Technische Hochschule
Brandenburg
University of
Applied Sciences

Fachbereich Informatik und Medien

MASTERARBEIT

Erstellung und Evaluation eines Leitfadens zur Nutzung von Progressive Web Apps

Vorgelegt von: Kevin Frank

am: 03.12.2018

zum

Erlangen des akademischen Grades

Master of Science

(M.Sc.)

Erstgutachter: Prof. Dr.-Ing. Thomas Preuss

Zweitgutachterin: Dr. Marie Gebhardt

Danksagung

„Der schlimmste aller Fehler ist, sich keines solchen bewußt zu sein.“[1]

Mit diesen Worten von Thomas Carlyle wird die Fehlbarkeit eines einzelnen Menschen verdeutlicht. Großes kann nicht als Individuum, sondern nur im Kollektiv erreicht werden. In diesem Sinne habe auch ich bei der Erstellung dieser Masterarbeit Unterstützung erhalten, der ein entsprechender Dank gebührt.

Zunächst bin ich Prof. Dr.-Ing. Thomas Preuss für die Betreuung meiner Arbeit dankbar. Meine Anfragen wurden immer kurzfristig und mit einem konstruktiven Feedback beantwortet. Weiterer Dank gilt Dr. Marie Gebhardt, Constanze Weber und Chris Braun, die meine Arbeit auf Herz in Nieren geprüft und somit zum Feinschliff beigetragen haben. Ebenfalls bin ich Achim Gosse, Kevin Taron und Peter Kröner dafür dankbar, dass sie das von mir im Rahmen der Arbeit erstellte Regelwerk geprüft und mit mir in einem Interview besprochen haben. Prof. Dr. Jörg Thomaschewski gilt nicht nur dafür Dank, dass er mir den Kontakt zu einigen Experten ermöglicht hat, sondern auch dafür, dass er mich noch vor dem Beginn der eigentlichen Arbeit bei der Themenwahl unterstützt und somit auf den richtigen Pfad geleitet hat. Einen großen Beitrag haben auch die Teilnehmer meiner Online-Umfrage geleistet, die auf Grund der Menge und der Anonymität der Umfrage an dieser Stelle nicht namentlich genannt werden können. Ich hoffe, dass mein Dank ihnen auf diesem Wege erreicht. Zuletzt will ich mich bei meinen Eltern bedanken, die mir während des gesamten Studiums den Rücken freigehalten und mich in vielerlei Hinsicht unterstützt haben.

[1] Carlyle, Thomas (1893), S. 5.

Abstract

Ein neuer Trend in der Webentwicklung besteht aus den von Google beworbenen Progressive Web Apps (PWA). Diese tragen das implizite Versprechen in sich, eine Lösung für alle Fälle zu sein. Weil solche „One Fits All“ Lösungen selten ihr Versprechen halten, untersucht diese Masterarbeit PWAs im Detail. Die Ziele waren dabei, den PWA-Begriff zu definieren, die Grenzen von PWAs zu untersuchen und diese in einem leicht zugänglichen Leitfaden zur Nutzung von PWAs festzuhalten. Der Leitfaden soll Entscheidungsträgern bei der Beantwortung der Frage unterstützen, ob sie in einer bestimmten Situation eine PWA entwickeln sollten oder nicht. Bei einer Recherche hat sich herausgestellt, dass PWAs lediglich Webanwendungen sind, die moderne Browsertechnologien nutzen. Entsprechend besitzen PWAs bis auf einige Ausnahmen dieselben Grenzen wie herkömmliche Web Apps. Mit diesem Vorwissen wurde dann sowohl eine Definition in deutscher Sprache als auch eine internationale Version in englischer Sprache verfasst. Aus einem detaillierten Vergleich von nativen und Web Apps wurden anschließend die Funktionen und Grenzen von PWAs ermittelt. Diese wurde in einem schriftlichen Regelwerk festgehalten, das von drei Experten in einem Peer Review geprüft und anschließend in einem Interview besprochen und korrigiert wurde. Das Ergebnis ist zwar nicht sehr repräsentativ, sowohl die Korrektheit und Vollständigkeit als auch das allgemeine Feedback der Experten ist jedoch überwiegend positiv ausgefallen. Um den leichten Zugang und eine bessere Nutzbarkeit zu gewährleisten, wurde der schriftliche Leitfaden als PWA umgesetzt. Mit Hilfe einer Online-Umfrage wurde zunächst die Konformität dieser Leitfaden-App zum schriftlichen Regelwerk untersucht, wobei eine Evaluation von Usability und User Experience für einen späteren Zeitpunkt außerhalb der Masterarbeit angedacht wurden. Entsprechend konnte der Nutzen der Leitfaden-App noch nicht nachgewiesen werden. Ebenfalls haben sich die Wahl und Umsetzung der Evaluationsmethodik als suboptimal herausgestellt, wodurch die Konformität der App nicht bestätigt, sondern lediglich Hypothesen aufgestellt werden konnten. Offen geblieben sind die Erhöhung der Repräsentanz der Ergebnisse aus der Evaluation des Regelwerks, die Untersuchung der Konformität der Leitfaden-App über eine verbesserte Methodik und die Prüfung und Optimierung selbiger in Bezug auf die Usability.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

API Application Programming Interface

DOM Document Object Model

GPS Global Positioning System

JSON JavaScript Object Notation

NFC Near Field Communication

PWA Progressive Web App

RFC Request For Comment

SIM Subscriber Identity Module

SPA Single Page Application

UX User Experience

W3C World Wide Web Consortium

1. Einleitung

Schon seit mehreren Jahrzehnten stehen Softwareentwickler vor der Herausforderung der plattformunabhängigen Entwicklung von Anwendungen. Eines der ersten Probleme ergab sich aus der Vielfalt der zu unterstützenden Betriebssysteme, woraus sich einige der aktuell am weitest verbreiteten Programmiersprachen - wie z.B. Python oder Java[2] - ergeben haben. Eine aktuelle Herausforderung erwächst aus der Vielzahl an Endgeräten, die jeweils unterschiedliche Eigenschaften und Fähigkeiten besitzen. Da auf so gut wie jedem Gerät ein Browser installiert ist, werden häufig plattformübergreifende Webanwendungen entwickelt. Diese haben jedoch häufig das Problem, dass aus Gründen der Sicherheit viele Funktionen nicht zur Verfügung stehen, weswegen eine Entwicklung von plattformabhängigen, nativen Apps notwendig wird. Ein neuer Trend namens Progressive Web Apps (PWA) will nun die Grenze zwischen nativen und Web Apps sprengen und somit dieses Dilemma auflösen.

Bei PWAs handelt es sich um ein von Google vorangetriebenes Konzept, nach welchem sich Webanwendungen in ihren Eigenschaften immer weiter an native Apps annähern. In der Theorie muss also nur noch eine PWA für alle Endgeräte entwickelt werden, in der Praxis halten solche „One Fits All“ Lösungen jedoch selten das, was sie versprechen.

