

« Progressive Web Apps versprechen das Beste aus zwei Welten »

Mit Progressive Web Apps (PWAs) verschwimmen die Grenzen zwischen responsiven Websites und Native-Apps. Doch nicht für jeden Zweck ist eine PWA die bessere Anwendung. Wo die Stärken und Schwächen von PWAs liegen und warum offene Standards wichtig sind, erklärt Phil Hofmann, Dozent an der ZHAW und Gründer von 200ok. Interview: Joël Orizet

Was finden Sie an PWAs spannend?

Phil Hofmann: PWAs versprechen das Beste aus zwei Welten: Es stehen immer mehr APIs zur Verfügung, mit denen Webapplikationen die nativen Features der Endgeräte nutzen können. Im Entwicklungsprozess kann man die Vorteile von Webtechnologien voll nutzen und sich dabei die teilweise mühseligen Mechanismen zur Veröffentlichung nativer Apps sparen. Zudem sind die PWAs niederschwelliger, da die App nicht erst über den Store gefunden und installiert werden muss. Etablierte SEO-Strategien können daher im vollen Umfang zum Bewerben von PWAs genutzt werden.

PWAs sollen also das Beste aus der Native-App- und der Web-Welt vereinen. Geht das Versprechen Ihrer Ansicht nach auf?

Das kommt wie immer auf den Anwendungsfall an. Manche Features lassen sich schon ganz gut nutzen, bei anderen hakt es noch. Aber die Möglichkeiten von PWAs werden immer vielseitiger. Gerade vor ein paar Wochen wurde die experimentelle «Web Share Target API» veröffentlicht. Mit ihr ist es nun nicht nur möglich, aus einer PWA das native Sharing-Menü zu öffnen, sondern ebenso die eigene PWA für dieses Menü zu registrieren. Ein Feature, das ich in PWAs schon lange vermisst habe.

Was ist in technischer Hinsicht neu an PWAs? Was macht sie «progressiv»?

Der Begriff «progressiv» wird in diesem Zusammenhang häufig wegen des Konzepts «progressive enhancement» verwendet, das sich vielleicht am besten mit «fortlaufende Verbesserung» übersetzen lässt. Die Basisfunktionalität wird dabei mit Standard-Webtechnologien aufgebaut, die mit steigendem Zugriff auf native Funktionalitäten kontinuierlich verbessert werden. Aus meiner Sicht ist das die positive Formulierung des Konzepts «graceful degradation», das es im Zusammenhang mit Webtechnologien schon lange gibt und das sich mit «eleganter Funktionsverlust» übersetzen lässt. Der Begriff wurde in einer Zeit geprägt, in der die Minimalstandards der Browser in Bezug auf Funktionalitäten, aber auch Sicherheit, noch wesentlich geringer waren und man sich sorgte, dass Nutzer

in der Konfiguration ihrer Browser Javascript deaktiviert haben könnten. In diesem Fall sollte die Website dann trotzdem noch funktionieren. Darüber spricht heute keiner mehr. Javascript nutzen alle. Die Frage ist, welche APIs unterstützt der Browser, auf dem die PWA gerade läuft.

Sehen Sie PWAs als mögliche Alternative oder eher als Ergänzung zu Webapplikationen?

Das ist völlig abhängig vom Anwendungsfall. Es gibt Fälle, die sich aus technischen Gründen nur mit einer Mobile-App umsetzen lassen, aber diese Fälle gibt es ebenso für Webapplikationen. Manchmal braucht es beides. Im Grunde sind es Technologien, die sich gut ergänzen.

Was können PWAs besser als native Apps und klassische Webapplikationen?

Es gibt eine Statistik, die man in diesem Zusammenhang immer wieder hört: «Der durchschnittliche Benutzer installiert 0 Apps pro Tag.» Dadurch, dass PWAs letztlich nur Webseiten sind, verspricht man sich eine grössere Reichweite, weil der Installationsvorgang viel niederschwelliger ist. PWAs sind ausserdem deutlich kleiner als kompilierte Native-Apps, was das Installationserlebnis weiter verbessert und Bandbreite beziehungsweise Datenvolumen spart. Updates von PWAs sind daher auch weniger schwerfällig als Updates von nativen Apps über den Store. Grundsätzlich begrüsse ich die Möglichkeit zum Einsatz der offenen Standards, auf denen auch das Web aufgebaut ist, und sehe mehr Potenzial bei der Integration von PWAs mit Webapplikationen und untereinander. Das Fehlen von gemeinsamen Standards hemmt diese Entwicklung im Kontext nativer Apps.

