



**altexsoft**  
software r&d engineering

**WHITEPAPER**

# **Pros & cons of native vs cross-platform mobile development with Xamarin**



# Native or Cross-Platform Mobile Development?

As the world is getting rapidly digitalized and global mobile data traffic grows extremely fast ([by 69 percent as of 2014](#)), the question is not about the need to create a multiplatform app, but rather about which tools and methods to use. [Android and iOS](#) continue to be the most supported mobile platforms, while WP8 is the platform that tops the users' wish list. According to the data provided by comScore, Android hovers at around **52 percent** penetration, the iPhone has a **42 percent** share and Windows Phones own **3 percent** of the market. So when considering your mobile development options, make sure you take into account all the three major platforms for your app to run on.

So the choice you will have to make regarding your app is whether to engineer it natively or use cross-platform mobile development tools. If you do decide to write separate versions of your app on each platform, your app will get truly **native** interfaces and great hardware-related features. Yet you will waste time replicating the app and spend a lot of money on the development and support. Assuming that you write the app in a single codebase and drop it into a **"magic box"** tool, the user experience and performance output is likely to be extremely poor. The third option would be using **Xamarin** as your cross-platform development tool. Its ultimate benefit is delivering native experiences while reducing costs and time to market. Anyhow, let us go through all the pros and cons of all the options together.





# Native mobile development as it is

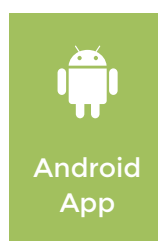
What you need when creating a native app is people having deep expertise with the language and API. Native mobile development results in truly native experiences that take the most out of the platform chosen. Thus, once you decide to provide your business with digital presence, you start leaning towards this exact option and quickly make the obvious choice in its favor.

Indeed, native environments are best fit for a number of applications. These are the [cases](#) when it's preferable to engineer a native mobile application:

- when developing mobile solutions based on video or game scenarios or apps with visually loaded design, navigation and animations;
- when integrating hardware-related features, such as gestures, multitouch events, geo-location tracking etc.;
- when developing utilities that manage system resources or Operating Systems (OS);
- when the solution requires processing large amounts of data on the client side.

However, the odds are that native mobile development might become the most **expensive** option for you. And the main reason for it would be quite limited code reusability. While building exclusive apps on the three major platforms such as iOS, Android and Windows, you staff three separate teams with different expertise; engineer all the features in different ways and then fix quite a number of different bugs. In addition, if you suddenly feel the market need to replicate the app on another platform, like Blackberry 10, you will have to start by writing specification requirements, gathering a team, and going through the entire development cycle all over again.

**Likewise, the challenge arises when your app requires some additional changes or enhancements.** You have to mobilize all the teams in order to complete a simple task like edit a text or modify a button color, which is quite wasteful. As a smart business owner or CTO, you would avoid all the time-consuming and cost-ineffective efforts. That is where cross-platform development tools enter the picture.





## “Write Once Run Anywhere” approach

Walking down this path of reasoning, next turn you see is most tempting: why not use “write once, run anywhere” approach with tools like **PhoneGap, Titanium, Sencha Touch?** The promise of it is about running on any operating system, of making the underlying code of the computer irrelevant. So you organize a single team to write a single piece of code, put it into “the magic box” adapting the app to the operating system and form factor of each supported device. Finally, the app runs across all the types of the devices. What could be simpler?

Yet it is not as good as it sounds. **What you need to take seriously is users’ expectations and preferences.** For instance, iOS users are used to a tab bar for navigation at the bottom of the screen, whereas Android users mostly use a side drawer in their apps. To enable corresponding experiences for your users on each platform, applying a completely cross-platform development approach is no good.

**As Xamarin chief Nat Friedman says, “by definition it [WORA approach] results in a mediocre experience.”**

The app you lastly get will have a number of platform-specific issues appearing here and there during the use on desktops, smartphones and tablets running different operating systems. Both the user experience and performance will suffer due to the limited use of each platform’s capabilities. Moreover, the framework’s vendor will take over control over the app’s features from your enterprise. In case the framework does fail to adopt the app, it leaves your mobile strategy at risk.