Ziel dieser Masterarbeit soll es deshalb sein, den PWA-Begriff zu definieren und deren technische Grenzen zu untersuchen. Werden solche Grenzen gefunden, sollen die gewonnenen Erkenntnisse zum Einsatz von PWAs in einem leicht zugänglichen Leitfaden bereitgestellt werden. Dieser soll Entscheidungsträgern bei der Beantwortung der Frage unterstützen, ob sie in einer bestimmen Situation eine PWA entwickeln sollten oder nicht.

Dazu soll in einer umfassenden Recherche zunächst der PWA-Begriff entmystifiziert und eine Grundlage zum Verständnis von PWAs etabliert werden. In einem weiteren Kapitel werden die wissenschaftlichen Methoden vorgestellt, die für den praktischen Teil der Arbeit relevant sind. In einer weiteren Recherche werden dann die Fähigkeiten von nativen und Web Apps ermittelt und gegenseitig verglichen, um somit die Fähigkeiten von PWAs abzuleiten. Diese werden dann in einem schriftlichen Regelwerk festgehalten und mit einer Kombination von Peer Reviews und Experteninterviews validiert und korrigiert. Für einen leichten Zugang wird das Regelwerk anschließend in einer Leitfaden-App umgesetzt, die wiederum anhand einer Online-Umfrage mit User Stories evaluiert wird.

[2] Vgl. Stack Overflow Talent (2017), S. 34.

2. Progressive Web Apps in der Theorie

Bevor der Leitfaden zur Nutzung von PWAs erstellt werden kann, sollte zunächst ein grundlegendes Verständnis darüber vorhanden sein. Weil auf Grund der Aktualität des Themas anscheinend noch keine etablierte Definition existiert, soll in diesem Kapitel eine Definition zu PWAs für den Rahmen der vorliegenden Masterarbeit gefunden werden. Dazu werden zunächst die verschiedenen Typen von Apps erläutert.

2.1. Typen von Apps

2.1.1. Native Apps

Als erstes sind hierbei die nativen Apps zu nennen. Bei diesen handelt es sich um Anwendungen, die für eine bestimmte Plattform bzw. ein bestimmtes Betriebssystem entwickelt wurden und direkt auf dem Endgerät installiert und ausgeführt werden.[3] Im Rahmen eines mobilen Nutzungskontexts bzw. bei mobilen Endgeräten werden die Anwendungen auch „Native Mobile Apps“ genannt. Da im aktuellen Trend die Grenzen zwischen mobilen und nicht-mobilen Anwendungen zunehmend verschwimmen und der Großteil der folgenden Betrachtungen auf alle Arten von Apps angewandt werden kann, wird im Weiteren auf die „Mobile“ Ergänzung verzichtet.

Die folgende Abbildung 1 visualisiert die Architektur einer nativen App[4]:

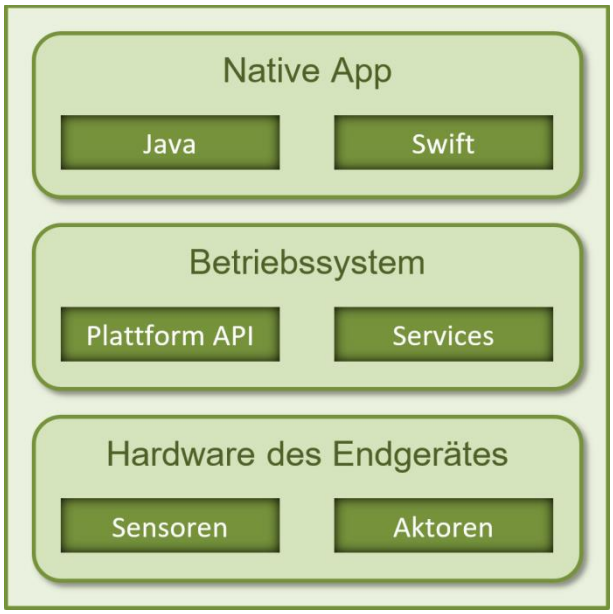

Abbildung 1: Architektur von Nativen Apps
Quelle: eigene Darstellung

[3] Vgl. Steyer, Ralph (2017), S. 30 f.
[4] Die Abbildung enthält nicht alle Elemente der Architektur nativer Apps, sondern nur solche, die für die Unterscheidung der verschiedenen App-Typen relevant sind.

Auf der untersten Ebene befindet sich die Hardware des Gerätes mit seinen Sensoren und Aktoren. Bei einem Smartphone zählen dazu z.B. die Kamera oder das Global Positioning System (GPS). Die Hardware wird durch ein Betriebssystem (auch Plattform genannt) gesteuert, welches wiederum ein Application Programming Interface (API) und verschiedene Services zu Umsetzung der nativen App anbietet. Da Android und iOS über 95% des weltweiten Marktanteils der mobilen Plattformen ausmachen[5], beschränkt sich die Arbeit im Folgenden auf diese beiden Betriebssysteme. Android ist eine von Google entwickelte Open Source Software, deren native Apps mit Hilfe der Entwicklungsumgebung Android Studio in der Programmiersprache Java entwickelt werden. Durch das Open Source Modell gibt es mit z.B. Samsung, HTC, LG usw. viele Hersteller, die Android-Geräte produzieren. iOS hingegen ist eine von Apple entwickelte Closed Source Plattform. iOS-Apps können zwar auch von anderen Herstellern mit der Entwicklungsumgebung Xcode in den Sprachen Objective-C oder Swift erstellt werden, Apple behält sich jedoch vor, der einzige Hersteller für iOS-Geräte zu sein.[6,7]

Da eine native App direkt auf dem Betriebssystem aufbaut, kann sie alle Services und Gerätefunktionalitäten nutzen, profitiert von einer hohen Performance und kann optimal auf die spezielle Plattform angepasst werden. Die Vermarktung nativer Apps findet über einen sogenannten App-Store (z.B. Google Play bei Android oder Apple App Store bei iOS) statt. Apps müssen einen vom App-Store definierten Qualitätssicherungsprozess durchlaufen, wodurch zwar ein gewisses Niveau an Qualität bei jeder App sichergestellt ist, jedoch gleichzeitig eine schnelle Veröffentlichung bzw. Aktualisierung verhindert wird. Ein weiterer Nachteil nativer Apps ist, dass deren Quellcode nicht auf andere Plattformen übertragen werden kann, wodurch für jedes Betriebssystem eine neue App entwickelt werden muss, was wiederum hohe Entwicklungs- und Wartungskosten mit sich bringt.[8]

2.1.2. Web Apps

Eben diesen Nachteil haben Web Apps nicht. Bei ihnen handelt es sich um Anwendungen, die mit standardisierten Webtechnologien wie HTML, CSS und JavaScript entwickelt werden. Diese HTML-, CSS- und JavaScript-Dateien sind Webressourcen, die auf einem Webserver im Internet gehostet werden. Mit Hilfe eines Webbrowsers kann über eine eindeutige URL auf die Webressourcen zugegriffen werden. Sie werden mittels HTTP übertragen.[9]

---

[5] Vgl. StatCounter (2018b), Abschnitt „Mobile Operating System Market Share Worldwide“.
[6] Vgl. Knott, Daniel (2016), S. 17.
[7] Vgl. Vollmer, Guy (2017), S. 16 f.
[8] Vgl. Malavolta, Ivano (2016), S. 1.
[9] Vgl. ebenda, S. 1 f.

