

How to Build Cross-Platform Mobile Apps



Nowadays the majority of businesses have to go mobile to reach a greater number of the target audience. Therefore, it becomes crucial for them to be presented on all possible mobile platforms.

Although native apps have proved to be fully interactive, intuitive, and perfectly consistent on all devices, it's time-consuming and costly to build two high quality, perfectly performing products.

In contrast, cross-platform mobile applications are gaining more popularity because they offer access to different mobile platforms using one codebase. What is more, due to the constant technological advancement, they have reached the same quality level as native apps, but with significantly reduced development cost.

This cross-platform mobile apps guide will walk you through the most popular tools for cross-platform app development and their key features.

Major Cross-Platform Mobile Development Tools

The market for cross-platform app development is growing fast. Therefore, the choice of cross-platform development tools is becoming wider so any developer can find one that fits their goals. Here are the most famous cross-platform frameworks.

React Native

React Native is an open-source framework owned by Facebook. According to Statista, it takes the lead among the most famous tools for cross-platform mobile app development. This framework allows developers to build native-like mobile apps for iOS and Android. React Native requires the knowledge of JavaScript and React.js to create user interfaces.

Cross-platform mobile frameworks used by software developers worldwide in 2019 and 2020

| | 2019 | 2020 | | 2019 | 2020 |
|--------------|------|------|----------------------|------|------|
| React Native | 42% | 42% | NativeScript | 11% | 5% |
| Flutter | 30% | 39% | Apache Flex | 5% | 2% |
| Cordova | 29% | 18% | Dojo | 4% | 1% |
| Ionic | 28% | 18% | Kotlin Multiplatform | - | 2% |
| Xamarin | 26% | 14% | Kivy | - | 1% |
| Unity | 12% | 11% | Corona | - | 1% |
| PhoneGap | 11% | 6% | | | |

Pros and Cons

As a JavaScript-based framework React Native has access to rich built-in libraries. It offers smooth integration with other projects and makes testing easier.

On the downside, React Native is a sophisticated framework. It needs a team of specially trained professionals as any error will cause a total project inconsistency.

Flutter

Flutter is an SDK introduced by Google. Although it is a new tool, it has already gained popularity among cross-platform programmers. The founders state that it can be used to develop a fully-customized application for mobile, web, and desktop platforms. This UI toolkit works on Dart and at the same time supports Swift, C, Java, and other programming languages.

Pros and Cons

Flutter ensures an easy and fast development process with access to a rich set of widgets. When it comes to drawbacks, this framework is used to create large-scale apps and doesn't provide enough support for Ready-to-Implement libraries.

If you're choosing between Flutter and React Native, you may want to check out this article explaining the pros and cons of each tool.

Xamarin

Xamarin allows developers to build mobile apps for the majority of platforms such as Android, iOS, Mac, and Windows. Although they all stand on a common codebase using C# there are no deficiencies in app quality. By using Xamarin developers can create unique UIs for one platform and then write some elements that can easily be used for others.

Xamarin lets developers write the UI code for a specific platform: one for Android using Xamarin.Android and another for iOS using Xamarin.iOS. After that, they should develop the business logic written in C# that can be applied on both platforms. Xamarin states that up to 90% of the code can be shared in this way.

Xamarin's IDE (Xamarin Studio) together with Visual Studio by Microsoft stands tall among market leaders and provides a high-level environment for efficient work.

Pros and Cons

Xamarin platform stands out with its “one language” approach. C# is one of the strongest points of the framework as it was significantly improved in comparison with other programming languages. Xamarin provides mobile cross-platform support with code sharing of up to 90%. It reduces both the development cost (since each new release of iOS or Android is complimented by Xamarin) and the time necessary for the app launch.

Speaking about some cons, working with Xamarin requires some specific knowledge. The developer should not only know C# and .NET programming but also have some basic skills in Java, Objective-C, Swift, and be familiar with Xamarin IDE and UI features. Xamarin doesn't have access to open-source libraries used for Android and iOS, nonetheless, it offers its own substitution.



Appcelerator

Appcelerator uses JavaScript codebase on Alloy MVC framework to build applications across platforms. It allows programmers to develop high-performance apps in a short time. Its SDK offers more than 5000 APIs for Android, iOS, Windows, HTML5, and Blackberry. Appcelerator fits best enterprise-level solutions as it offers a Virtual Private Cloud that meets their need for secure connections to corporate information.

Pros and Cons

Appcelerator makes the development process fleet. Programmers need to write some lines of code and spend a couple of minutes instead of long hours. Moreover, JavaScript makes the tool more attractive for developers as they don't have to study other languages.

When it comes to shortcomings, Appcelerator is notorious for being buggy and laggy. Although the tool has gained popularity due to its flexibility then developers start to notice many limitations. Neither is it as smooth and convenient as native app environments.

