

**zühlke**  
empowering ideas



Whitepaper

# Wann ist eine Progressive Web App sinnvoll?

Mai 2022

# Wann ist eine Progressive Web App sinnvoll?

## Was ist eine PWA?

Eine PWA ist die Erweiterung einer klassischen Web App mit Funktionalitäten, die eigentlich für native mobile Applikationen vorbehalten sind. Eine PWA soll sich an-

fühlen wie eine native App, sie läuft aber eigentlich in einem Web-Browser.

## Welche Eigenschaften hat eine PWA?

<b>Progressiv:</b>	Funktioniert für jede Art von Browser
<b>Anpassungsfähig:</b>	Passt sich an jeder Art von Screen an: Desktop, Mobile, Tablet ...
<b>Netzwerk unabhängig:</b>	Kann offline benutzt werden
<b>App ähnlich:</b>	Verhält sich bei Interaktionen, Navigation, Benutzeroberfläche usw. wie eine mobile App
<b>Installierbar:</b>	Kann, wie eine klassische App, auf dem Startbildschirm installiert werden.

## Wie ist eine PWA aufgebaut?

Aus technologischer Sicht wird die PWA wie eine klassische Web-App gebaut. Wir sprechen hier also von den Standardtechnologien wie: Javascript, HTML und CSS als Basis und die darauf aufbauenden Frameworks wie z.B.: React JS, Angular JS, Vue JS. Zusätzlich werden PWA-spezifische Konfigurationen in einem sogenannten Manifest mitgegeben:

- App-Icon welches auf dem Startbildschirm erscheint
- Wie die App angezeigt wird, z.B.: in eigenem Fenster, mit oder ohne Browser UI, usw.
- Farbschema

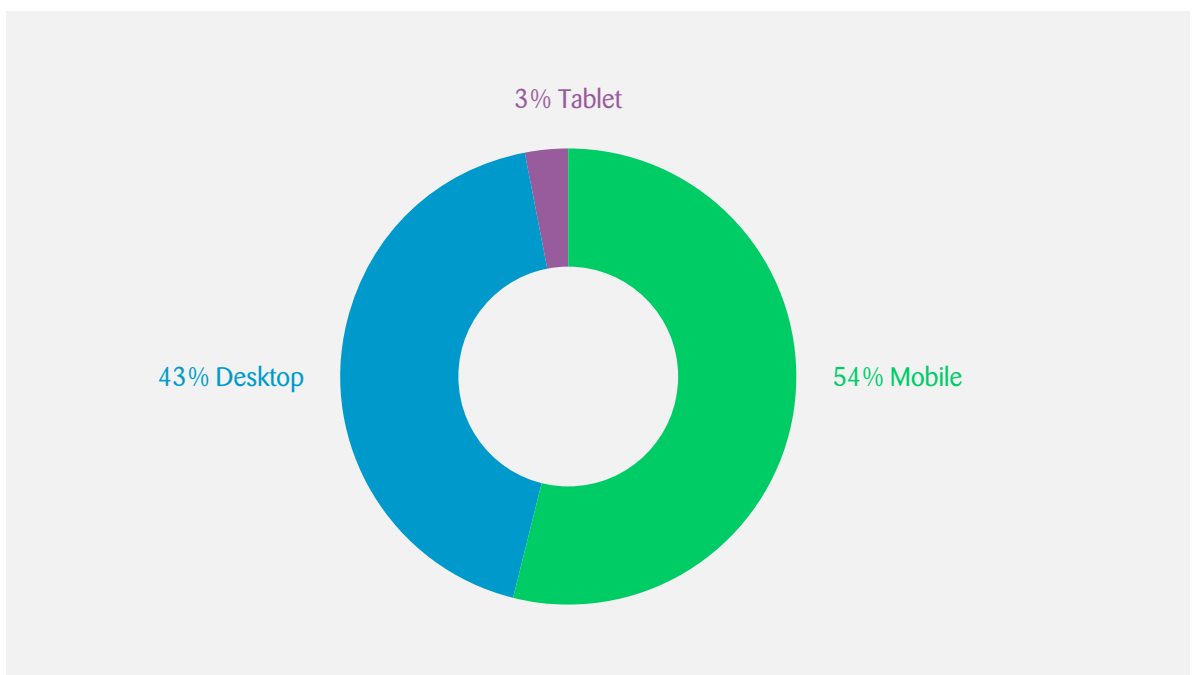
Folgende App-spezifischen Funktionalitäten können mittels PWA genutzt werden:

- Welche Daten, wie lange, offline verfügbar sind und wann diese aktualisiert werden
- Lokaler Datenspeicher kann genutzt werden
- Im Android App Store kann eine PWA veröffentlicht werden
- Gerätespezifische Funktionalitäten, wie Geolokalisation, Kontakte usw. (Siehe Kapitel «Geräte Funktionalitäten»)
- Anzeige und Kontrolle ausserhalb der App, z.B.: in einem Widget oder als Siri-Vorschlag

## Eleganter Mittelweg zwischen Web und Nativ

Webseiten werden mittlerweile häufiger von Smartphones aufgerufen als von anderen Geräten. Diese Tendenz ist steigend. Wenn man schon eine Webseite hat, kann man diese meistens mit wenigen Schritten offline-

verfügbar, allgemein Smartphone-freundlich und installierbar machen. Dies ermöglicht eine PWA mit wenig Aufwand und einem Responsive UI.



**Abbildung 1:** Marktanteil Mobile, Desktop und Tablet: <https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet>

Eine Native App wiederum würde einen viel grösseren Entwicklungsaufwand bedeuten und muss zur Veröffentlichung durch einen Genehmigungs-Prozess um dann in einen Store, wie Google Play oder Apple Store, zu gelangen. Je nach Funktionalität und Anforderungen der Applikation kann dieser Zeit- und Arbeitsaufwand mittels einer PWA umgangen werden.

Eine PWA ist ein guter Mittelweg zwischen Web und Nativ. Neben verschiedenen Vorteilen bringt eine PWA aber auch gewisse Einschränkungen und Limitierungen. Diese möchten wir in diesem Artikel aufzeigen und auch einen Leitfaden zur Verfügung stellen wann eine PWA in Erwägung gezogen werden kann und wann eher nicht.

# Vorteile & Features

## Entwicklung startet schneller

Die Entwicklung einer PWA unterscheidet sich minimal von der Entwicklung normaler Webapps. Für viele beliebte Frameworks, wie Angular, ReactJS oder Vuejs, gibt es bereits Vorlagen für eine PWA mit denen der Start der App sehr einfach wird. Diese braucht es jedoch nicht

einmal. Alles was eine PWA benötigt ist ein Manifest, ein Service Worker und die Homepage der App. Es wird keine komplizierte Pipeline mit MacOS benötigt und es müssen keine Accounts bei Apple und Google verwaltet werden.

## Günstiger für einfache Apps

Für einfachere Apps lohnt sich der Entwicklungsaufwand einer normalen App nicht, sei es mit Cross-Platform Technologien wie ReactNative oder Flutter. Eine PWA hingegen kann sehr schlank gehalten werden. Durch die

niedrigen Anforderungen kann bereits in kurzer Zeit eine schlanke App ausgerollt und Feedback dazu eingeholt werden.

## Sofort «Live»

Eine PWA auszurollen, funktioniert beinahe gleich, wie der Rollout einer normalen Website. Sobald die Artefakte der App ausgerollt sind, wird jeder Browser die neus-

te Version herunterladen und automatisch beim Benutzer installieren. Es ist kein Approval von Apple oder Google nötig.

## Website und App kombiniert

Die entwickelte PWA kann die meisten Vorteile auch in Webseiten-Form ausspielen. Beispielsweise können alle statischen Daten im Browser des Benutzers zwischengespeichert werden, sodass lediglich dynamische Daten geladen werden. Eine PWA kann also ebenfalls dieselbe Website auf dem Desktop darstellen. Mittels Reactive

Design kann derselbe Inhalt, anders arrangiert, auf einem breiten, grossen Bildschirm dargestellt werden. Ebenfalls ist es auch möglich, eine moderne Single Page Applikation in eine PWA umzubauen. Somit ist der Wartungsaufwand im Vergleich zu normalen Mobile Apps viel geringer.