Die Architektur von Web Apps wird durch Abbildung 2 beschrieben:

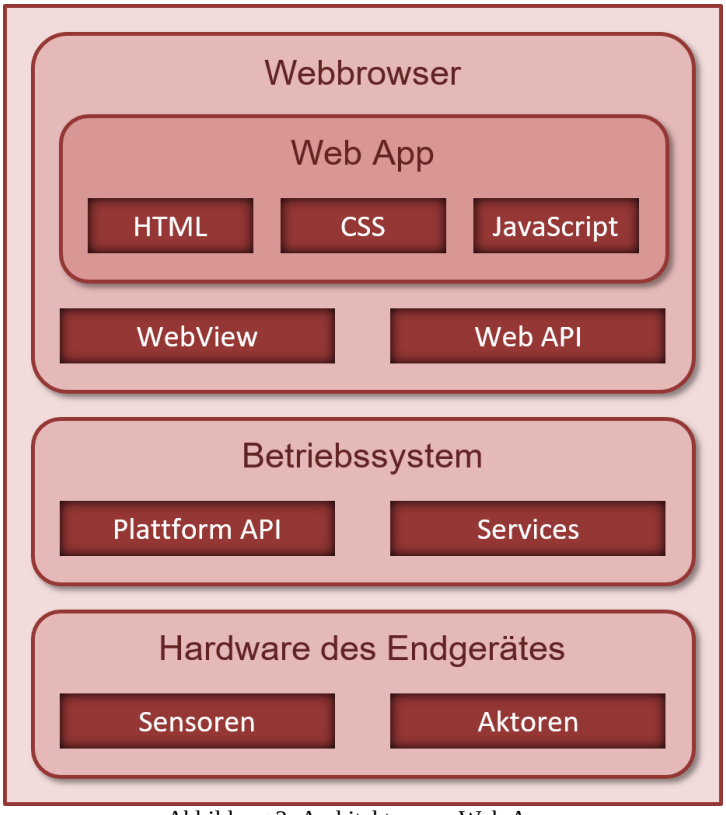

Abbildung 2: Architektur von Web Apps
Quelle: eigene Darstellung

Die untersten zwei Schichten (Hardware und Betriebssystem) unterscheiden sich nicht von der Architektur der nativen Apps, da diese durch das Endgerät und nicht durch den Typ der App vorgegeben werden. Anstatt so, wie bei nativen Apps, die Web App direkt auf dem Betriebssystem auszuführen, wird sie in einem Webbrowser ausgeführt, der auf dem Betriebssystem läuft. Dieser enthält mit der WebView eine Rendering-Engine zur Darstellung von HTML und CSS und eine Web API, auf die mittels JavaScript zugegriffen werden kann. Prinzipiell sind Browser auf jeder Plattform verfügbar. Für mobile Plattformen können beispielhaft die Browser Chrome für Android oder Safari für iOS genannt werden. Da also immer mindestens ein Webbrowser vorhanden ist und Web Apps mit standardisierten Webtechnologien entwickelt werden, sind diese auf jeder Plattform lauffähig. Für mehrere Plattformen kann eine Code-Basis wiederverwendet werden, was geringere Entwicklungs- und Wartungskosten nach sich zieht. Hinzu kommt, dass Web Apps mittels Suchmaschinen im Internet gefunden werden können. Eine Verteilung über App Stores ist nicht nötig, wodurch die langwierigen Genehmigungsprozesse entfallen. Somit können Web Apps sofort aktualisiert werden, um z.B. schnell Fehler zu beheben. Nachteile sind jedoch, dass Web Apps kaum bis gar nicht offline nutzbar sind, da auf die Webressourcen über das Internet zugegriffen wird. Diese Tatsache und der Browser als Zwischenschicht sorgen zusätzlich für eine schlechtere Perfor-

mance. Zuletzt beschränkt der Browser mit seiner Web API u.a. aus Sicherheitsgründen die Web App, wodurch diese nicht auf alle Funktionen und Services des Endgerätes zugreifen kann.[10,11]

Eine Web App ist jedoch nicht unbedingt gleich einer anderen Web App, denn auch hier können noch verschiedene Typen unterschieden werden. So besteht eine herkömmliche Webseite aus mehreren HTML-Dokumenten, die untereinander verknüpft sind. Der primäre Fokus liegt auf der Bereitstellung von Informationen, wodurch es sich eher um ein passives Angebot handelt. Eine spezielle Form der Webseite ist die sogenannte Single Page Application (SPA). Diese besteht nur aus einer HTML-Seite, deren dynamischen Inhalte bei Bedarf nachgeladen werden. Zuletzt gibt es noch die Rich Internet Applications. Als sehr interaktive Webseiten bieten sie ein aktives Angebot an und ähneln vom Verhalten her am meisten den Desktop-Anwendungen bzw. nativen Apps.[12]

2.1.3. Hybride Apps

Ein weiteres App-Modell ist das der hybriden Apps. Ihre genaue Funktionsweise hängt vom verwendeten Framework ab, alle versuchen jedoch dasselbe zu erreichen: Die Stärken von nativen und Web Apps sollen kombiniert und gleichzeitig ihre Nachteile beseitigt werden. Frameworks, die zum aktuellen Zeitpunkt eine hohe Präsenz am Markt haben, sind PhoneGap, Cordova, Ionic und React Native.[13] Um einen Überblick über hybride Apps zu geben, sollten an dieser Stelle nur PhoneGap und Cordova vorgestellt werden.

PhoneGap war ursprünglich ein Open Source Framework der Firma Nitobi. Im Oktober 2011 wurde PhoneGap von Adobe Systems übernommen, wobei der Quellcode des Frameworks der Apache Software Foundations für das Open Source Projekt Apache Cordova bereitgestellt wurde. Aktuell ist *Cordova* eine Engine, wohingegen PhoneGap eine Distribution von Cordova ist. Da dies Unterschied für das Verständnis hybrider Apps nicht relevant ist, werden die Begriffe „Cordova“ und „PhoneGap“ im Folgenden synonym verwendet.[14] Die Architektur einer Cordova App wird in Abbildung 3 unten visualisiert. Wie man der Abbildung entnehmen kann, werden Cordova Apps mit den im Abschnitt 2.1.2 Web Apps beschriebenen Webtechnologien entwickelt. Anstatt die Webressourcen jedoch in einem Browser auszuführen, werden sie in einem nativen Wrapper gebündelt. Der Wrapper enthält mit einer WebView

[10] Vgl. Knott, Daniel (2016), S. 21 ff.
[11] Vgl. Vollmer, Guy (2017), S. 17 f.
[12] Vgl. Steyer, Ralph (2017), S. 28 ff.
[13] Vgl. G2 Crowd (o.J.), Abschnitt „G2 Crows Grid for Mobile Development Frameworks“.
[14] Vgl. Steyer, Ralph (2017), S. 5 f.