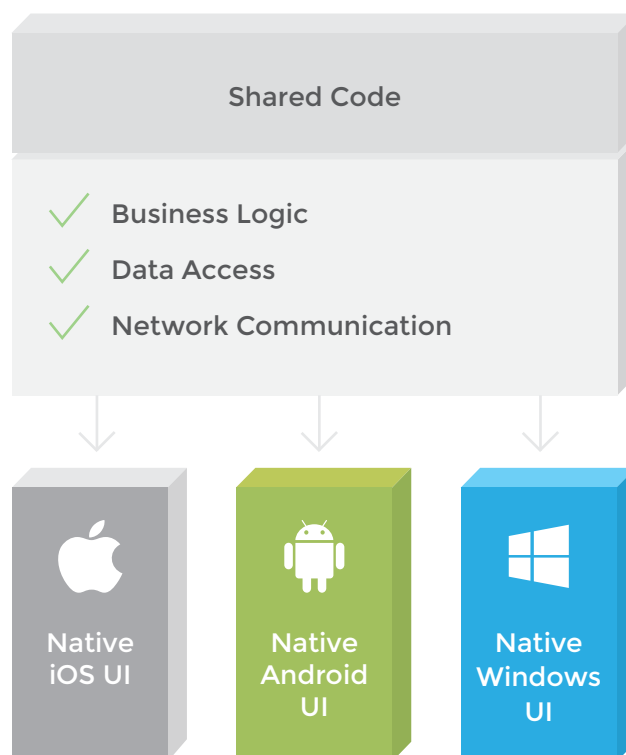


## ‘You-can-have-it-all’ solution

The real compromising option would be using a development model enabling advantages of the both extremes: time-optimized and cost-saving code share and reuse, and at the same time rich performance regardless of the device used. Xamarin is just the right mobile development platform answering these purposes. Here is why.

Xamarin mobile development platform combines flexibility in code reusing and advantageous native experiences. **The platform lets engineers integrate the core logic of your app in a portable code layer used for all the devices.** It in turn becomes the foundation for building different native user interfaces for each operating system separately. Developers have access to the full spectrum of the platform functionality (e.g. iBeacons, Android Fragments) and make the best use of it. As a result, the pages, layouts and controls are fully-featured, whereas the app looks native across devices and performs accordingly.

Due to the reasonable allocation of resources, Xamarin development is timesaving and cost-effective. Your only expert team works on the project in the familiar API, knowing exactly how the app will work and behave in advance. Xamarin controls are convenient and help



engineers enable a native functional product in a timely manner. **Your team develops an app with much of the code reused, single toolset available and APIs unified.** This cuts expenses on knowledge transferring, development framework configuration and deployment of various app's features. In other words, you get the attractive and well-performing app up and running faster and less expensive than when building several different native apps.



## Down to details: what is Xamarin and why use it

If you have not used Xamarin yet, you most likely have not heard of it. According to the research2guidance's Cross Platform Tool Benchmarking 2014, low awareness is the main barrier for making use of this convenient cross-platform development tool. So let's get familiar with its capabilities.

**The .NET based Xamarin platform allows developers to use C# in creating various apps for desktop/PC's, smartphones, tablets, smart TVs and even in-car devices.** Since C# is a simple, contemporary, multipurpose object-oriented language, you can take great advantage of strong typed language, lambdas, LINQ, async programming.

**Xamarin supports a number of operating systems, including iOS, Android, WP8, and Apple OS X.** As an additional service, you can

test your app on over 1000 mobile devices immediately in their Test Cloud that offers continuous integration, illustrative reports, test for fragmentation, and object-based UI testing. Moreover, you have an opportunity to analyze your app's performance and users' behavior with Xamarin Insights.

Once you get to know Xamarin capabilities, the [study](#) says there is a very slight chance for you to switch to its competitors. **User satisfaction for Xamarin is very high: 75%** of users are pleased with their pre-installed applications, access to device hardware features and Cloud API services. Currently, over 650,000 developers use Xamarin to engineer crucial enterprise and consumer mobile apps in a range of business verticals. The list of companies using Xamarin is quite impressive and lengthy. Have a look:

**BOSCH**



**GitHub**



**DOW JONES**



**McKESSON**

**nielsen**

**Honeywell**



**FOURSQUARE**

*Kellogg's*





## 8 ultimate advantages of Xamarin

1

**SIGNIFICANT REDUCTION IN TIME TO MARKET AND COSTS.** You save an enormous amount of time and money by avoiding cross-platforms code duplication. That enables programmers take time to do a number of programming tasks regarding database usage, network access and business logic. Investing in a single team lets you manage financial resources more effectively. Of even greater appeal, your app gains the advantage of entering the market ahead of the competition. While your rivals are still managing three teams working separately on building native apps, your product is already conquering customers.

2

**OVERALL RISK MITIGATION AND LOWER TCO.** As soon as you release the product, its market journey just begins. You will have to implement the app's enhancements, feature updates and modify the app taking into account new emerging technologies, which is costly in many ways. That is when Xamarin support will be of great value. Xamarin delivers new releases within two weeks after new device operating system features are available. Consequently, you app stays up-to-date at all times.

3

**WIDE FUNCTIONALITY INTRODUCED USING NATIVE APIS.** Xamarin's bindings equally correspond those of native environments. It means, if you request any native features for your app (e.g. GeoFence, accelerometer sensor, voice recording, push notifications), your development team can use native documentation to easily build them with C#. Additionally, your app gets the same visual appeal as in the development stage, so while creating the product you can be aware of how exactly it will look like in the end.

