most part, hybrid apps provide the best of both worlds. Existing web developers who have become gurus at optimizing JavaScript, pushing CSS to create beautiful layouts, and writing compliant HTML code that works on any platform can now create sophisticated mobile applications that don't sacrifice the cool native capabilities. One point to note is that not all Hybrid Apps are created the same, there is a wide spectrum that these apps can span from very ***webby*** to almost ***native.***

While you can create Hybrid Apps from first principles creating your own thin native wrapper, the more typical (and advisable) approach is to leverage tools like PhoneGap, Cordova or Ionic that take care of the basic plumbing and provide access to many of the native device features and APIs.

Considerations for Comparing the Options

In this section, you will discover the criteria that are most important while comparing the different mobile app development options. You will look at them from the lens of a Business Owner or Product Owner who defines the product vision of the mobile app under development, as well as the lens of the IT Executive/Manager or Technical Architect or Mobile Developer that is responsible for the technical implementation of the mobile app. Typically, the right choice of what mobile app development approach to take for YOU is rarely obvious and (almost) invariably requires a thorough trade-off analysis.

Business Owner

As a Business Owner or Product Owner, some of your key concerns are around functionality, budget, timeline, user experience, and business agility.

1.1 Strategic Importance of the App

Is the app you are building strategic to your business and does it provide a key differentiator over your competitors? Or is this app needed only to satisfy some RFP or regulatory requirement? The strategic vs. tactical importance of your app will determine the time and money you want to devote to this initiative. Is a ***webby*** hybrid app built using PhoneGap adequate or do you need the ***fully*** native app that is the ultimate Ferrari that roars down winding roads?

1.2 Your User Base

Who will be using your apps? The millennials and tech-savvy, discerning users who expect the smoothest animations/transition effects, augmented reality, drag-and-drop multitasking support and more? Or the consumer who simply needs the quick, easy-to-use app that enables him/her to view and pay his bill? Or an internal employee who needs an app to get his/her work done with high levels of productivity and doesn't care so much about design niceties? Understanding your users will help you make a more informed decision.

1.3 Your Competition

If you are in a very competitive market where you cannot afford to be late to the market with new features that take advantage of new device features (such as Face ID when iPhone X came out, or when the Apple Watch was released), then you may need to consider fully native apps. On the other hand, if your development speed / throughput will be lower with fully native apps because of the skills on your team, then when faced with competition and with time-to-market being a key consideration, you may need to consider an alternative to fully native apps such as cross-platform native apps.

1.4 Targeted Devices

Are you targeting only smartphone users? Or do you also want to support smart watches, TVs, and advanced features on tablets? If the answer is the latter, then you are more likely to encounter limitations when you stray away from fully native apps.

For example, it is difficult if not nearly impossible to implement advanced features such as a smartwatch UI, picture-in-picture tablet video, split-screen multitasking, and others using a non-native approach. Having said that, it is true that you can implement just about any feature using a bridge to native code, though that does add native code and the associated complexity that you were likely trying to avoid in the first place. The difference, however, is that this will be limited to the specific feature that you implement natively rather than the full app being native.

1.5 Implementation Costs

Money, money, money! This is always an important consideration as budgets are never unlimited and you have to take into account the available budget to roll out your product (i.e., the mobile app under development).

When budgets are extremely limited, you may need to look hard at hybrid apps; on the other hand, when the app is strategic and you have access to enough funding, you may be able to leverage fully native apps. However, cross-platform native apps provide good value for money and make sense for a wide range of budgets.

1.6 Time to Market

If you need to deliver the same/similar functionality through both websites and mobile apps, and time-to-market is a key consideration, then the Hybrid App approach may make sense so you can embed web-based features in your mobile app to roll the functionality out quickly.

And if you are trying to balance time-to-market and other factors, such as User Experience and Performance, then you may want to strongly consider Cross-Platform Native Apps.

1.7 Look and Feel

Are you a stickler for user experience (UX) and want your apps to be true to the UX guidelines and conventions specific to Android and iOS? Android and iOS applications generally shouldn't look and act the same as their user interface guidelines are completely different. If you don't mind ignoring many of the guidelines and developing one UI for both platforms, a hybrid or cross-platform application may be a good choice for you.