wieder eine Rendering Engine für HTML und CSS, wodurch er ähnlich wie ein Browser arbeitet. Anders als ein Webbrowser bietet der native Wrapper jedoch keine Web API, sondern ein mittels verschiedenen Plugins erstelltes Foreign Function Interfaces an. Dabei handelt es sich um einen Mechanismus, mit dem Routinen oder Services durch Quellcode in einer anderen Programmiersprache aufgerufen werden können. Dadurch ist es dann möglich, innerhalb des JavaScript-Codes auf die geschützten Funktionen und Services des Gerätes zuzugreifen.[15,16]

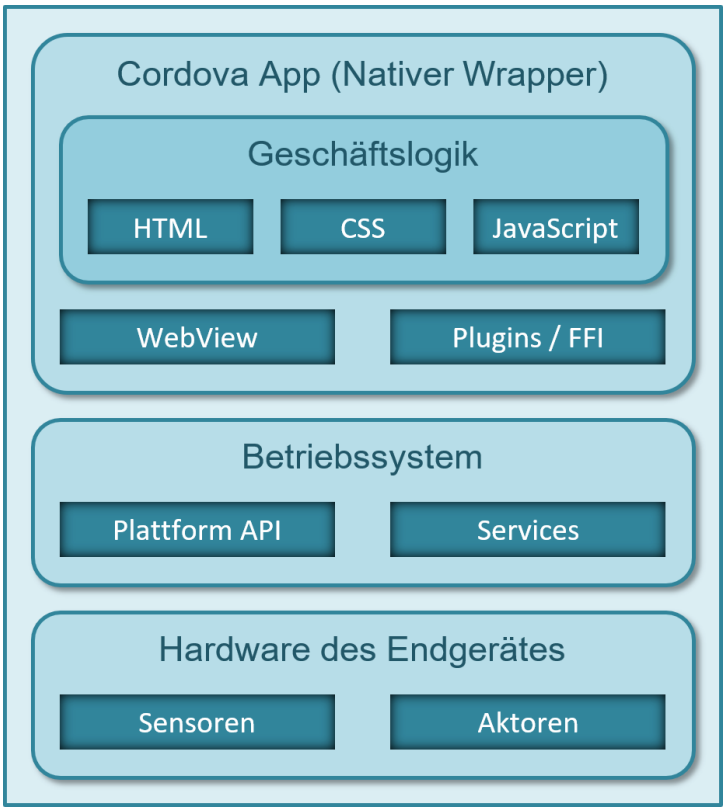

Abbildung 3: Architektur von Cordova Apps
Quelle: eigene Darstellung in Anlehnung an Apache Cordova (o.J.), Abschnitt „Architecture“ und Que, Peixin u.a. (2016), S. 611

Dadurch, dass Cordova Apps einmalig mittels HTML, CSS und JavaScript entwickelt und anschließend für jede unterstützte Plattform „gewrappt“ werden können, sind sie quasi plattformunabhängig und bringen geringere Entwicklungs- und Wartungskosten als native Apps mit sich. Durch den Wrapper lassen sich Cordova Apps ähnlich wie native Apps installieren und über einen App Store veröffentlichen, was wiederum den Vorteil der höheren Qualität durch den Genehmigungsprozess der App Stores, aber auch den Nachteil der langsameren Aktualisierung mit sich bringt. Ein weiterer Nachteil ist, dass die Performance solcher Apps durch den nativen Wrapper als zusätzliche Schicht schlechter als die der nativen Apps ist.[17]

[15] Vgl. Gasston, Peter (2014), S. 222.
[16] Vgl. Que, Peixin u.a. (2016), S. 6 ff.
[17] Vgl. Vollmer, Guy (2017), S. 19.

2.2. Progressive Enhancement und Graceful Degradation

Im Begriff der "Progressive Web App" sind die im Kapitel 2.1.2 vorgestellten Web Apps enthalten, jedoch auch das Wort „Progressive", das durch das „Progressive Enhancement" geprägt wurde. Um letzteres jedoch verstehen zu können, muss zunächst die *Graceful Degradation* (dt. „würdevolle Herabstufung") erläutert werden. Bei ihr handelt es sich um ein Paradigma zur Webentwicklung. Demnach werden Webseiten zunächst für aktuelle, gängige Browser entwickelt. Nach der Fertigstellung der Webseiten werden diese dann durch ein nachträglich implementiertes Workaround so angepasst, dass sie zumindest irgendwie in älteren Browsern funktionieren. Ausgangspunkt für die Graceful Degradation war, dass der Nutzer ja nur den alten Browser aktualisieren muss, der ja ohnehin kostenfrei nutzbar ist. Es gibt jedoch einige Gründe, die gegen ein Browserupdate sprechen und bei diesem Gedankengang nicht beachtet wurden:

- Richtlinien in einem Unternehmen verbieten die Aktualisierung des Browsers.
- Die Hardware des Endgerätes ist zu alt für ein Software-Update.
- Kostspielige Hilfsmittel zur Gewährleistung von Barrierefreiheit (z.B. Screenreader) sind nur mit älteren Versionen des Browsers funktionsfähig.[18]

Aus diesen und weiteren Gründen hat Steven Champeon im Jahre 2003 mit dem *Progressive Enhancement* (dt. „fortschreitende Verbesserung") eine Modernisierung des früheren Prinzips der Graceful Degradation vorgestellt. Demnach sollen Webseiten in mehreren Stufen entwickelt werden. Auf der untersten Stufe wird eine Seite so eingerichtet, dass deren Inhalt in der einfachsten Form von jedem Browser aus zugänglich ist. Unterstützen die Browser weitere Funktionalitäten, so werden diese auf weiteren Stufen bereitgestellt, ohne jedoch die Funktionsfähigkeit bei älteren Browsern zu beeinflussen. Auf diese Weise erfolgt eine fortschreitende Verbesserung der Webseite. Im Vergleich zur Graceful Degradation wirkt das Progressive Enhancement also nicht nachträglich, sondern vor und während der Entwicklung. Erstellen Entwickler Webanwendungen nach dem Progressive Enhancement, müssen diese gleichzeitig akzeptieren, dass ihre Anwendung in mehreren Browsern unterschiedlich aussehen kann und eine unterschiedliche User Experience (UX) bietet.[19]

Zur Einhaltung des Vorgehens des Progressive Enhancements sollten einige Aspekte beachtet werden:[20] Neben der Einhaltung der üblichen Trennung von Inhalt, Gestaltung und Logik in HTML, CSS und JavaScript sollte eine semantische HTML-Auszeichnung genutzt werden.

---

[18] Vgl. Hellbusch, Jan E. & Probiesch, Kertin (2011), S. 197 f.
[19] Vgl. Castro, Elizabeth & Hyslop, Bruce (2014), S. 6 f.
[20] Vgl. SELFHTML (2018b), Abschnitt „Progressive enhancement".

