CROSS-PLATFORM MOBILE APP DEVELOPMENT WITH

XAMARIN

SAVING YOUR TIME, MONEY AND IMPROVING YOUR VISIBILITY

Mobile applications are becoming increasingly popular. According to Statista, in 2017, consumers downloaded 178.1 billion mobile apps to their connected devices. By 2022, this figure is projected to grow to 258.2 billion app downloads. In order to compete in this market, it is essential to have an app for multiple devices. Today, businesses that invest in mobile strategy need to consider the overwhelming number of mobile devices with different operating systems. It's no longer enough to just develop an app for one platform[2].

However, it is not that easy for a majority of companies, especially ones with a limited budget. Many are still struggling with the iOS vs. Android debate and sometimes need to consider just one platform due to costs. Luckily, there is an innovative solution for developing both apps simultaneously for less money. This is when cross-platform app development comes into play. It empowers developers to develop one app which can run on multiple platforms. Therefore, it saves the developers time as they need to write just one code, and at the same time, it saves the resources of the client.

One code for all platforms

A cross-platform application is an app with a single code base which can be used to run on different mobile operating systems like Android, iOS, Windows or BlackBerry. It performs like a native application and also has access to native libraries. The quality, performance and user experience are also comparable.

> **"The quality of the application depends on the quality and skills of the development team. There are also native apps that are poor quality,"** says Ruel Dognidon, a Principal Mobile Applications Engineer at ERNI in the consulting firm's Philippines office.

The only negative side is the size of the app. The generated app is larger than the native counterparts because it includes extra libraries. However, if you keep in mind that the performance difference between a native app and an app developed by multiplatform development tools is not very significant, the option to develop a cross-platform app becomes much more interesting.

5 reasons why to choose cross-platform app development

There are several benefits when we compare it to the development of native or hybrid applications:

1. Single development team - If you wanted to develop an app natively for both iOS and Android, you would need two separate teams. In cross-platform app development, you need just one.

2. Cost savings - If you are searching for developers of native apps, you will discover that they are quite hard to find and rather expensive. By having only one development team, you save money while increasing the company’s profit. You can also count on lower costs in the long run, since you only need to maintain one development team working to develop one codebase.

3. Time - Developing a native app separately for each platform is more time-consuming as the developer needs to write three codes for three different platforms. Writing just one codebase is much faster, which also means faster time to market.

4. Code sharing - In cross-platform app development, there is only one codebase – developers write the code only once and deploy it to all platforms where you want to spread your app.

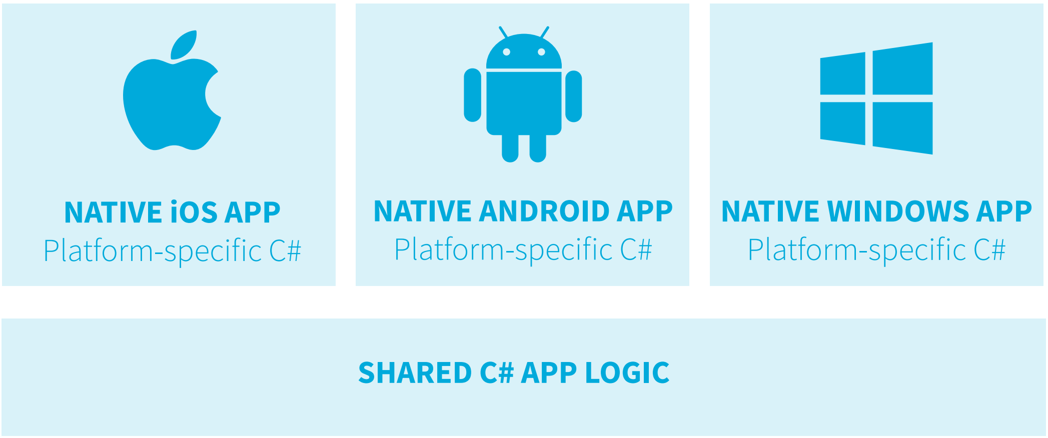
5. Bigger audience - If you see the cross-platform approach through, you can target a broader audience - one app can be deployed to iOS, Android, Windows Phone and so on.