Was sind die Nachteile?

Die grosse Unbekannte ist die Verfügbarkeit von APIs. Auf den proprietären Plattformen fällt es den grossen Playern offensichtlich leichter, sich auf eine stabile API festzulegen. Bei den APIs, gegen die man mit einer PWA programmiert, bleibt einem oft nur zu hoffen, dass diese weiter unterstützt werden. Ein weiterer, häufig diskutierter Nachteil ist, dass abhängig davon, wie man seine Zielgruppe

« Die grosse Unbekannte ist die Verfügbarkeit von APIs. »

Phil Hofmann, Dozent an der ZHAW und Gründer von 200ok



« Bei Projekten, für die der Funktionsumfang im Vorfeld nicht klar abgegrenzt werden kann, kann sich die Entwicklung einer PWA als Sackgasse erweisen. »

Phil Hofmann, Dozent an der ZHAW
und Gründer von 200ok

erreicht, eine Präsenz im Store aber wünschenswert ist. Manche Zielgruppen erwarten eine Native-App und suchen direkt im Store. Google hat unlängst auf diesen Einwand reagiert und bietet seit Kurzem auch die Möglichkeit, PWAs in den Google Play Store zu stellen – eine etwas absurde, aber nachvollziehbare Entwicklung.

Für welche Anwendungen eignen sich PWAs besonders?

Bei Verfügbarkeit der notwendigen APIs lassen sich Anwendungen mit beschränktem und klar abgegrenztem Funktionsumfang, abgesehen von ein paar Spezialfällen, gut mit PWAs umsetzen.

Was sind das für Spezialfälle?

Spezialfälle könnten Anwendungen in Bereichen sein, in denen es beispielsweise auf Leistung oder Sicherheit ankommt. In diesen Fällen können die offenen Standards des Webs mitunter zu suboptimalen Lösungen führen. Bei Projekten, für die der Funktionsumfang im Vorfeld nicht klar abgegrenzt werden kann, kann sich die Entwicklung einer PWA als Sackgasse erweisen, nämlich dann, wenn sich im Verlauf der Entwicklung Abhängigkeiten auf Schnittstellen ergeben, die nur Native-Apps zur Verfügung stehen.

Wo sehen Sie die technischen Herausforderungen bei der Entwicklung von PWAs?

Eines der Probleme, das sich vermutlich nicht einfach lösen lässt, ist das der Webviews. Webviews sind in Native-Apps eingebundene Browser-Fenster, die zur Anzeige von

Webseiten genutzt werden. Webviews werden zum einen eingesetzt, um die Usability zu verbessern, und zum anderen, um zu verhindern, dass der Benutzer etwa beim Folgen eines Links die Anwendung verlässt. Der beschränkte Funktionsumfang dieser Webviews untergräbt das PWA-Konzept, weil dann nicht die notwendigen APIs zur Verfügung stehen.

PWAs wurden 2015 von Google angestossen. Welches Interesse verfolgt der Konzern mit dem neuen App-Format?

Die grossen Player haben sich mit ihren Stores für Native-Apps schon immer ein Wettrennen geliefert. Vermutlich verspricht sich Google hier einen Vorteil, indem es die Vorreiterrolle für ein App-Format einnimmt, das die Einstiegshürde zur Veröffentlichung dauerhaft senken könnte.

Apple hat sich hingegen lange gegen die Verbreitung von PWAs gestäubt. Warum?

Vermutlich aus demselben Grund. Google hat die Vorreiterrolle, also muss Apple auf die Position «Not invented here» gehen.

Können sich PWAs auch ohne die volle Unterstützung von Apple gegenüber Native-Apps durchsetzen?

Ich glaube nicht, dass es hier wirklich ums Durchsetzen geht, da ich es eher als ein Ergänzen von Technologien sehe. Aber natürlich würde ich mir wünschen, dass auch die grossen Player den Einsatz offener Standards unterstützen.