4

**APPEALING NATIVE USER INTERFACES.** Having standard native UI controls at hand, developers are able to create corresponding look & feel of the app's pages, customized to the specific mobile platform. To cite an example, engineers are able to use C# commands and refer to Apple's CocoaTouch SDK frameworks and Google's Android SDK as namespaces. Meanwhile, they can also get hold of Android Fragments, iBeacons, and similar platform-specific features. So Xamarin makes it easy for programmers to use C# syntax and access the platform-specific UI controls. Thanks to this helpful feature, your app gets rich native user experience and all the wide functionality you require.



5

**ROBUST NATIVE PERFORMANCE.** Mobile apps built with the help of Xamarin perform natively, because the tool leverages the platform-specific hardware acceleration. The C# code for iOS apps is ahead-of-time (AOT) compiled to assembly level ARM assembly language. The C# code for Android apps in turn is compiled to intermediate language (IL) and then packaged with MonoVM + JIT'ing. As for Windows Phone, you compile the C# code for it to IL and then it is executed by the runtime with no additional tools needed.

6

**GREATEST POSSIBLE BUG ELIMINATION.** With Xamarin, you can avoid introducing platform-specific errors due to the significant amount of the shared code. Furthermore, the single programming language and API allow developers spend less time hunting and fixing bugs. Therefore, you save time for engineers to integrate additional sophisticated functionality and polish the user experience.

7

**EXTENDED LANGUAGE CAPABILITIES.** C# has proved to be considerably beneficial development language over the years. Its up-to-date patterns such as generics, LINQ, implicit typing, extension methods, async methods, and closures provide the basis for building clear and solid code. Meanwhile, your iterations can be implemented very rapidly and safely, without interrupting the development process itself. If you do not need your app built from scratch, Xamarin also offers full integration with backend systems such as sales, inventory, customer service, supply chain, and CRM.

8

**SIMPLY IMPLEMENTED CORE LOGIC.** Xamarin makes engineering an app for enterprises or customers easy and convenient. The availability of .NET and other C# Libraries (for instance, HttpClient), supported by PCLS and Microsoft, along with the option to choose from the two integrated development environments (Visual Studio or Xamarin Studio) simplifies the job of developers. What you get in turn is almost limitless opportunities for creating complex apps to streamline business operations.





## Conclusion

In the light of the torrid growth of mobile digital market, your enterprise have to keep up with emerging technology trends and address arising challenges. To achieve that, first of all you app should provide engaging and rich native experiences to your customers across platforms and devices. When it comes to actually implementing your idea into a beneficial app, consider all your mobile development options. Determine the full spectrum of features your app will require and plan the development process accordingly. Among others, Xamarin can become a crucially helpful tool for you to make the best of cross-platform mobile development.

## References

<http://www.gartner.com/technology/topics/trends.jsp>

<http://research2guidance.com/cross-platform-tool-benchmarking-2014/>

<http://marketingland.com/smartphone-share-stabilizes-in-us-competitive-only-at-margins-128138>

<http://www.comscore.com/Insights/Market-Rankings/comScore-Reports-March-2015-US-Smartphone-Subscriber-Market-Share>

<https://tecordeon.wordpress.com/2015/03/19/is-xamarin-platform-a-better-way-for-creating-modern-mobile-apps-2/>

<http://www.seamgen.com/blog/reasons-to-use-xamarin/>

<http://crystalnix.com/blog/2014/12/30/does-xamarin-really-decrease-time-market-your-mobile-app/>

<http://www.theappbusiness.com/native-vs-hybrid-developing-cross-platform-mobile-apps/>

<http://www.willowtreeapps.com/blog/xamarin-or-native-development-tools-for-ios-android-projects/>

# About AltexSoft

[AltexSoft](#) is an innovative software R&D company focused on Web/SaaS/Cloud, Mobility, Big Data, and Machine Learning. With a proved track record of delivering innovative solutions and being industry-oriented, AltexSoft provides a range of services, including Custom software engineering, Data Science and IT consulting since 2007. AltexSoft has been accumulating expertise and providing custom IT solutions in a number of industries, including but not limited to: E-commerce & Retail, Healthcare & Life Sciences, Travel & Booking, Finance & Banking, Marketing & Advertising, Aviation & Transportation Systems.

**TO LEARN MORE ABOUT XAMARIN MOBILE DEVELOPMENT, FEEL FREE TO CONTACT US**

## US Sales HQ

140 Encinitas Blvd., Suite 134,  
Encinitas, CA 92024  
+1 (858) 449-1499  
[scott.holmen@altexsoft.com](mailto:scott.holmen@altexsoft.com)

## Global HQ

32 Pushkinskaya Str.,  
Kharkiv, Ukraine 61057  
+38 (057) 714-1537  
[sales@altexsoft.com](mailto:sales@altexsoft.com)