While React Native generally does a good job of abstracting User Interface differences between iOS and Android devices, there are cases where some React Native components provide an oversimplified user interface or worse, a user interface for either only iOS or Android.

1.8 Feature Parity

How important is it for you to maintain feature parity between your Android and iOS apps? In other words, when you roll out a feature for one platform, do you also want to make it available at the same time on the other platform? Feature parity is much easier to achieve with cross-platform native apps and hybrid apps, and takes significantly more effort and development team capacity to achieve with fully native apps.

IT Development Manager / Technical Architect

As a VP, Director, or Manager of IT, or Enterprise Architect / Mobile Developer, some of your key concerns are around developer capacity, app performance, native feature support, community interest and long-term viability of the development platform/approach, code reuse, and development agility.

2.1 Required Developer Skills

A key factor is knowing what knowledge / skills are required to be effective in developing apps using a particular development approach. It is almost impossible to find and retain developers that are not only polyglots but also have expertise with numerous development tools and platforms - the typical enterprise has needs for not just mobile developers, but also web and integration developers. Often, the budget-strapped IT team needs to do more with less, and share mobile/web developers.

If you have a team that is strong in the Microsoft technology stack and specifically C# skills, Xamarin may be a better choice. On the other hand, if your developers live and breathe JavaScript, then React Native or Ionic might fit you well. And if you have a large enough team that allows developers to build up and stay within their areas of expertise, then going fully native may be a viable option for you.

2.2 Developer Pool (For Hiring)

How strong is the available pool of mobile developer candidates that can work with the mobile app development approach you choose? Do you need the flexibility to share developers between your web and mobile teams? Finding a developer that knows Swift, Java, XCode, Android Studio, iOS, Android for mobile app development AND is also an expert in Web technologies such as HTML5, CSS and JavaScript is like looking for a unicorn. You need to know how much time, energy and resources you can allocate to finding the right developers.

2.3 Application Security

If security is critical, you would likely prefer a cross-platform or fully native application. For full control of offline support, data encryption, credential storage, certificate pinning, jailbreak detection, and other important concerns, consider a cross-platform or fully native application which will allow you to access more security features.

2.4 Code Size and Reusability

Code size matters! Having a larger code base requires more effort for maintenance and slows down developer productivity and onboarding, while also reducing development speed. Cross-platform native and hybrid apps can share code between multiple mobile platforms and result in smaller code bases.

Additionally, you can potentially share code between your mobile apps and web apps using some cross-platform app techniques resulting in even greater “code size savings”.

If you need help with achieving optimum code reuse between your React Native mobile app and your React-based web application, please contact XTIVIA to learn about our best practices and learn how we can work together using joint teams.

2.5 Native Feature Support

There are multiple aspects to consider when evaluating the breadth and depth of native feature support provided by the mobile app development options as follows:

- **Completeness** - can you use the native features you want to leverage with a given development approach? With fully native apps, you always have access to all the native device APIs and you are guaranteed to have complete access to all native features the moment they are released and the platform SDK supports it. However, with cross-platform native apps, you may not have access to all native features and may need to access them via a bridge that requires native development.
- **Speed** - how quickly are new features supported by the development tool / framework that you choose? With fully native apps, this is instantaneous. With the other approaches, it tends to depend on the complexity of the new feature / API and how active the chosen tool's developer / community is in updating the tool.
- **Extensions and plugins** - often, you may need an extension and/or plugin to access native features with cross-platform and hybrid apps. As these features change, it is important to evaluate how quickly the related extensions / plugins are updated to ensure continued support.