PhoneGap

One of the most popular frameworks, PhoneGap, belongs to Adobe. It operates on Apache Cordova and uses HTML5, CSS, and JavaScript for cross-platform app development. Apache Cordova provides access to PhoneGap Toolset which in turn offers developers new codes from the community base for free.

Pros and Cons

PhoneGap requires the knowledge of HTML5, CSS, and JavaScript which are the most famous programming languages. So, developers don't have to get new skills while businesses don't need to hire extra specialists. Mobile apps built in the PhoneGap framework have access to phone native resources as the tool interacts with mobile hardware. This framework is flexible and has all the necessary tutorials in free access.

On the contrary, PhoneGap doesn't support some functionalities and may have problems when dealing with native applications.

Sencha

Sencha framework uses HTML5 for developing web apps which can be then converted into native apps by PhoneGap or Sencha's SDK. Sencha offers a big variety of highly efficient products such as Sencha Architect, Sencha Touch Chart, Sencha Space, and Sencha Eclipse Plugin.

Pros and Cons

Sencha stands out with many sophisticated visual solutions making it possible to create complex and efficient apps. The framework includes a flexible layout manager and a reliable data package. It brings mobile apps closer to a native look offering a great variety of themes for all major platforms.

However, Sencha doesn't have access to some phone native property resources such as camera, accelerometer, and contacts and doesn't support push-notifications. Moreover, it has poor performance with animation apps.

Alternative Cross-Platform Editors and IDEs

For the businesses and developers who don't require sophisticated cross-platform features here is a list of alternative tools. Some of them can cover a big variety of tasks while others are used for more limited objectives.

Ionic

Over 5 million programmers use Ionic for building apps. It is an open and free source tool that offers a rich library of UI components. Ionic works on HTML, CSS, and JavaScript making it intuitive for many programmers. It has access to native phone features offering a big choice of plugins. Using a single codebase Ionic lets developers create Android, iOS, and Web apps.



Corona

Corona is a completely free framework mostly designed for building gaming apps for mobile phones and desktop systems. This cross-platform tool works on Lua, a scripting language used for many famous games (Angry Birds, Warcraft, etc). Corona offers a big variety of plugins and access to many native libraries.

Yapp

Yapp cross-platform tool helps to develop cross-platform applications for events, conferences, and meetings.

Xojo

Xojo framework deserves particular attention as it allows developers to create applications on a single codebase not only for Web, mobile devices, and desktops but also for Raspberry Pi. This cross-platform tool offers ample opportunities and drag-and-drop UI for quick app development.

AppsMoment

This framework is a perfect solution for newbies. AppsMoment has a wide choice of templates covering all the main categories. Its 200 templates keep the coding time at a minimum. It allows developers to publish their apps not only in Apple's and Android's stores but also in Window's and Amazon's ones.

Qt

Qt is a cross-platform IDE based on C++. This framework has an impressive list of companies working with them such as Mercedes, LG, Peugeot, and others. Qt provides access to a C++ library and offers a wide variety of APIs for improved mobile app development.

Dealing with Cross-Platform Development

Once you select a suitable cross-platform tool, it is necessary to remember that working with cross-platform frameworks has some peculiarities.



Dedicated API Services for Cross-Platform Mobile Apps

Cross-platform mobile apps work with back-end API. Such APIs sum all the functional details and use them for the back-end server. However, sometimes developers have to use a case-by-case approach to both the platform and the consumers. Here comes an idea of dedicated API services. Such an approach allows developers to change each back-end API separately making it possible to adjust the application for particular demands of mobile and web users. However, this approach implies some barriers. The most significant one is the ownership of individual APIs which often needs to restructure some organizational services.

Testing

When the app development is finished there comes the process of testing. This step ensures the elimination of some errors, prevents future technical issues, increases customer satisfaction, and finally results in application approval in app stores. Its main aim is to find out problems connected with the platform differences in configuration.

Testing is a time-consuming and monotonous process. Therefore, there is an option to automate it using one of the existing tools. It is time-saving and cost-cutting, but still, the human presence is preferred as some issues can't be scanned in this way.

The most common problems identified during the testing stage are connected with interface consistency and user expectations. Cross-platform mobile applications have to run smoothly on any of the platforms they were designed for and be user-friendly to meet the needs of all the customers.

Conclusion

Although everyone acknowledges that native apps offer robust user experience and optimized performance they are complicated and time-consuming to develop. In contrast, cross-platform app development requires less time and fewer resources. It is not only easier to develop cross-platform apps but also to maintain and update them.

Cross-platform tools are the future of mobile development. They allow businesses to reach a wider audience with fewer efforts and expenses. Moreover, a big choice of cross-platform frameworks makes it easier for developers to find a suitable tool that will fit best their needs.

Feel free to contact us and ask any questions: info@scand.com