

Wie sinnvoll und effizient sind Hybrid Apps?

Dominic Beusch, Wirtschaftsinformatiker, exec. MBA in Digital Transformation

Sie möchten eine neue Applikation entwickeln, die sowohl desktop- als auch mobilefähig ist, aber dafür nicht zwei unterschiedliche Applikationen programmieren? Soll der Anwender lokale Hardware wie beispielsweise eine Kamera benutzen können und Push-Notifikationen, also Benachrichtigungen, erhalten oder gibt es Einschränkungen? Welche Hardwarefunktionen benötigt der Anwender im Alltag und wie steht es mit der Sicherheit? Die heutige Technologie bietet mehrere Ansätze, unterschiedliche Anforderungen effizient und kostensparend anzugehen, so dass die Anwender Freude an der Nutzung der Applikation haben. Wir unterscheiden zwischen folgenden Varianten:

Native App vs. Progressive Web Application (PWA) vs. Hybrid App

Native Apps sind Anwendungen auf mobilen Endgeräten, die speziell für dessen Betriebssystem (iOS oder Android) konzipiert und entwickelt werden. Native-App-Komponenten werden verwendet, um die Benutzerschnittstelle (User Interface) so darzustellen, dass sich der Benutzer auf seinem Betriebssystem «zu Hause» fühlt. Native Apps besitzen optimierte Designs und Funktionen für das jeweilige Betriebssystem. Des Weiteren bieten sie im Rahmen der Sicherheit gute Möglichkeiten, die darin enthaltenen Daten zu schützen.

Eine **Progressive Web Application (PWA)** ist eine vollständig browserbasierte Anwendung, die auf einem Webserver betrieben wird und Aufgaben über das Internet unter Verwendung von Technologien wie JavaScript ausführt. User müssen auf eine Webanwendung über einen Browser zugreifen, unabhängig davon, welches Gerät beziehungsweise, welchen Browser sie verwenden. Das Benutzerinterface ist responsive, sprich, es passt sich der Grösse des Screens an. Mit dieser Variante sind grosse Einsparungen möglich, falls man nur eine Applikation bauen und die Nutzung auf unterschiedlichen Geräten anbieten möchte.

Im Wesentlichen ist eine **Hybrid App** eine Browseranwendung in nativem Gewand, das heisst, sie vereint beide oben genannten Technologien. Hybride Apps basieren auf gängigen Web-Technologien wie HTML5, CSS oder JavaScript. Sie laufen in einem Webview Container, in dem sie als native Apps verpackt werden und greifen somit auf native APIs (Schnittstellen) und Funktionen des Betriebssystems zu. Dies bedeutet: Die Hybrid App kann innerhalb ihres Containers ebenfalls auf eine responsive Website zugreifen, dies ist für den Anwender jedoch nicht sichtbar. Diese Variante ermöglicht es, die Vorteile beider Welten (Native App und Web App/PWA) zu nutzen.

Zum Vergleich der drei Varianten zeigt die folgende Aufstellung mögliche Vor- und Nachteile:

	Native App	Web App (PWA)	Hybrid App
Kosten	\$\$\$	\$	\$\$
Entwicklungszeit	🕒🕒🕒	🕒	🕒🕒
Performance	✅✅✅	✅✅	✅✅
User Experience	✅✅✅	✅✅	✅✅
Plattformunabhängige Entwicklung	❌	✅	✅
Push-Notifikationen	✅	✅	✅
Gerätefunktionen	✅✅✅	✅✅	✅✅✅
Sicherheitsfeatures	✅✅✅	✅✅	✅✅✅
Offline-Verfügbarkeit	✅✅✅	✅✅	✅✅✅
App Store Upload möglich	✅	❌	✅
Betriebsarm	✅	✅✅✅	✅✅

Der Kompromiss ist klar erkennbar: Zeit und Kosten versus Funktionen und User Experience

Einsatzzwecke und Zielgruppen – zwei Beispiele aus der Praxis

Usecase 1: Eine Applikation für Mitarbeitende mit hohen Sicherheitsanforderungen

Grobe Ziele und Anforderungen:

- Eine webbasierte Applikation, die nur Mitarbeitenden zur Verfügung stehen soll.
- Diese soll auf dem Desktop und auf dem iPhone lauffähig sein.
- Die Usability und die Performance müssen hoch sein.
- Es bestehen hohe Anforderungen an das Sicherheitslevel, da die Geräte teilweise auch privat genutzt werden.
- Das Nutzen einer Kamera und der Taschenlampe soll möglich sein.
- Die Applikation soll mit einem Mobile Device Management (MDM) in Bezug auf die Softwareverteilung verwaltet werden.

Das Kernkriterium hierbei: Die Applikation muss gewisse Sicherheitsfeatures besitzen, da die Nutzer auf personenbezogene und schützenswerte Daten zugreifen können. Einerseits war Geolocation ein Thema, andererseits musste die Möglichkeit bestehen, dass die Applikation in einem geschützten Container läuft, so dass keine Daten ausserhalb der Applikation ausgelesen und z.B. an Apple gesendet werden können. Der Kunde hat sich in diesem Fall für eine **Hybrid App** entschieden, da er

durch diese Lösung die Sicherheitsfeatures bewerkstelligen und dennoch die responsive Version der Web-Applikation nutzen kann. Damit spart er viel Zeit, Ressourcen und Geld.

Usecase 2: Web-Applikation und native App sind bereits vorhanden, doch es sollen Redundanzen eliminiert und Kosten eingespart werden.

Grobe Ziele und Anforderungen:

- Die webbasierte Applikation soll Cross-Plattform-fähig sein und somit auf den gängigen Geräten problemlos funktionieren.
- Usability und Performance müssen hoch sein.
- Das Nutzen der Kamera und das Erhalten von Push-Notifikationen müssen möglich sein.
- Die Wiederverwendbarkeit der bestehenden Web-Applikation muss maximiert werden.
- Die Betriebskosten müssen so tief wie möglich sein.

Diese responsive Web-Applikation läuft seit ein paar Jahren, das heisst, sie ist bereits für einige Geräte optimiert. Nichtsdestotrotz gibt es aus historischen Gründen nach wie vor eine native App, die aus Kosten- und Ressourcengründen in naher Zukunft deaktiviert werden soll. Die Analyse zeigt, dass die bestehende Web-Applikation einfach in eine **Progressive Web App (PWA)** umgewandelt werden kann und dabei alle Anforderungen erfüllt – in diesem Fall das Kriterium, dass die Applikation auf unterschiedlichen Screen-Grössen und Geräten genutzt werden kann. Die erste Version wird im Rahmen einer Publizierung getestet und bei Erfolg die native App deaktiviert. Damit lassen sich Geld und Ressourcen für die Weiterentwicklung der App einsparen beziehungsweise anderweitig einsetzen.

Key-Take-Away

Es gibt bewährte Technologien und Methoden, um Applikationen für mehrere Anwendungsfälle effizient zu entwickeln und zu unterhalten. Dazu müssen Applikationen aber nicht zwingend für mehrere Plattformen erstellt werden, was unterschiedliches Knowhow, unterschiedliche Kompetenzen und unterschiedliche Entwicklungsarbeit bedingt. Wie «Usecase 1» zeigt, kann eine Hybrid App eine sinnvolle und effiziente Variante sein.