Alex Wacker

SWDV-665

PWAs vs Native App Development

For many years, the only way to develop an application for a user to consume was through a native application using a programming language for that specific platform. That process worked well for creating tailor-made experiences that ran great and had access to all of a device’s capabilities. However, given the higher cost of native app development, developers began looking for alternatives to still be able to create content for their audiences, but ultimately not break the bank. Progressive Web Applications, or PWAs, were the answer to this problem. I will outline the advantages and disadvantages between native app development and Progressive Web Application development.

Progressive Web Apps, as we know them today, really took off around 2015 when web apps were able to take advantage of new features like service workers. Service workers are the key component that allow a web app to access things like offline viewing, background data syncing, and push API responses. It makes the web app feel much more like its native cousin.

One of the main advantages of PWAs that was hinted at above is the decreased development cost. A native application has loads of service and maintenance fees associated with it. According to Savvycomsoftware.com, a project could cost upwards of $140,000[1]. A Progressive Web Application in general will cost a fraction of that. The build process is typically much quicker as well for PWAs. For a native app built for Android and iOS, both native code bases have to be built separately. For a PWA, there is only one website to build and maintain. User retention is also better for PWAs, given the barrier to entry is much smaller. A native app requires the user to go into their platforms respective store, wait to download and install it, then load the app. A PWA doesn’t require a download at all. A simple web bookmark will allow a user to save the PWA endpoint, and it will even show up on their home screen and app draw like a native application installed from the App Store. This method also has bonuses when it comes to storage space. A native app can take up a considerable amount of space, whereas a PWA is basically just a URL endpoint that takes up negligible room on the storage.

With all that said, there are of course some disadvantages of Progressive Web Apps. Security is one such point. It’s not that PWAs are considered unsecure. In fact, they’re much more secure than a standard web page, since they require the use of HTTPS. However, a native app has more options and flexibility than a PWA does. They can implement Multi-factor Authentication, certificate pinning, and other measures not directly available to PWAs. Performance is another comparison point that Native applications excel at. PWAs are much faster compared to traditional web pages, however, they are still running in a browser that runs through Service Workers tied into the hardware’s systems. A native app can tie directly into the operating system and minimize any performance costs much easier than a PWA can.

Progressive Web Application are a much needed evolution in app development for their cost and simplicity in using. Depending on the use case, any business should weigh the pros and cons of whether to develop on a PWA based platform or go the native code route.