



Iurii Luchaninov, Solution Architect at MobiDev

LINKEDIN

With more than a decade of expertise in developing and modernizing applications, at MobiDev, we have crafted our own approach and best practices to legacy app migration implementation. We recommend using the following checklist to verify that the process goes smoothly.

Comparison of Flutter vs React Native vs Capacitor as alternatives to Apache Cordova

Feature	Flutter	React Native	Capacitor
---------	---------	--------------	-----------

UI FEATURES

UI Component Library	Rich set of pre-built widgets	Basic set of components	Uses Ionic UI or custom components
Animation Support	Built-in, highly optimized	Basic, can use libraries	Basic with CSS animations
UI Development Speed	Very fast due to rich widget library	Fast with pre-built components	Moderate, depends on web skills

PERFORMANCE

	Flutter	React Native	Capacitor
Performance	Native performance	Near-native performance	Depends on web-view; generally lower than native due to hybrid approach
Animation Performance	Smooth and optimized	Generally smooth, might need optimization	Can be laggy due to web-based animations
CPU Usage	Efficient, optimized for performance	Efficient, but can spike with complex JS	Higher CPU usage due to hybrid nature
Memory Usage	Generally low, optimized memory handling	Moderate, varies with app complexity	Higher memory usage due to WebView

CUSTOMIZABILITY

	Flutter	React Native	Capacitor
UI Customization	High customization with widgets	Moderate, through JavaScript and CSS	High, depends on Ionic or custom components
Custom Animations	Highly customizable with built-in animation framework	Customizable, but might need third-party libraries for complex animations	Customizable using CSS or JavaScript-based animations
Platform-Specific Code	Easy to implement with platform-specific widgets and code	Supported, using conditional rendering and native modules	Supported, though may require more effort to manage different platform-specific implementations
Theming and Branding	Strong support with Material Design, Cupertino widgets and any custom design.	Moderate, relies on JavaScript solutions	Strong, especially with Ionic for theming

Here's a comparison table for Flutter, React Native, Capacitor, focusing on their native modules and plugins:

NATIVE MODULES AND PLUGINS

	Flutter	React Native	Capacitor
Native Module Support	Full access via platform channels	Near-Full access via native modules performance	Supports native plugins with JavaScript bindings
Plugin Ecosystem	Growing ecosystem, with many official and community plugins	Extensive ecosystem, vast collection of community-driven plugins	Moderate ecosystem, heavily reliant on the Ionic community
Ease of Creating Native Modules	Moderate; requires knowledge of Dart and platform-specific languages (Java, Kotlin, Swift)	Moderate to easy; uses JavaScript with native languages (Java, Kotlin, Swift)	Moderate; requires understanding of web, JavaScript, and native languages
Custom Plugin Development	Supported; requires familiarity with Dart and native development	Supported; relatively easier with JavaScript and native code	Supported; can be complex due to hybrid nature
Maintenance of Plugins	Official plugins are well-maintained; community plugins vary	Official and popular community plugins are well-maintained	Ionic plugins are well-maintained, others may vary
Cross-Platform Plugin Availability	Most plugins are cross-platform, but some may be platform-specific	Most plugins are cross-platform, with some platform-specific variations	Most plugins are cross-platform due to Ionic's web-based approach

COMMUNITY

	Flutter	React Native	Capacitor
Community Size	Large and rapidly growing	Very large and well-established	Moderate, growing with Ionic support
Community Contributions	Strong, with many community plugins, tutorials, and packages	Extensive, with a vast array of community-driven libraries and tools	Moderate, with a focus on web technologies and Ionic contributions
Community Tools	Numerous, including CLI tools, IDE extensions, and testing frameworks	Numerous, including CLI tools, IDE extensions, and various testing libraries	Supported by the Ionic ecosystem, with many web-focused tools
Corporate Backing	Backed by Google, providing strong support and development	Backed by Facebook (Meta), ensuring long-term support and development	Backed by Ionic, ensuring consistent updates and community support

SUSTAINABILITY

	Flutter	React Native	Capacitor
Ecosystem Stability	Rapidly growing, with a stable ecosystem and increasing adoption	Very stable, with a mature ecosystem and wide adoption	Moderately stable, with steady growth in the web and mobile hybrid space
Update Frequency	Frequent updates with new features, performance improvements, and bug fixes	Frequent updates, with regular improvements and a large community contributing	Regular updates, often tied to Ionic and web standards
Cross-Platform Evolution	Strong focus on cross-platform development, with continued enhancements	Continued evolution with a focus on cross-platform capabilities and native performance	Evolving with the web and hybrid technologies, with a focus on cross-platform use
Environmental Sustainability	Efficient use of resources, with a focus on optimizing performance across platforms	Efficient, but heavy reliance on JavaScript and React ecosystem can lead to resource consumption	Hybrid approach may lead to higher resource consumption due to reliance on WebView

MobiDev can help you with:

- Current application code and architecture assessment by providing a [software audit](#)
- Migration approach selection, taking into consideration business goal, technical limitations, budget and time restriction formed into a Tech Strategy with timelines, milestones, resource allocation, and risk management strategy
- Allocation of cross-platform experts as a [dedicated development team](#) or in-house [team augmentation](#) with Senior and Middle level engineers.

SCHEDULE A CALL WITH A MOBIDEV EXPERT

