OUR GUIDE TO

Native VS Hybrid App Development

gcd.

Ambitious Software

gcdtech.com
info@gcdtech.com

Everyone knows about apps, right? We all have them and use them every single day. But for those who are developing apps or those who are considering investing in a new business app, it's a more complex piece of software.

In fact, there are some big questions that must be asked when developing or asking for an app to be developed. One of those main questions is, should I use native or hybrid app development?

But why is it such an important consideration?

Because ultimately app development is **all about the users**, as we've already mentioned apps and mobile devices are part of our everyday life, so making sure users have a responsive and reliable experience is essential. No one has time for bad user experiences.

Deciding which app development approach to take is therefore an important one as each one offers a differing experience for users. In this guide, we'll take a look at the main ones plus a few additional new technologies that have been causing a scene - we're talking about you React and Flutter.

The Three Options

These are the three core approaches to app development that are widely used and as a custom software development company the ones we use most often when creating a solution for our clients. But each offers particular characteristics with specific outcomes, so deciding **which to choose will largely depend on the business objectives** and the overall goals of the project.

Some of those objectives might be:

- You want your users to have a **seamless experience** that fully integrates into the platform of their choice (Android or iOS).
- You need to **launch quickly on multiple platforms** and the experience doesn't need platform specifics.

We'll take a look at the advantages and disadvantages of each approach from a user experience perspective, from a developer point of view, a UX designer point of view and a business perspective.

But let's start at the beginning...

1. **WEB**
2. **NATIVE**
3. **HYBRID**

gcd.

gcd.

1

Web App Development

What is a web application?

According to Stackpath, "A web application is a computer program that utilizes web browsers and web technology to perform tasks over the Internet."

Typical languages for building a web application are **JavaScript**, **CSS** and **HTML5**. It's generally a straightforward build so it's great if you need a quick and simple solution with a wide reach. Server-side scripts handle the storage and client-side scripts present the information to users no matter where they are and what device they are using, this info could be surfaced as content management systems or online forms.

But is a web app the right choice for my business? I hear you ask. Let's take a look at the Pros and Cons for it.

PROS OF WEB APPS

- Less expensive than other options.
- Can run on all platforms.
- Easy maintenance as they use a common code base.
- They don't require approval from the app marketplace so can be released immediately.
- Because it doesn't have to be approved by the app marketplace updates don't have to be manually installed by the user, instead it auto updates when the web app is opened.
- They can take advantage of responsive design offering specific experiences for multiple screen sizes

CONS OF WEB APPS

- They rely on a browser, this means you're restricted when it comes to speed, metrics, features and hardware.
- Web apps can be less interactive
- The user has to take a few steps to access the web app, i.e. type in a URL, this may have an impact on user experience.
- Because they aren't in the app store, marketing has full responsibility getting it out there and making the most out of the branding opportunity.

gcd.

2

Native App Development

The next app software to be considered in your custom software journey is native app development - **developed specifically for a mobile operating system** using Objective-C or Swift for iOS and Java or Kotlin for Android. These languages are what the operating system (OS) use and they must also be built using their selected Integrated Development Environment (IDE).

The app is developed using the OS guidelines which means that technical features and user experience will be as close as possible to what the user is used to/expects. These app defined gestures will improve user adoption as the look and feel is consistent with their normal experience.

When you use the camera or message app on your phone those are native apps, building an app within a particular OS and all its guidelines is essentially what native development really means. Native apps have the additional advantage of being able to access the built-in capabilities of each device, such as the camera, microphone or GPS, giving the best performance.

PROS OF NATIVE APPS

- Developers have access to OS guidelines, full feature set and built-in capabilities giving native apps the best performance.
- Native apps are distributed via the App Store - improving visibility and providing support.
- The user is more familiar with the user experience of native apps so usability is improved.
- It's a more personalised product, which improves customer loyalty thanks to improved accessibility.
- Before reaching the app store, the native app must be approved by the operating system, improving QA and security.

CONS OF NATIVE APPS

- It's not easy developing native apps, they use difficult programming languages which require experienced developers.
- Native apps are more costly upfront compared to web or hybrid apps.
- They are not the best option for simple applications.
- Native apps may incur extra development costs when the platform is upgraded.

Despite its early high cost investment, in the long term native app development will save money with its high performance and great rate of user adoption thanks to its familiar User Interface (UI).

But What About Progressive Web Apps?

Here's an interesting segway - Progressive Web Apps (PWAs). What are they? Well they are a **blend of both hybrid and web apps**. They deliver an app-like experience to users, but this more advanced version of a web app means that features can be improved until the experience becomes more similar experience to a native app, features like push notifications, offline working, touch gestures and access to device hardware.

There are still some improvements to be made with PWAs, currently Apple's platform is more limited, it doesn't support push notifications, background sync or the web manifest file. But as we all know the world of development moves rapidly so soon this shouldn't be a concern.

There are a few things a Progressive Web App must have to be considered so:

- **Progressive** - Work for every user, regardless of browser choice, because they are built with progressive enhancement as a core tenet.
- **Responsive** - Fit any form factor, desktop, mobile, tablet.
- **Connectivity independent** - Enhanced with service workers to work offline or on low quality networks.
- **App-like** - Use the app-shell model to provide app-style navigation and interactions.
- **Safe** - Served via HTTPS to prevent snooping and ensure content has not been tampered with.
- **Discoverable** - Are identifiable as "applications" thanks to W3C manifests and service worker registration scope allowing search engines to find them.
- **Re-engageable** - Make re-engagement easy through features like push notifications.