Semantik ist die Bedeutung von Zeichen, Wörtern und Symbolen. Verschiedene HTML-Elemente geben ihren Inhalten unterschiedliche Bedeutungen. Bei einer semantischen HTML-Auszeichnung stimmen die Bedeutungen des Inhaltes mit der Semantik des HTML-Elements überein. So kann ein Text beispielsweise mittels CSS fett gedruckt werden und eine höhere Schriftgröße erhalten, Suchmaschinen erkennen diesen jedoch nur als Überschrift, wenn er sich in einem `<h1>` Element befindet.[21] Zudem sollte darauf geachtet werden, dass zur Erzeugung von Interaktivität sogenanntes „Unobtrusive JavaScript“ (dt. „unaufdringliches JavaScript“) verwendet wird. So können Event-Handler z.B. direkt im HTML über das Attribut `onclick` definiert werden. Für ein Progressive Enhancement sollte stattdessen das Skript selbstständig aktiv werden und die Ereignisüberwachung starten, indem es beispielsweise das betroffene Element über seine ID ermittelt und einen Event-Handler registriert.[22] Zuletzt sollten für eine Abwärtskompatibilität Feature Tests genutzt werden. Dabei handelt es sich um JavaScript Code, der prüft, ob der aktuell verwendete Browser eine bestimmte Funktionalität unterstützt.[23] Eine umfangreiche Bibliothek vorgefertigter Feature Tests bietet Modenizr.[24]

2.3.Progressive Web Apps

Nachdem nun die verschiedenen Typen von Apps und das Progressive Enhancement vorgestellt wurde, können die PWAs in diesen Kontext eingebettet werden.

2.3.1. Architektur und Eigenschaften von Progressive Web Apps

In der Abbildung 4 unten ist die Architektur von PWAs dargestellt. Diese unterscheidet sich von der Architektur von Web Apps lediglich in zwei Komponenten: dem Service Worker und dem Web App Manifest. Entsprechend ist auch die Funktionsweise von PWAs nahezu identisch zu der von Web Apps. Es ist also bereits erkennbar, dass es sich bei PWAs um eine erweiterte Form von herkömmlichen Web Apps handelt.

Der Service Worker und das Web App Manifest werden im folgenden Kapitel 2.3.2 detailliert beschrieben, zunächst sollen jedoch die Eigenschaften von PWAs genauer betrachtet werden. Google selbst beschreibt PWAs durch drei Wörter: Reliable, Fast und Engaging.[25] Ein zuverlässiges (eng. „reliable“) und sofortiges Laden einer Webseite, selbst bei schlechter Netzwerkverbindung, soll durch die Service Worker sichergestellt werden. Grob beschrieben agiert der Service Worker als clientseitiger Proxy und kann somit selbst bei fehlender Verbin-

[21] Vgl. SELFHTML (2018c), Abschnitt „Semantik“.
[22] Vgl. SELFHTML (2018a), Abschnitt „unaufdringliches JavaScript“.
[23] Vgl. Maurice, Florence (2012), S. 18.
[24] Vgl. Ates, Faruk u.a. (o.J.).
[25] Vgl. Google Developers (o.J.).

dung Anfragen aus dem Cache beantworten. Zudem sollen PWAs schnell (eng. „fast“) und unmittelbar auf Nutzerinteraktionen reagieren und Animationen flüssig abspielen. Mit Engaging ist gemeint, dass eine PWA den Anwender zur Nutzung verleitet. Das soll z.B. über Push Notifications, das Look & Feel einer nativen App oder die durch das Web App Manifest ermöglichte Installation auf den Startbildschirm des Gerätes erfolgen.

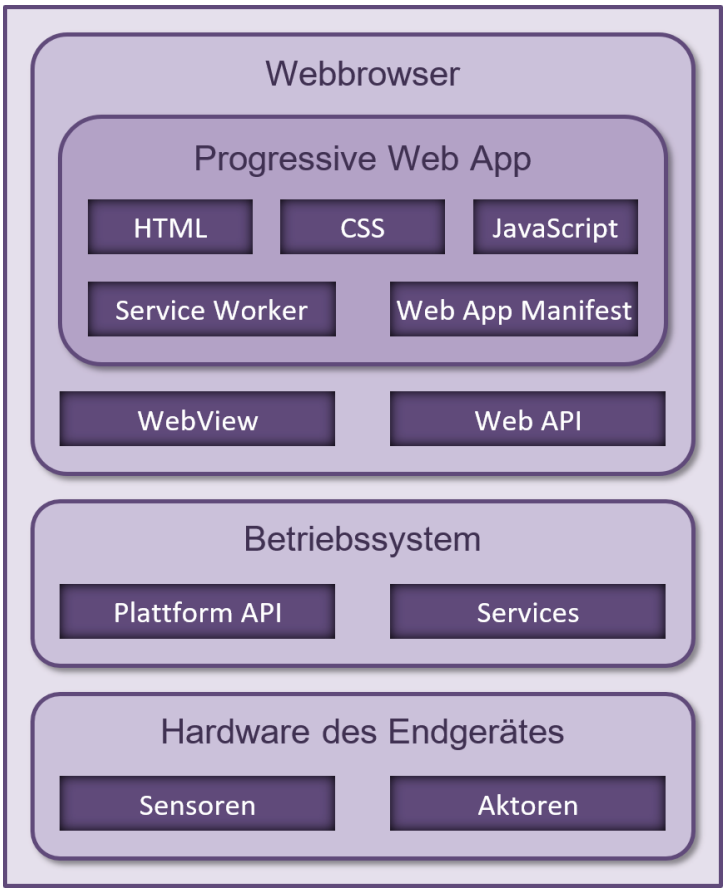

Abbildung 4: Architektur von Progressive Web Apps
Quelle: eigene Darstellung

Alex Russell, ein Senior Softwareentwickler im Google Chrome Team, führt neben diese drei Eigenschaften noch weitere auf[26], die ebenfalls durch andere Quellen bestätigt werden.[27] So sollen sich PWAs in jedem Fall responsiv verhalten, um sich der Bildschirmgröße eines jeden Endgerätes anzupassen. Wichtig ist, dass sich PWAs bei der Nutzung moderner Webtechnologien, wie z.B. Service Worker, progressiv verhalten (siehe Kapitel 2.2 Progressive Enhancement und Graceful Degradation), um ältere Browser nicht auszuschließen und bei modernen Browsern eine möglichst gute Nutzererfahrung zu gewährleisten. Durch eine obligatorische HTTPS-Verbindung (Hypertext Transfer Protocol Secure) sollen PWAs ein Mindestmaß an Sicherheit mit sich bringen. Eine weitere Eigenschaft „Discoverable“ (dt. entdeckbar) be-

[26] Vgl. Russell, Alex (2015), Abschnitt „Escaping the Tab: Progressive, Not Hybrid“.
[27] Vgl. Liebel, Christian (2017a).

zieht sich darauf, dass PWAs durch Suchmaschinen gefunden und durch das Web App Manifest von herkömmlichen Web Apps unterschieden werden können. Zuletzt sollen alle Seiten einer PWA mittels einer URL verlinkt werden können. Diese Eigenschaft erscheint zunächst trivial, ist jedoch für SPAs relevant, bei denen die dynamisch nachgeladenen Zustände referenzierbar sein sollen.

2.3.2. Progressive Web App Technologien

In den oben beschriebenen Eigenschaften stecken bereits einige Technologien, durch die sich PWAs von herkömmlichen Web Apps unterscheiden.