Two types of cross-platform applications

In practice, we encounter two main types of cross-platform apps: cross-platform native and cross-platform hybrid. **Cross-platform native** leverages on better native application performance compared to hybrid because the code is compiled into native controls. Therefore, it feels and runs like a native app. It has full access to all native libraries that the platform can offer. Xamarin, React Native and NativeScript are the most common examples of native cross-platform tools. **Cross-platform hybrid** makes it possible to embed HTML5 (web) apps inside a thin native container. This native app container allows the web apps to access and leverage native platform features and device hardware. However, they are struggling to achieve a fully native look and feel, as they are basically WebViews. Therefore, this approach can sometimes result in performance issues when an app's UI is rich in components. Also, the hybrid applications themselves do not enjoy as much demand as native apps.

Xamarin as a gold standard

There are various technology approaches that can be used for building a cross-platform application, including JavaScript frameworks, wrappers, runtime library or source code translators. Xamarin is a source code translator and is considered the gold standard when it comes to cross-platform development tools. A .NET code or C# code is compiled into the native application for major application platforms like Android, iOS, UWP, Tizen and even iOS watches or Android wear. Before Xamarin, a developer had to learn three different programming languages in order to bring his product to three different mobile platforms and also had to know three different development environments. For the iOS platform it was Objective C, for Android Java and for mobile Windows platform C#. The Xamarin tool has made this process much easier and the developer just needs to use C# to create an application for all three platforms.



This tool has been owned by Microsoft since 2016. It originally came from the open source Mono Project created by the company Ximian and its developers Miguel De Icaza and Nat Friedman. The project was launched in 2001 with the aim to run .NET Framework applications on other platforms. Later, it found a continuation in a newly founded company called Xamarin. The name Xamarin pays tribute to its predecessor Mono (Spanish for monkey), combining the word Tamarin (a type of monkey) and the word Ximian (the X was carried over). After at least part of the original Mono team had moved to the new company and legally took official stewardship of the project, things started to accelerate.

Development of cross-platform apps with Xamarin is also a part of ERNI's Swiss software engineering portfolio and is considered one of their core competencies.

> **"We have been using Xamarin at ERNI since 2012, already before Microsoft acquired this platform,"** explains Ruel.

Even though there are some other tools that can be used for cross-platform app development such as React Native or Flutter, they prefer Xamarin. "Our customers are mainly enterprise companies or they want apps for business use and Xamarin can leverage on enterprise application development patterns." Moreover, it has been polished for more than seven years and has a very strong foundation from Mono. "We also use it because we have a lot of .NET enterprise application developers who can transfer their know-how and support Xamarin mobile developers," explains Ruel. As a result, they can share methodologies, patterns, coding guidelines and best practices as they are using the same coding language – C#. Therefore, can work as a one big team. Moreover, it has a strongly supported cloud feature, especially in Azure.

Two different approaches for developing with Xamarin

After making the decision to create a cross-platform app with the Xamarin tool, there is another decision to make – whether to develop it with Xamarin Native or Xamarin Forms. Each type has its advantages and disadvantages.

Xamarin Native uses the existing UI Framework on each platform. It includes Xamarin iOS, Xamarin Android, Xamarin UWP and even watches are supported on Mac OS. The application logic stays on C#, which can be shared on all platforms; only the UI and native libraries support is on the platform itself. Therefore, we still have sharing, which can reach up to 90 percent code sharing if there is not too much UI custom platform implementation.

Advantages:

Use of UI Design tool for each platform (Storyboard for iOS, UI Designer for Android, UI Designer for UWP).

All UI controls are easily exposed at your fingertips – whatever controls a platform can offer, developers can use them on Xamarin Native, especially when designing UI.

Disadvantages:

Developers need to design and implement each view for each platform. This means multiple work for the same app.

Since the developers use the UI design tool for the platform, they need some native development know-how, especially on the UI development.