2.6 Performance

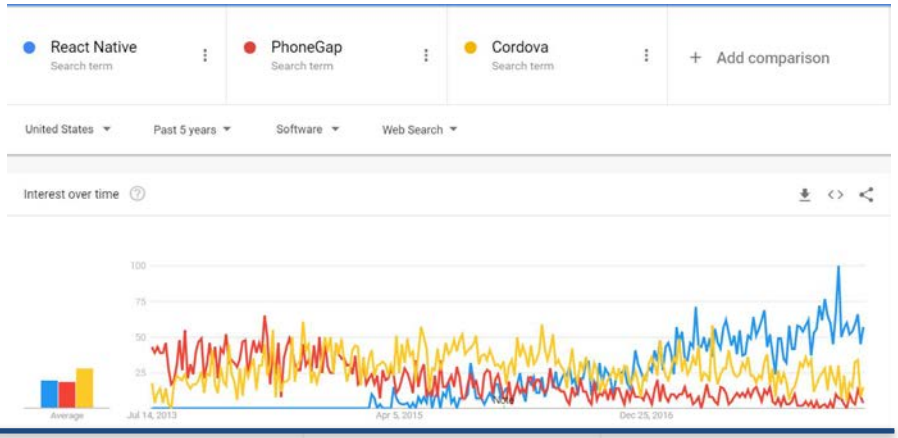
App Responsiveness and Performance matters - in some situations more than others. Often, enough though, if your app requires integration with various back-end systems, it is the responsiveness (or the lack thereof) of the back-end system APIs that will affect the end user's perception of your app's performance; this issue will affect all the mobile app development approaches equally. However, as mentioned earlier, fully native apps are the fastest, followed closely by the cross-platform native apps, and then the hybrid apps.

A specific issue with React Native apps is that they can be slower than fully native apps due to the additional overhead of the JavaScript runtime thread which live-translates your JavaScript into native view layout code. For many apps, this is imperceptible and, in general, in many enterprise mobile apps the mobile app development approach may not impact performance enough to matter.

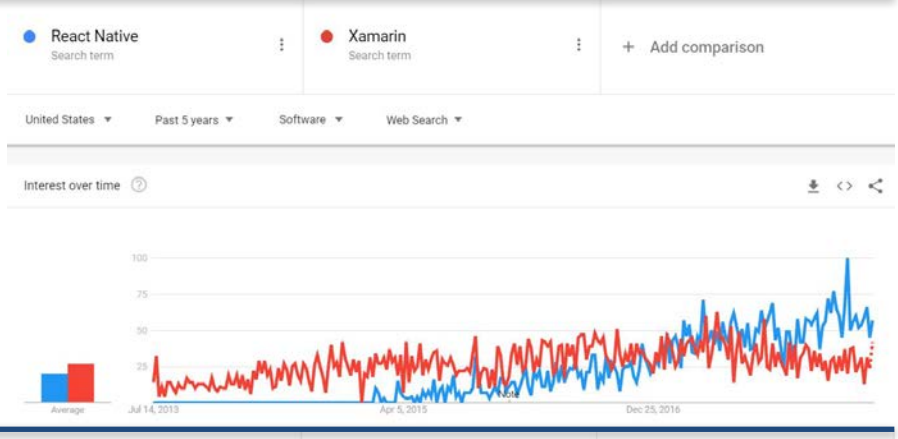
2.7 Developer Pool (For Hiring)

How vibrant is the developer community that is using the specific development approach / tool that you plan to use? How committed is that tool's developer to maintaining and enhancing that tool as the mobile devices and SDKs evolve? One way to measure community interest is to look at Google Search Trends as a data point.

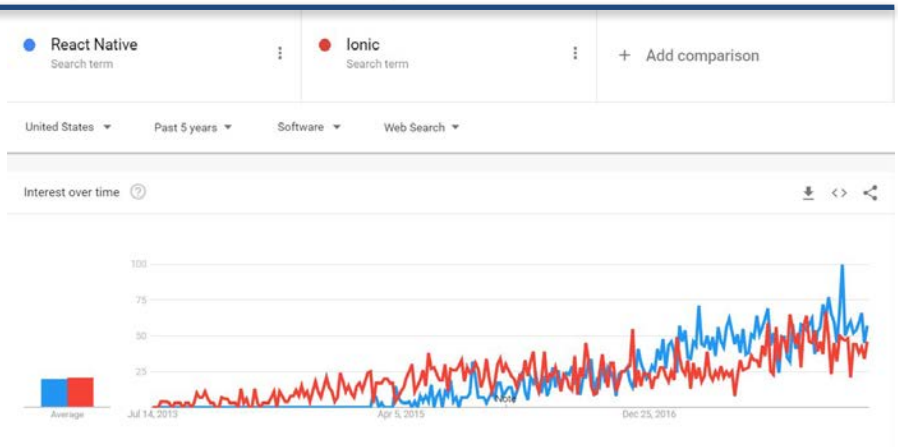
Comparing React Native with PhoneGap and Cordova (Q4'16-Q1'17 is when React Native takes over in search popularity and today React Native clearly outranks the other two)