2.3.2.1. Service Worker

Die wohl wichtigste, technische Neuerung ist dabei der Service Worker. Dieser ist eine Spezialform des Web Workers. Eine sehr lange Zeit gab es nur die Möglichkeit, Programmlogik in einem Browser sequenziell abzuarbeiten, da dem JavaScript-Kontext vom Browser nur ein Thread zur Verfügung gestellt wurde. Das sogenannte Multithreading, also die parallele Ausführung von Programmcode, wurde durch den Web Worker ermöglicht. Bei diesem handelt es sich um ein Skript, das in einem eigenen Thread im Hintergrund läuft. Der Web Worker kann nicht so, wie der Main Thread im Vordergrund, auf das Document Object Model (DOM) der Seite zugreifen, er kann jedoch mit dem Main Thread über eine nachrichtenbasierte Schnittstelle kommunizieren. Ein Web Worker wird dabei einem Browserfenster bzw. einer Registerkarte zugeordnet und wird beendet, wenn der Tab geschlossen wird.[28]

Der Service Worker ist eine spezielle Form des Web Workers. Bei ihm handelt es sich ebenfalls um ein eigenständiges JavaScript, das in einem separaten Thread im Hintergrund ausgeführt wird und keinen Zugriff auf das DOM der Seite hat. Anders als der Web Worker ist der Service Worker jedoch nicht an die Webanwendung gebunden, sondern läuft im Hintergrund weiter, auch wenn die Seite geschlossen wurde. Das erlaubt dem Service Worker, unabhängig von der Webseite auf verschiedene Ereignisse zu reagieren und Netzwerkanfragen zu unterbrechen und zu bearbeiten. Dadurch kann der Service Worker als clientseitiger Proxy zwischen Webanwendung und Webserver agieren. Da der Service Worker somit sehr hohe Rechte bzw. Privilegien genießt, ist aus Sicherheitsgründen eine Übertragung mittels HTTPS eine Voraussetzung. Die Ausnahme ist hierbei ein lokaler Test über `localhost`, bei dem eine unverschlüsselte HTTP Verbindung erlaubt ist.[29] Da die Service Worker Technologie ver-

[28] Vgl. Liebel, Christian (2017b), Abschnitt „Abgrenzung zum Web Worker“.
[29] Vgl. Braun, Herbert (2017), S. 30 f.

gleichsweise modern ist, wird sie noch nicht von allen Browsern unterstützt. An dieser Stelle muss das Progressive Enhancement angewandt werden, indem durch Feature Tests geprüft wird, ob der aktuelle Browser Service Worker unterstützt. Bei Bedarf muss eine Alternative implementiert werden. Google, welches die PWAs am meisten vorangetrieben hat, unterstützt mit seinem Chrome Browser die Service Worker seit dem Release 40 vom 21.01.2015 am längsten.[30] Darauf folgt Mozilla, dessen Browser Firefox eine Unterstützung ab Version 44 vom 26.01.2016 bietet.[31] Erst seit Anfang 2018 sind Service Worker in Apples Safari Browser mit der iOS Version 11.3 funktionsfähig[32] und zuletzt hat Microsoft die Service Worker mit einem Update am 10.04.2018 in den Edge Browser integriert.[33]

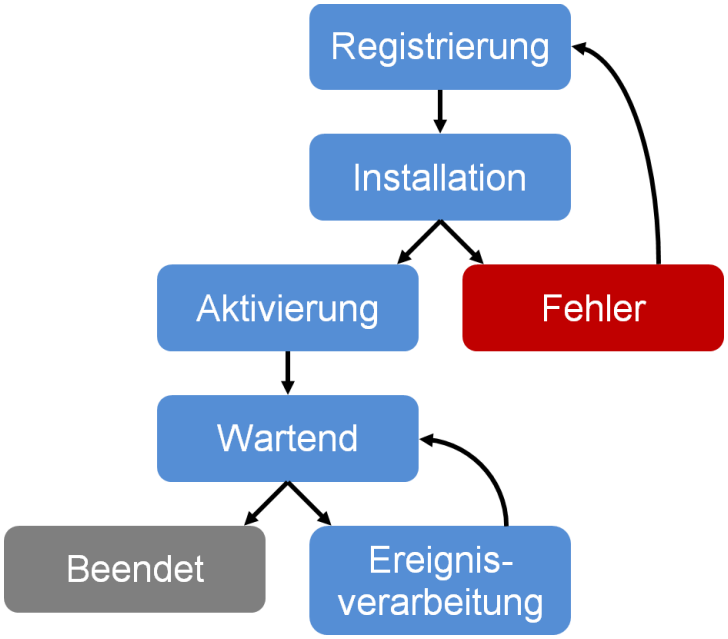

Abbildung 5: Lebenszyklus eines Service Worker
Quelle: eigene Darstellung in Anlehnung an Gaunt, Matt (2018b), Abschnitt „The service worker life cycle“

Um die Funktionsweise und die Fähigkeiten eines Service Worker besser verstehen zu können, sollte sein in Abbildung 5 visualisierter Lebenszyklus erläutert werden. Der Zyklus ist unabhängig von der eigentlichen Webanwendung. Diese muss jedoch zunächst im eigenen JavaScript Code den Service Worker registrieren. Durch die Registrierung wird dieser im Hintergrund installiert. Während der Installation können im Service Worker individuelle Logiken abgearbeitet werden, wie z.B. ein initiales Laden und Cachen von Ressourcen. Tritt bei der Installation ein Fehler auf, wird diese durch eine erneute Registrierung bei einem erneuten Laden der Webseite wiederholt. Eine mehrfache Registrierung ist dabei unproblematisch, da der Browser denselben Service Worker erkennt und nur einmalig installiert. Läuft aktuell kein

[30] Vgl. Chrome Platform Status (2018).
[31] Vgl. Mozilla (o.J.), Abschnitt „developer“.
[32] Vgl. Fablet, Youenn (2018).
[33] Vgl. Pflug, Kyle & McCormick, Libby (2018), Abschnitt „Offline web sites and push notifications“.

zuvor installierter Service Worker, erfolgt die Aktivierung, in welcher z.B. Ressourcen aus einem alten Cache gelöscht werden können. Ist der Service Worker aktiv, kontrolliert er alle Seiten in einem definierten Anwendungsbereich. Dieser Anwendungsbereich kann auf Unterseiten beschränkt werden, aus Sicherheitsgründen jedoch nicht über die Domäne hinausgehen, aus der der Service Worker geladen wird. Ein aktivierter Service Worker verarbeitet eine Reihe definierter Ereignisse. Treten momentan keine Ereignisse auf, wird er pausiert und verbraucht keine Rechenleistung. Soll die Logik eines Service Workers aktualisiert werden, muss sich der neue Quellcode lediglich vom alten unterscheiden und er muss erneut registriert werden. Nach der Installation wird der neue Service Worker jedoch in einer Warteschlange eingereiht. Erst, wenn alle Seiten, die vom alten Service Worker kontrolliert werden, geschlossen wurden, wird der alte beendet und der neue aktiviert.[34]