## Zugriff auf Geräte Features

Eine PWA kann auf einige Mobile-spezifische Funktionalitäten zugreifen, prominentestes Beispiel ist der genaue Standort des Geräts. Ebenfalls ist es möglich Benachrichtigungen an das Gerät zu senden, mit Ausnahme von iPhones oder iPads. Ein weiteres Feature einer PWA ist, dass sie sich als Payment Handler registrieren kann. Somit kann ein Benutzer mit einer PWA bezahlen, ähnlich

wie bei ApplePay oder GooglePay. Sobald ein Händler die Bezahlmethode der PWA akzeptiert, können Transaktionen zwischen Benutzer und Händler stattfinden. Natürlich kann die PWA auch Hintergrundaktualisierungen ausführen, beispielsweise um Daten des Benutzers auf dem Gerät zu speichern, sodass der Benutzer die PWA komplett offline benutzen kann.

# Nachteile & Hindernisse

## Integration für Geräte und Browser

Mittlerweile gibt es eine grosse Menge an verschiedenen Browsern, die auf unzähligen verschiedenen Geräten laufen. Natürlich hat nicht jedes Gerät die gleichen Funktionalitäten. Auf einem iOS Gerät zum Beispiel funktionieren Push Notifikationen aus einer simplen PWA nicht. Auch verschiedene Browser Versionen können limitierend sein. Z.B der Payment Handler funktioniert noch

nicht auf allen Versionen von Firefox. Die Limitierungen der verschiedenen Browser Versionen und Geräte müssen bei der Entwicklung beachtet werden. Eine native App ist von einem App-Store kontrolliert und kann nur für gewünschte Geräte installiert werden. Eine PWA hingegen kann von jedem Web-Browser geöffnet werden, der auf irgendeinem beliebigen Gerät läuft.

## Installieren einer PWA

Apps werden normalerweise in einem der bekannten App-Stores, (Google Play oder iOS App Store), installiert. Bei einer PWA erfolgt die Installation normalerweise nicht über einen App-Store. Die Installation erfolgt über einen Menu-Knopf im Browser, welcher die Option zeigt: «Zum Startbildschirm hinzufügen». Viele Nutzer wissen nicht, dass diese Möglichkeit besteht.

Um die Nutzer besser auf die Installationsmöglichkeit hinzuweisen, kann ein eigener Installation-User-Flow eingebaut werden. Diese Möglichkeit gibt es nur auf Android.