Comparing React Native with Xamarin (Q1'17 is when React Native takes over in search popularity and today React Native is about 2x as popular as Xamarin from search perspective)

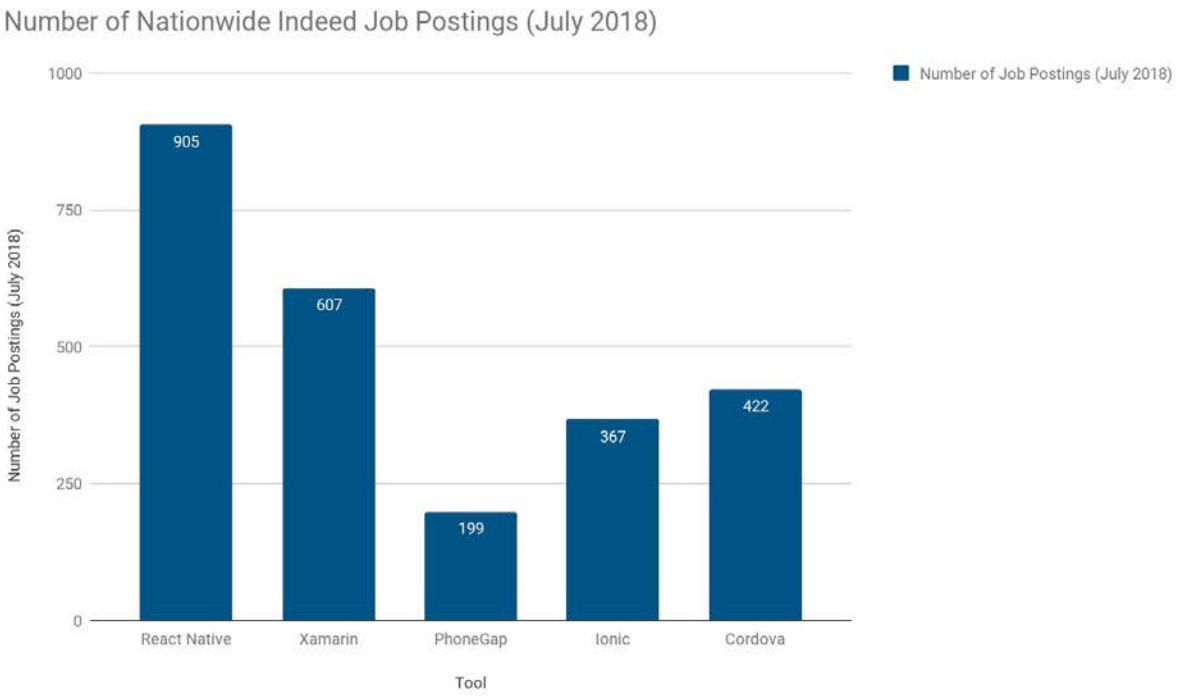


Comparing React Native with Ionic (Q1'17 is when React Native takes over in search popularity and today React Native is more popular than Ionic from a search perspective; over the last 12 months, React Native was about 35% more popular than Ionic)



2.8 Hiring Company Competition

The number of companies that are looking to hire Mobile App Developers with skills in the tool that you are considering is a good indication of the momentum behind a given tool. You can gauge this using a site like Indeed and running a nationwide search for jobs requiring that skill.