Eines der Ereignisse, das nun verarbeitet werden kann, ist das `fetch` Event. Dieses wird aufgerufen, wenn die Webanwendung eine Ressource vom Webserver anfragt. Der Service Worker kann nun diese Anfrage unterbrechen und die Ressource nicht über das Netzwerk sondern aus Cache laden. Werden die Ressourcen der Webanwendung nach dem sogenannten App Shell Modell strukturiert, ist es in Kombination mit dem Cache möglich, die Seite offline bereitzustellen. Nach diesem Modell wird die statische Hülle der Webanwendung von den dynamischen Daten getrennt. Die Ressourcen für die Hülle ändern sich nur selten und können somit im Cache gespeichert werden. Zusammen mit einem initialen Satz an Daten kann die Webanwendung dann aus dem Cache beschleunigt geladen und selbst ohne Netzwerkverbindung gestartet werden. Aktuellere Daten können dann bei Bedarf und vorhandener Verbindung nachgeladen werden.[35] An diesem Beispiel ist auch gut erkennbar, dass abhängig von der Art der Ressource unterschiedliche Methoden bzw. Strategien zum Caching verwendet werden sollten. Jake Archibald, ein Entwickler der Google Chrome Teams, stellt in seinem „Offline Cookbook“ eine Reihe von Strategien vor, die im Zusammenhang mit Caching genutzt werden können.[36] Da diese Strategien lediglich eine kleine Rolle für das Verständnis der später entwickelten Leitfaden-App spielen, werden sie im Anhang A beschrieben.

2.3.2.2. Push Notifications

Neben dem `fetch` Event kann der Service Worker ebenfalls das `push` Event verarbeiten. Dieses Ereignis wird aufgerufen, wenn der Worker eine Push Notification für seine Webseite erhält. Mit Push ist dabei der Prozess gemeint, durch den Informationen vom Webserver über

[34] Vgl. Gaunt, Matt (2018b), Abschnitt „The service worker life cycle“.
[35] Vgl. Lynch, Max (2016), Abschnitt „Service Workers“.
[36] Vgl. Archibald, Jake (2014), Abschnitt „Serving suggestions - responding to requests“.

einen Push Service an den Client gesendet werden. Hat der Client bzw. der Service Worker die Informationen erhalten, erfolgt die Notification, also die Anzeige der Nachricht beim Nutzer.[37]

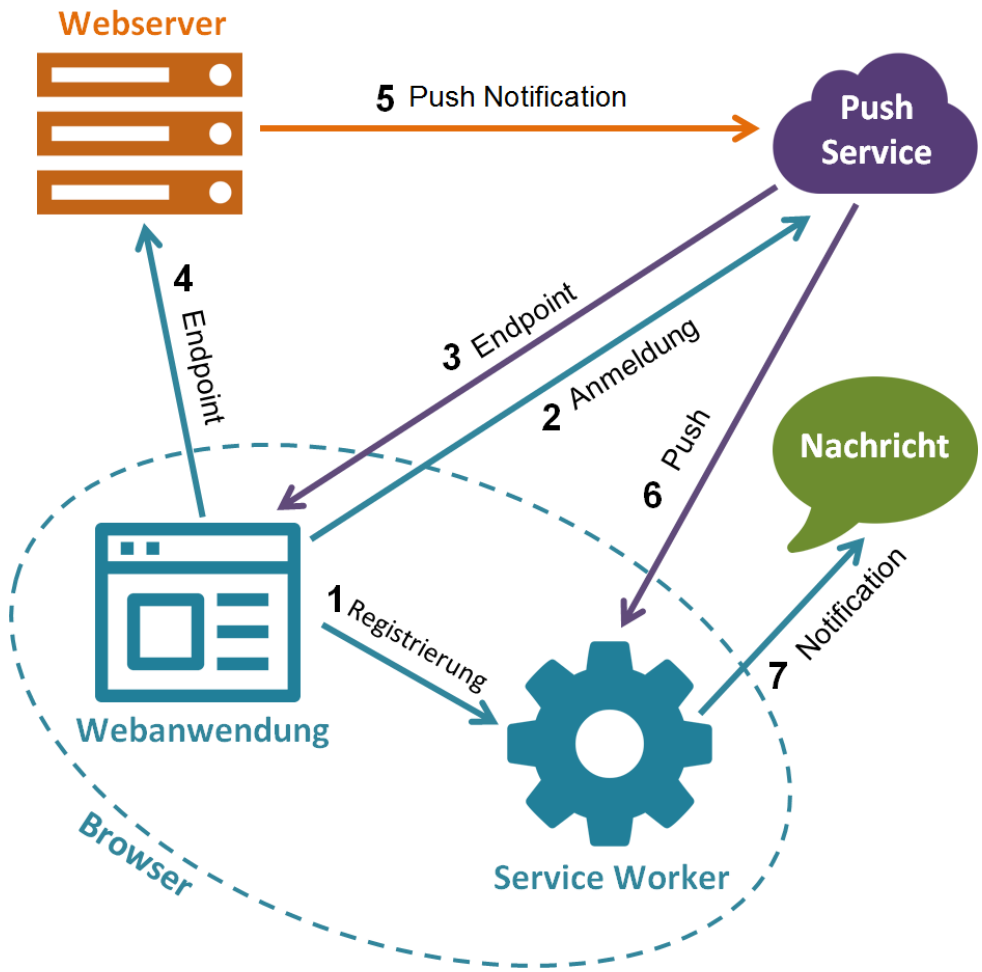

Abbildung 6: Funktionsweise von Push Notifications
Quelle: eigene Darstellung in Anlehnung an Braun, Herbert (2018), S. 176

Das Empfangen des Pushs und die Verarbeitung der Notification ist jedoch ein Schritt, der einiges an Vorbereitung benötigt. Die vollständige Funktionsweise von Push Notifications soll anhand der Abbildung 6 erläutert werden. Nachdem die Webseite vom Webserver geladen wurde, muss sie im ersten Schritt einen Service Worker registrieren, der die `push` Events verarbeiten kann. Im folgenden Schritt zwei muss sich die Webanwendung bei einem Push Service anmelden. Der Push Service verwaltet die Anfragen zum Versenden einer Push Notification und deren Empfänger. Für die Anmeldung am Push Service wird ein Server Schlüssel benötigt. Anhand dessen kann der Push Service erkennen, ob der Webserver berechtigt ist, einer Webseite eine Push Notification zu senden. Mit der Anmeldung erhält die Webanwendung im dritten Schritt eine URL, einen sogenannten Endpoint, durch den der Empfänger einer Push Notification bestimmt werden kann. Diesen Endpoint muss der Client im vierten

[37] Vgl. Medley, Joseph (2018), Abschnitt „Two technologies“.