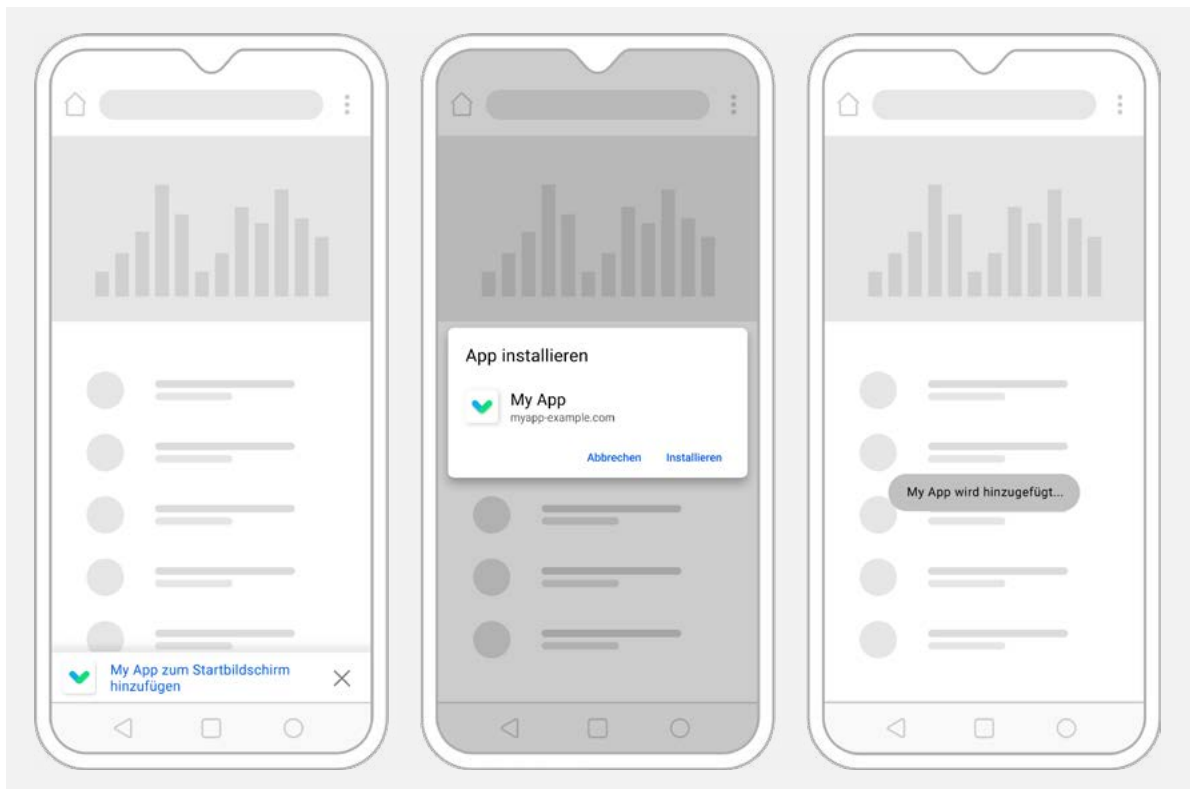


Abbildung 2: Android mit direktem Link zur Installation

Auf iOS kann kein Link erstellt werden, der die Installation direkt aufruft. Die Installation kann ausschließlich über das Browser-Menu getätigt werden. Man kann dem

iOS-Nutzer mit einer grafischen Anleitung erklären wie die App einfach auf den Hauptbildschirm installiert wird.

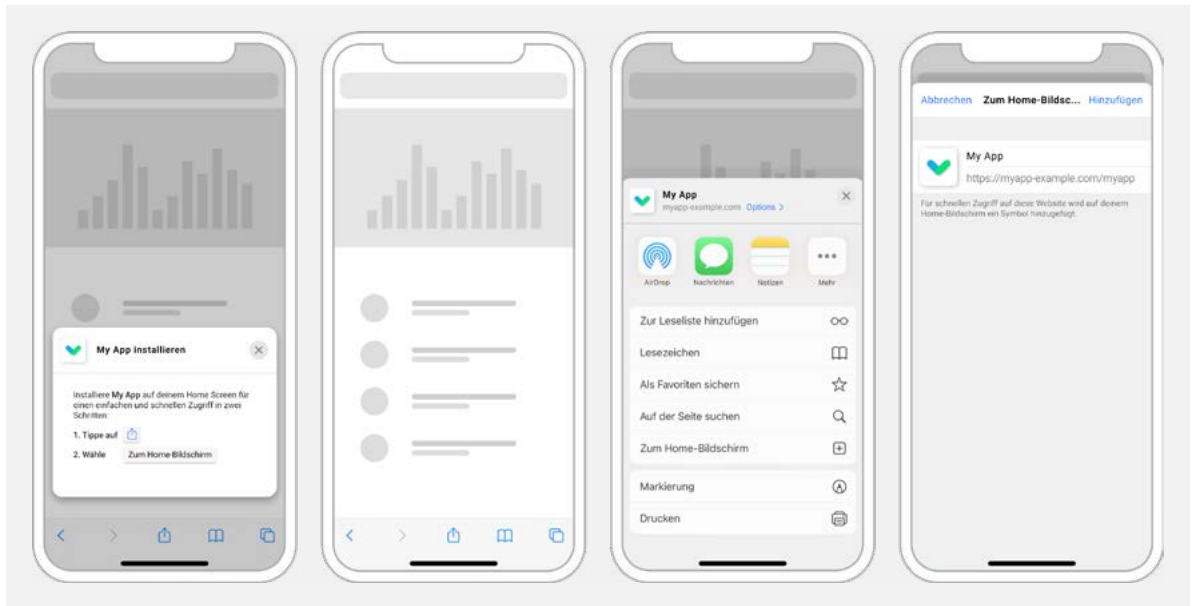


Abbildung 3: iOS mit Installationshinweis und den zusätzlichen Schritten

Falls gewünscht kann die PWA auch im Google Play veröffentlicht werden. Für iOS gibt es diese Möglichkeit nicht.