This data is based on a keyword search on Indeed.com with a tool name and "Mobile" such as Ionic Mobile or "React Native" Mobile, and as such is an approximation and not precise

2.9 Technology Stability

How committed are the organizations and developer communities to continuing their investment in the mobile app development tool / framework? Clearly, as long as Android and iOS are alive, Google and Apple will continue to invest in their corresponding native development tools and SDKs.

However, when it comes to the cross-platform native and hybrid tools, this is definitely something to consider.

Nonetheless, with the kind of broad-based usage they enjoy (with many well-known companies whose entire business model depends on their mobile apps), the risk becomes lower.

2.10 Development Agility

One of the biggest factors in developer productivity is the code-build-test cycle and how optimized it is with a given toolchain. For example, the development process with React Native is well-optimized, such that the moment you modify a line of React Native code and save it, the changes can be automatically seen in the app - this increases developer productivity significantly.

Another factor in developer productivity is the availability of Extensions & Libraries that serve various needs such as OAuth 2.0 or UI Components, and enhance developer productivity when available. The mobile development tools differ significantly when it comes to this consideration.

2.11 Application Security

You often need to launch other apps from the app you develop - for example, to launch the Maps app for directions, or the Mail app for email, or the Photos / Camera apps for photos. While it is easy enough to do this going from one native app to another, things are not always that easy in other cases and you then need to rely on a third-party extension.

2.12 Acceptance Into the App Store

This consideration is only applicable to hybrid mobile apps that do the bare minimum to meet App Store acceptance guidelines; per App Store Review Guideline 4.2: "***Your app should include features, content, and UI that elevate it beyond a repackaged website. If your app is not particularly useful, unique, or "app-like," it doesn't belong in the App Store.***"

If your hybrid app is particularly ***webby***, then you may run into objections from Apple. A hybrid app that incorporates push notifications and has a native menu while relying on WebViews for all content / actions, is still likely to get approved.

Comparing the Different Development Approaches

Now that you have seen a summary of the various criteria that you should consider (most of which you probably already intuitively knew) when evaluating the different mobile app development approaches, let us summarize how they compare on these criteria in a single view.

Criteria	Mobile Apps		
	Fully Native Apps	Cross-Platform Native Apps	Hybrid Apps
Strategic Importance of the App	5	4	3
Your User Base	5	4	3
Your Competition	It Depends	It Depends	It Depends
Targeted Devices	5	3	2
Implementation Costs	1	4	5
Time to Market	1	4	5
Look and Feel	5	4	3
Feature Parity	1	4	5
Required Developer Skills	1	4	5
Developer Pool (For Hiring)	2	5	5
Application Security	5	4	2
Code Size and Reusability	1	4	5
Native Feature Support	5	4	3
Performance	5	4	3
Developer Community Interest	5	5	3
Hiring Company Competition	5	5	4
Technology stability	5	3	3
Development Agility	1	4	5
Interaction with Other Apps	5	3	2
Acceptance Into the App Store	5	5	4

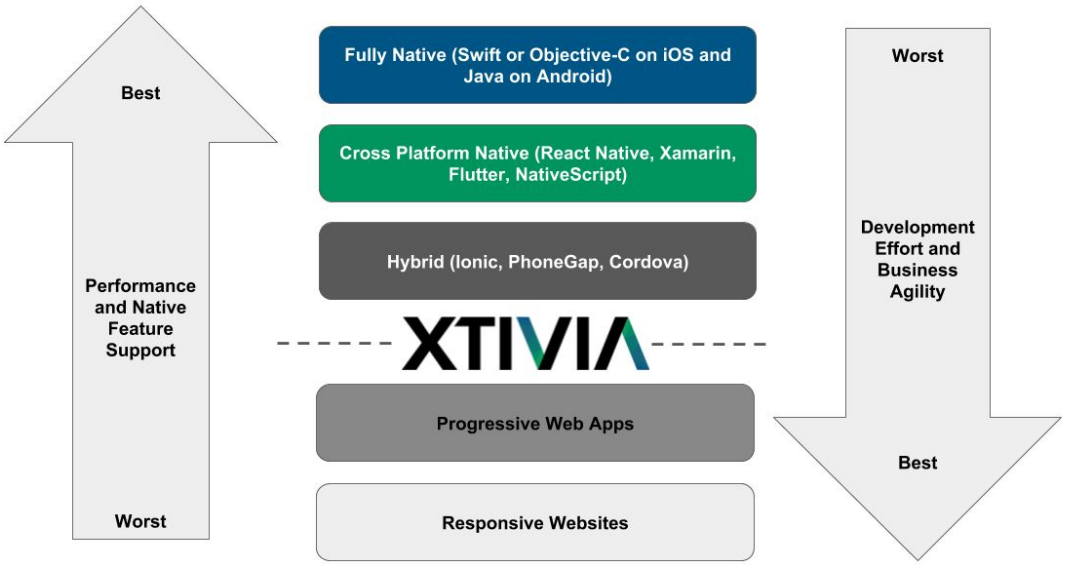
Legend	
5	Best
4	Excellent
3	Good
2	Average
1	Poor