You can read the full list over here.

gcd.

3

Hybrid App Development

We've covered web apps with their easy deployment, native apps with their smooth features and recognisable UI and Progressive Web Apps with their combination approach. But now it's time to take a look at a third app development option - hybrid app development.

What is a hybrid application?

As the word 'hybrid' suggests they are a **mix of web app with a native shell.** They work across multiple platforms and have native features. It can be installed on a device and is largely built in HTML, CSS and JavaScript. A hybrid app uses a native shell which is downloadable and loads the code via Webview.

PROS OF HYBRID APP DEVELOPMENT

- Hybrid apps don't need a web browser like web apps.
- They have access to a device's internal APIs and device hardware.
- Only one codebase is needed.
- Consistent UI language across multiple platforms maintaining brand consistency.

CONS OF HYBRID APP DEVELOPMENT

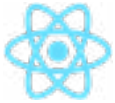
- Hybrid apps are much slower than native apps, their performance can also suffer as they load in webview.
- You're dependent on a third-party platform to deploy the app's shell.
- The more customisation the app requires the higher the costs can be, costs that can be saved with native app development.
- As it is a hybrid there will be issues that stem from both native systems and hybrid systems, which makes bug fixing more difficult.
- User experience is often sacrificed with a hybrid app in comparison to a native experience.

Is There Another Development Path?

We've established that native provides better performance and User Experience, with hybrid providing a flexible mix of web app and native with limitations that can be more cost effective. But within native app development there have been some new approaches…

Two platforms, in particular, Xamarin and React Native have both been heralded as another option within native app development.

Xamarin lets developers build one app that runs on many platforms using C# with either Visual Studio or Xamarin Studio. With C#-shared codebase, IDE, language and APIs developers can use Xamarin tools to write native iOS, Android, and Windows applications with native user interfaces and share code across multiple platforms.



React Native (something we love to work with at GCD) also allows developers to build authentic native iOS and Android apps with one codebase. With React Native, developers can create a mobile app that's identical to a product developed using either Objective-C or Java. React Native, however, is written in JavaScript and React.

Xamarin and React Native technology is very complex and can be classified as either native or hybrid.

Made by Google

The newest kid on the development block is Flutter, Google's UI framework released in May 2017.

Flutter gives developers the ability to create various native applications with only one codebase. Put simply: one programming language and one codebase but you can create apps for iOS, Android, Web and Desktop.

There are two parts to Flutter:

1. **A SOFTWARE DEVELOPMENT KIT (SDK):**

   Which is a collection of tools that help developers create apps, including tools to compile the code into the native machine code, iOS, Android, Web, Desktop.

2. **A FRAMEWORK:**

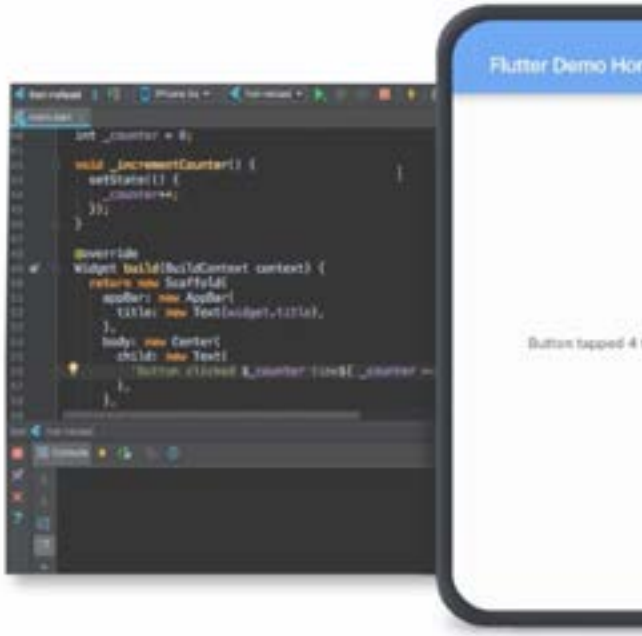
   Which is a collection of reusable UI elements (buttons, text inputs, sliders, and so on) that you can build from and customise to your own needs or the project specification.

Flutter uses its own programming language, also developed by Google, called Dart. It's a typed object programming language with a similar syntax to JavaScript. It focuses on front-end development and can be used to create both mobile and web applications, however, it is versatile and there are tools to develop back-end.

We had a quick chat with one of our own resident Flutter experts on the pros and cons of Flutter:

PROS OF FLUTTER

- Widget philosophy
- Really easy to use
- Good community support
- Developed by Google



CONS OF FLUTTER

- A relatively big learning curve.
- People who are already familiar with JavaScript or TypeScript aren't tempted to learn a new language but most fail to realise that it is indeed quite similar
- It isn't the best for multithreading

So there you have it our guide to the Native Vs Hybrid discussion, as you have hopefully picked up, it really depends on the individual business. Each development approach has its own positives and negatives some of which will be deal breakers for you, but with a constantly evolving landscape who knows what the next big thing will be. Right, now focus on that balance between business and providing your users with the best user experience possible.

If you've any questions on any of these please do reach out to us, we have a talented team of developers and designers who love complex problems and finding custom solutions to fix them

gcd.

Ambitious
Software

gcdtech.com
info@gcdtech.com