## Geräte Funktionalitäten

Eine PWA ist stark abhängig vom Browser und dessen Kapazitäten. Gewisse Geräte-internen Funktionalitäten sind in einem Browser und deshalb auch in einer PWA nicht verfügbar, wie z.B: Gesichts-/Fingerabdruck ID,

Altimeter Sensor. Auf iOS sind ausserdem noch Push Notifikationen, Bluetooth und Zugriff auf Batterie Informationen nicht verfügbar.

	iOS	Android	Desktop
Kamera Aufnahme	✓	✓	✓
Geolokalisierung	✓	✓	✓
Push-Notifikationen	×	✓	✓
Single Sign On	✓	✓	✓
Zugriff auf Kontakte	×	✓	×
Bluetooth	×	✓	✓
Bezahlung / Payment	✓	✓	✓
Sensorik	✓	✓	×
Netzwerk Info	×	✓	✓
Widgets	×	×	—
App Clips, Instant Apps	×	×	—
Smart Watch	×	×	—

## Testing

Grundsätzlich kann fürs Testen ähnlich vorgegangen werden wie bei einer Browser Applikation, die mit den Standard-Webtechnologien entwickelt wurde. Bei einer PWA muss zusätzlich noch Wert auf die verschiedenen Browser Versionen auf den Mobilien Geräten gelegt werden. Auch «reactive design»-Tests für Geräte mit ver-

schiedenen Bildschirmgrößen dürfen nicht unterschätzt werden.

Gewisse Browser wie z.B. Google Chrome bieten sehr gute Werkzeuge um PWA-spezifische Technologien, z.B. Caching von Daten, zu analysieren und testen.

# Migration zu einer PWA

## (Existierender Web Plattform zu PWA)

### Kurze Ladezeit

Eine fundamentale Eigenschaft jeder PWA ist, dass sie schnell lädt. Die Qualität der Internetverbindung sollte beim Laden der WebApp eine möglichst kleine Rolle spielen. Es können natürlich Daten von einem Back-End geladen werden. Business-Logik oder das Framework

des Front-Ends können im Browser Cache installiert werden. Schlussendlich sollten alle statischen Daten wie Bilder, Schriftarten und auch Front-End Code im Browser gecached werden und nur beim Deployen einer neuen Version auf das Gerät heruntergeladen werden.

### Responsivität

Responsivität ist ein weiteres Merkmal einer guten WebApp. Eine PWA soll sich «App-like» anfühlen, und muss somit auf jeder Plattform, sei es Smartphone, Tablet oder Desktop ein gute User-Experience bieten. Die WebApp soll schnell auf Eingaben des Benutzers reagieren und

Rückmeldung geben, was gerade passiert. Eine Responsive WebApp fördert erwiesenermassen das Engagement der Benutzer der WebApp und somit auch die Zufriedenheit mit dem Produkt.