Disclaimer: This scoring is certainly subjective and you can argue that a 4 should be a 3 or a 1 should be a 2, and so on. This is the reason why there is no total score included and you should only look at the relative scoring for a given criterion.

Summary

Summary

The mobile toolchain space is constantly evolving with new releases from Apple and Google, which introduce new features and functionality in each release of the iOS, as well as Android platforms. Additionally, the physical mobile devices continue to change and introduce their own innovations. The native toolchains, like Swift, XCode, Java and Android Studio ,are updated to support these enhancements immediately, while the Cross-Platform Native and Hybrid App toolchains typically lag behind by a few months. Web browsers are also evolving but not quite at the same pace - they tend to make leaps every few years.



You now have a summary of the most important trade-offs in this visualization - there are no black and white answers for ALL scenarios. You simply need to perform your own cost-benefit analysis taking into account your short-term needs and longer-term direction.

First and foremost, decide if you really need a mobile app or if a non-app based approach will do - Responsive Websites and Progressive Web Apps (PWAs are gaining more and more interest) may serve your needs in some scenarios.

Summary

If you decide that you need to develop a mobile app, the following will apply.

1. You will likely want to **go fully native** if you check off most, if not all of these boxes:
    a. Mobile developers and adequate development capacity to develop and maintain 2 separate iOS and Android apps
    b. The budget to build two apps
    c. Cannot compromise on access to native features and APIs
    d. Cannot afford to wait to have access to new features and APIs as soon as they become available
    e. Have a need for speed and a UX design that allows for no concessions
    f. Want to take no chances with a development toolchain that may lose its momentum and are unconvinced by the big names behind them (such as Facebook, Google, Adobe)
2. If fully native apps are not for you, then you will want to consider cross-platform native apps (using React Native or Xamarin or other similar tools) if:
    a. You want to leverage your existing web developers to develop and/or maintain the mobile app and prefer them using their current skills (JavaScript or C#)
    b. While you are willing to make some compromises, you want to be as close as possible to the native experience
    c. You want feature parity between your iOS and Android apps and want to increase ROI
    d. You want to optimize development time with fewer resources
3. If cross-platform and fully native apps are not for you, then you will want to consider Hybrid apps if:
    a. Your budget is extremely limited
    b. You are willing to compromise on the UX
    c. You want the highest possible reuse between your mobile and web apps
    d. Your app is not a strategic tool for your business nor is it a major differentiator against your competition

In our experience, for the typical enterprise mobile app, the cross-platform native or hyrbid apps approach makes the most sense, and we see a significant market momentum behind the React Native and Xamarin.

This eBook is brought to you by

your trusted partner for Mobile App Development.

Ideally, this eBook has given you information you can use to determine your best strategy. However, there is additional help available, through XTIVIA, to personalize solutions for your needs.

If you would like to discuss your Mobile App Strategy and/or need help with your Mobile App Implementation project, please reach out to a trusted advisor at XTIVIA here http://www.xtivia.com/contact/ or info@xtivia.com.

Thanks for reading

Choosing the Right Mobile Development Strategy for YOU

Need more guidance with mobile development?
Call 1-888-685-3101 ext. 2
Or visit xtivia.com for more info