mobidev

mobidev.biz    contact@mobidev.biz    +1 916 243 0946 (USA/Canada)

**Iurii Luchaninov,** Solution Architect at MobiDev

“Below you'll find a detailed comparison of cross-platform frameworks based on key development criteria such as performance, speed, cost, scalability, and more, backed by my 12+ years of hands-on experience. However, we must remember that technology is merely a tool. What truly matters is the expertise of the software engineers who work with it. If you are in doubt about which framework to move forward with for your product development, you can contact us to get tech consulting services to find the most suitable option to meet your business needs."

# Comparison of the Best Cross-Platform App Development Frameworks

| | Flutter | React Native | NativeScript | MAUI (Xamarin) |
|---|---|---|---|---|
| **Programming languages** | Dart | JavaScript, Java, Swift, Objective-C | JavaScript/TypeScript | C#/.Net |
| **Supported by** | Google | Meta | OpenJS Foundation | Microsoft |
| **Platform support** | Android, iOS, MacOS, Windows, Linux, Web | Android, iOS, MacOS, Windows, Web | Android, iOS, Web | Android, iOS, Mac, Windows, Web (limited) |
| **UI** | 🟩 Uses its own rendering engine, which allows you to create UI of any complexity and in some cases even faster than native ones. | 🟨 React Native UI is flexible enough to cover the majority of web cases. It doesn't have a full set of native component counterparts. Usually, they are substituted with community plugins. | 🟦 Similar to ReactNative, it builds on top of native views. It has the same issues as React Native. NativeScript allows using Angular 2+, Vue.js and their own UI framework to build UI. | 🟨 UI elements are based on native elements so it doesn't have a full set of native components counterparts. On the other hand, It has its own unique views and layouts. |
| **Performance** | 🟩 Performance is the same as native Android or iOS apps. This is due to Flutter architecture, which can be broken down into three components: a platform-specific embedder, an engine, and a framework layer. | 🟨 Even with the New Architecture, which is currently experimental, React Native isn't compiled into native code, and it still has the JavaScript layer, making it less performant than Flutter. | 🟦 It has a more limited performance compared to ReactNative since everything works on a JavaScript thread. | 🟨 MAUI is the superstructure above native SDK and UI that brings universality but causes degradation of performance. |
| **Customization** | 🟩 UI is extremely flexible and allows for building interfaces the same way native platforms do. We can define custom-painted views and custom layouts without the need to opt into native platforms. | 🟨 Has the ability to use custom paint but it's very limited and has very poor documentation. There is no way to define custom layouts without opting into a native platform. | 🟦 Doesn't have the ability to use custom paint and define new layouts. Has a feature to call native methods on views directly from JavaScript allowing you to make changes to views available on platforms. | 🟨 Has varied collections to customization, but if it requires building something out of scope, implementation becomes very complicated. |
| **Stability** | 🟩 Has stable releases that don't break apps and allows maintainers to react to required changes made by Google and Apple promptly. | 🟨 It’s in Beta. Has some breaking changes and issues related to new changes in native platforms from time to time. But overall, it has good stability. | 🟥 Has some issues with memory leaks if developers are not familiar with the framework on a qualified level and don't know how to mitigate them. | 🟨 May have some issues with the layout and component view on some platforms. Microsoft regularly produces new releases that bring both new features and resolve known issues. |
| **Web Compatibility** | 🟩 It has a full version for creating web and desktop applications. | 🟩 You can share only the logic part of the application with the web. Requires additional solutions. | 🟨 You can share only the logic part of the application with the web. | 🟥 Developers can use the Blazor framework to provide a bridge to use MAUI resources, but this is not a common case. |
| **Security** | 🟩 Relies on Dart’s built-in security features. Also allows you to use platform-specific security features. | 🟨 Relies on the built-in security features provided by the underlying mobile platforms (iOS and Android). Allows you to leverage platform-specific security features and security libraries available in the JS ecosystem. | 🟨 Leverages the security features of the underlying mobile platforms and uses security libraries available in the JavaScript ecosystem similar to React Native | 🟨 Leverages the security capabilities provided by the underlying platforms. Also offers secure storage options for sensitive data. |
| **Community and Long-term Support** | 🟩 Benefits from Google support and active community and extensive third-party support. | 🟩 Benefits from open-source nature and community development. | 🟥 Has a limited community compared to other frameworks. | 🟨 Benefits from Microsoft's support and investment. |

🟩 Good  🟨 Satisfactory  🟥 Poor  🟦 Average