### Offline-Fähig

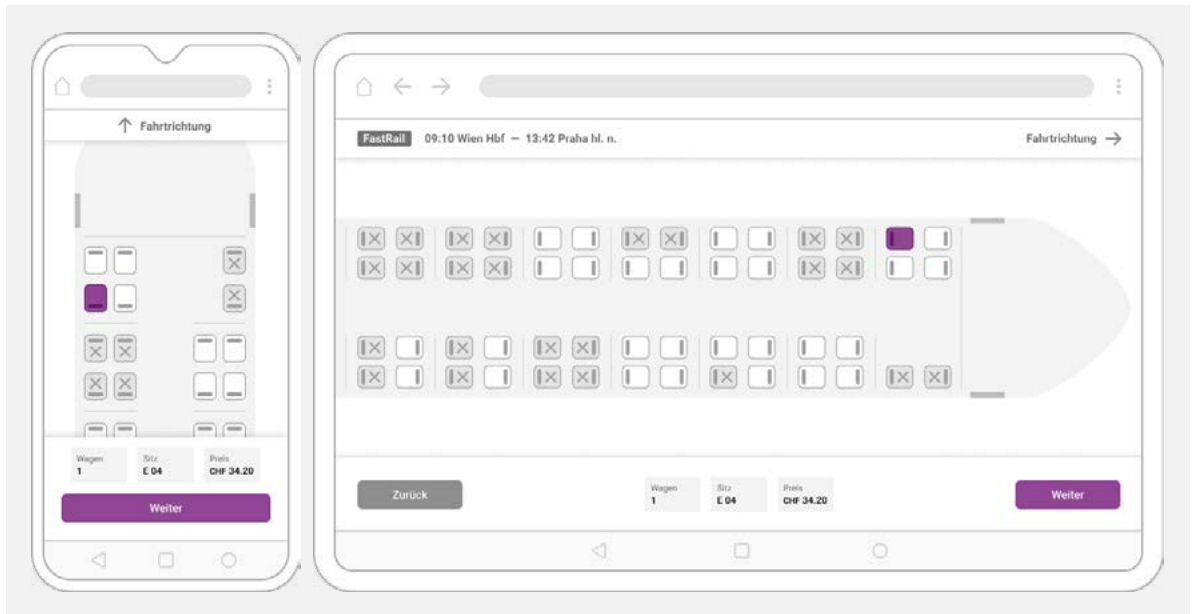
Zudem soll eine PWA unabhängig von der Qualität der Internetverbindung funktionieren. Wie schon eingangs erwähnt bietet eine PWA gute Caching Mechanismen an. Zudem kann es bei einigen Anwendungsfällen Sinn machen, nicht nur statische Daten zu cachen. Immer mehr Nutzer verwenden mehrheitlich oder sogar ausschliesslich mobile Endgeräte. Schlechte oder gar keine Internetverbindung sind somit ebenfalls Anwendungsfälle, welche die PWA abdecken muss. Für eine WebApp, welche beispielsweise Flugtickets anzeigt, ist es beson-

ders wichtig, dass das Ticket auch im Offline-Modus angezeigt werden kann.

Für jedes moderne Javascript Framework gibt es Erweiterungen, die helfen die Web App in eine PWA umzubauen. Diese helfen, die erwähnten Punkte umzusetzen um beispielsweise statische Daten zu cachen, während dynamische Daten (Benutzerspezifische Daten) vom Back-End geladen werden.

# Erfahrungen mit PWA

## Platzreservierungs App – Handy, Tablet und Browser App in einem



Im Rahmen einer Marktforschung sollte eine Platzreservierungs-App entwickelt werden, welche es erlaubt Passagieren von Zügen schon im Vorhinein einen Platz mit Zusatzdienstleistungen zu reservieren. Für dieses Projekt entschieden wir uns für eine PWA.

Die App sollte schnell entwickelt werden mit wenig Entwicklern. Eine PWA benutzt hauptsächlich Web Frontend Technologien und es sind keine Nativ Android oder iOS Kenntnisse nötig. Die Anforderungen an die App waren unkompliziert und nur wenige Geräte spezifischen Funk-

tionen waren nötig. Die Applikation sollte auch via Browser auf Desktop zugänglich sein. Aus diesen Gründen passte eine PWA perfekt zu den gegebenen Anforderungen.

Mit einem Mobile-First Ansatz ist es wichtig, dass die App auch bei schlechter Netzwerkverbindung möglichst schnell geladen werden kann. Wir entschieden uns für das JavaScript Framework VueJS, welches sehr klein gehalten werden kann. Die PWA erfüllte alle Anforderungen und konnte schnell entwickelt werden.