The MVVM pattern (Model-View-ViewModel, a commonly used pattern by .NET developers) requires extra effort and third-party libraries need to be implemented.

Xamarin.Forms was developed one year later than Xamarin Native. "Right now, it has grown into a very stable platform which we lean towards when we are deciding on the architecture or the app implementation," says Ruel. It has a common UI Framework from all platforms it supports.

Advantages:

- Code sharing can reach up to 99 percent. Developers only write one UI implementation. There are not very many specific adjustments for each platform.

- It supports the MVVM development pattern out of the box. Therefore, .NET developers will feel familiar with Xamarin.Forms.

- UI Development is closely similar to WPF's UI development. Thus, .NET developers have an easier transition switching from WPF to Xamarin.Forms and can easily create a Xamarin.Forms mobile app without too much studying.

Disadvantages:

- No WYSIWYG (drag & drop) UI tool. There is a static preview, but it is limited to Android and iOS. Platforms like UWP, Tizen and Mac OS are not supported yet.

- Limited UI control kit. If you are targeting some specific look, then developers will need to make some adjustments. It is not impossible, but involves extra work.

Choosing between Xamarin Native and Xamarin.Forms depends on the budget and the UI preferences of the client.

"If the client wants an app that somehow looks almost the same on all platforms, we prefer Xamarin.Forms because we create only one UI. If he wants to have an app to have a native look and feel or resemble how the app would originally look on Android or iOS, then we would prefer Xamarin Native," says Ruel.

However, customers usually choose Xamarin.Forms because of the costing counterpart.

OUR CASE STUDY

Consumer App developed from scratch in one month

ERNI, as a strong expert in cross-platform app development with Xamarin, has already developed numerous applications using this platform. They have helped several businesses from various sectors to provide innovative solutions for their customers. They have, for instance, developed a cross-platform consumer app for a German world-leading chemical specialty company with 33,000 employees as part of their digitalisation strategy and for branding purposes.

> **The customer required a job fair mobile app which would increase participation and number of applicants in their booth during the job fair. They wanted to have it for two operating systems, but only four weeks were left before the event.**

Therefore, the developers had only one month for the project implementation, which included the idea, concept, development and launch.

In the end, ERNI delivered a mobile app developed using Xamarin. Forms for both the Android and iOS platforms, as well as an online web portal to manage app data. The project took four weeks and ran on a weekly sprint, with the first week dedicated to the design concept and requirements gathering. During the second week, developers focused on the backend integration. In the third stage, the web portal was created and developers did some bug fixing and retesting. On the final week, ERNI did an integration test, ran crowd testing and provided support in deployment to app stores. Moreover, offsite support was provided to the customer during the actual job fair. The application included such features as user registration, a company video and the option to take a quiz and receive the results. As an added value, ERNI also made app statistics available covering app usage, users and device analytics. And what was the outcome?

> **The Job Fair App was ready and well-received during the actual Job Fair, with a 25% download rate from the total attendees. The app also received high client satisfaction and was noticed by those at the main office in Germany, who became interested in rolling out the app to other offices.**

Based on this example, we can see the benefits of cross-platform app development in practice. Despite having little time, developers managed to develop an app for two operating systems with all the requirements and provided the customer with agile development, skills, flexibility, fast delivery and good quality.

Cross-platform app development slowly becoming the norm

With so many advantages, we may ask ourselves what the future holds for cross-platform app development. Will it replace native app development one day? Nowadays, all companies develop apps for at least the Android and iOS platforms, and there is a real need for cost-effective cross-platform development. We can also see the big players in the mobile world that are leading towards this trend. Microsoft acquired Xamarin; Google initiated the Flutter application platform; Facebook has React Native for cross-platform development. It is slowly becoming the norm and companies should consider their mobile strategy if they want to keep up with the continuing growth of mobile app downloads.

Patric Lengacher
Managing director
ERNI Philippines and APAC

Phone: +63 2 531 59 82
E-Mail: patric.lengacher@erni.ph