## Termin-Buchungs App

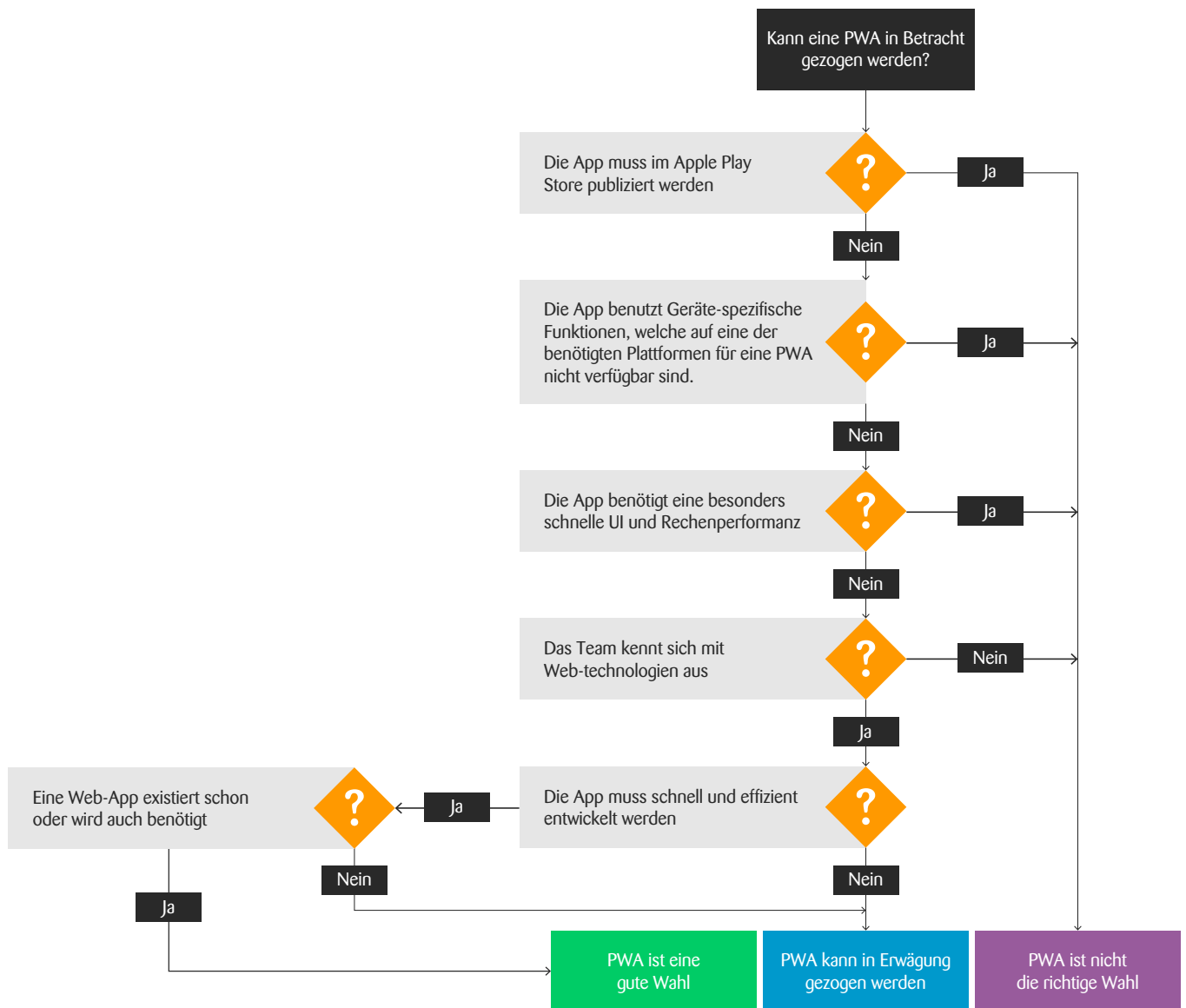
Um das Potential einer App zu eruieren, in welcher Kunden Termine buchen und verwalten können, wurde ein Prototyp für eine Versicherung gebaut. Dieser Prototyp wurde innerhalb der Firma des Kunden verwendet, um Usability Tests und Demonstrationen durchzuführen. Um schnell und ohne viel Projektsetup starten zu können, wurde eine PWA verwendet. Obwohl das Entwicklerteam bereits Erfahrung in Entwicklung von Apps hatte, war eine PWA die günstigere und vor allem einfachere Alternative.

Kern der Anwendung war, schnell einen Termin bei einem Dienstleister in der Nähe des Benutzers buchen zu können. Mit einer PWA war es einfach möglich, den Gerätestandort abzufragen und mögliche Termine anzuzeigen. Bei der Entwicklung wurde ein Mobile-First Ansatz verwendet, somit war die Anwendung auch Desktop fähig. Zudem war es wichtig, dass die Termine in der App ersichtlich waren, auch wenn das Gerät schlechte oder gar keine Internetverbindung hat.



# PWA Leitfaden

Als Hilfe zur Entscheidungsfindung, ob eine PWA in Erwägung gezogen werden sollte, kann folgendes Diagramm helfen:



# Kontakt



**Doris Rogger**  
Head of Mobile  
doris.rogger@zuehlke.com  
+41 31 561 39 35