Schritt an den Server schicken, um von diesem Nachrichten erhalten zu können.[38] Will der Webserver nun im fünften Schritt eine Push Notification versenden, muss dieser einer Anfrage an die API des Push Dienstes stellen. Das Format der Anfrage wird dabei durch das Web Push Protocol[39] definiert. Eine Anfrage enthält verschiedene Informationen, wie z.B. den eigentlichen Inhalt der Nachricht, den Empfänger über den Endpoint und Optionen zum Versand der Nachricht. Die Nachricht an sich muss verschlüsselt sein, sodass der Push Dienst diese nicht auslesen kann. Als Versandoptionen können beispielsweise die time-to-live, die Dringlichkeit der Nachricht oder ein Thema angegeben werden, unter welchem mehrere ähnliche Nachrichten zusammengefasst werden. Die time-to-live kommt zum Einsatz, wenn der Empfänger offline ist und die Nachricht nicht zugestellt werden kann. Kann der Push Service die Nachricht im sechsten Schritt bis zum Ablauf der angegeben Zeit nicht zustellen, wird diese gelöscht.[40] Wurde die Nachricht jedoch erfolgreich versandt, wird im Service Worker das `push` Event aufgerufen. Bei dessen Verarbeitung kann nun der Inhalt der Nachricht ausgelesen und im siebten Schritt dem Nutzer angezeigt werden.[41]

2.3.2.3. Web App Manifest

Die letzte der drei PWA-Technologien ist das Web App Manifest. Dabei handelt es sich um eine Datei, die verschiedene Informationen zur App enthält, die dafür benötigt werden, um eine PWA sowohl vom Aussehen als auch vom Verhalten näher an eine native App zu bringen. Zudem kann eine PWA durch das Manifest von einer herkömmlichen Web App unterschieden werden. Die Manifest-Datei wird in der JavaScript Object Notation (JSON) erstellt und über ein `<link>` Element in den Metadaten der Webseite eingebunden. Bei dieser Einbindung kommt wieder das Progressive Enhancement zum Einsatz, da ältere Browser das ihnen unbekannte Manifest einfach ignorieren. Konkret enthält das Manifest Informationen wie Namen und Icons zur App, Themen- und Hintergrundfarben, einen Anzeigemodus oder eine Start URL.[42] Eine vollständige Liste aller Attribute kann dem Anhang B Inhalt des Web App Manifest nach dem W3C Standard entnommen werden. Diese Informationen werden benötigt, um die PWA wie eine native App auf das Endgerät installieren zu können. So kann die PWA beispielsweise nach erfolgter Installation vom Startbildschirm gestartet werden. Beim Start wird ein Splashscreen mit den im Manifest definierten Namen und Icon angezeigt. Anschließend wird die App anhand der Start-URL aufgerufen und abhängig vom Anzeigemo-

[38] Vgl. Gaunt, Matt (2018a), Abschnitt „Step 1: Client Side“.
[39] Vgl. IETF (2016).
[40] Vgl. Gaunt, Matt (2018a) , Abschnitt „Step 2: Send a Push Message“.
[41] Vgl. ebenda, Abschnitt „Step 3: Push Event on the User’s Device“.
[42] Vgl. Gaunt, Matt & Kinlan, Paul (2018).

dus im Vollbild, mit dem Rahmen einer nativen App oder im Browserfenster dargestellt. Ergänzend sollte erwähnt werden, dass es sich aus technischer Sicht um keine echte Installation handelt, denn Funktionalitäten wie der Offlinebetrieb oder die Push Notifications stehen auch ohne Installation im Browser zur Verfügung. Die Installation einer PWA kann eher mit einem erweiterten Lesezeichen in einem Browser verglichen werden.[43] Um jedoch eine PWA überhaupt installieren zu können, müssen einige Bedingungen erfüllt werden. Diese Voraussetzungen unterscheiden sich zwar im Detail von Browser zu Browser, in den meisten Fällen muss die App jedoch über eine HTTPS-Verbindung übertragen werden, einen Service Worker registrieren und ein minimal gefülltes Web App Manifest bereitstellen. Google verlangt für den Chrome Browser darüber hinaus, dass eine Heuristik zur Nutzung der App erfüllt sein muss. Eine detaillierte Auflistung der Bedingungen getrennt nach Browser kann dem Anhang C entnommen werden. Sind alle Voraussetzungen erfüllt, zeigt der Browser automatisch ein Installationsbanner an, und der Nutzer kann die PWA über einen „Add to Home Screen“ Button auf den Startbildschirm installieren.[44]

2.3.3. Definition von Progressive Web Apps

Nachdem nun alle theoretischen Grundlagen zum Verständnis von PWAs erläutert wurden, kann eine Definition zu diesem Begriff für die Masterarbeit festgelegt werden. Die Basis von PWAs sind Webanwendungen, die mit den Webtechnologien HTML, CSS und JavaScript entwickelt wurden und auf die mittels URL über das Internet zugegriffen werden kann. Ein weiterer, zugrunde liegender Gedanke der PWAs ist das Progressive Enhancement. Demnach werden moderne Technologien nur genutzt, wenn sie vom Browser unterstützt werden. Dadurch wird eine fortschreitende Verbesserung der Anwendung und der Nutzererfahrung gewährleistet. Hinzu kommen weitere Eigenschaften. So beschreibt Google PWAs als zuverlässig, schnell und zur Nutzung motivierend. Alex Russell ergänzt eine responsive Oberfläche für alle Endgeräte, eine sichere Verbindung über HTTPS sowie die Entdeck- und die Verlinkbarkeit der Progressiv Web App. Primär kommen Technologien wie Service Worker, Push Notifications und das Web App Manifest zum Einsatz. Zuletzt wäre für die Definition noch die Installation von PWAs relevant, deren Voraussetzungen eine HTTPS-Verbindung, mindestens ein registrierter Service Worker und die Angabe eines minimalen Web App Manifests sind.

---

[43] Vgl. Liebel, Christian (2017c), Abschnitt „Ein Banner fordert zur Installation auf“.
[44] Vgl. LePage, Pete (2018).

Das ist nun eine große Menge an Eigenschaften, die in einer prägnanten Definition in Gänze keinen Platz findet und somit auf das Wesentliche reduziert werden muss. So ist beispielsweise der Begriff der Webanwendung sehr geläufig, wodurch nicht explizit auf die einzelnen Webtechnologien bzw. auf den Zugriff über das Internet mittels URL eingegangen werden muss. Das Progressive Enhancement hingegen ist nicht ganz so geläufig und muss somit erläutert werden. Anstelle einer ausführlichen Beschreibung, wie im Kapitel 2.2 Progressive Enhancement und Graceful Degradation, wird eine kurze und aussagekräftige Formulierung genutzt. Weiterhin kann auf die Eigenschaften „entdeckbar“ und „verlinkbar“ verzichtet werden. Ersteres wird indirekt durch das Web App Manifest gewährleistet und Letzteres ist inhärent im Begriff der Webanwendung enthalten.

Auf Basis dieser Vorüberlegungen soll nun die folgende Definition gelten:

> *„Progressive Web Apps sind responsive und per HTTPS übertragene Webanwendungen, die nach dem Grundsatz des Progressive Enhancement die Fähigkeiten der Browser für eine fortschreitende Verbesserung nutzen, wodurch mittels Offlinefunktionalität über Service Worker, eine Installation anhand eines Web App Manifests und Push Notifications eine zuverlässige, motivierende und native Nutzererfahrung gewährleistet wird.“*

Eine internationale Definition in englischer Sprache kann dem Anhang D entnommen werden.

3. Evaluationsmethodiken

Nachdem die Theorie hinter PWAs beleuchtet wurde, sollen nun einige Evaluationsmethodiken vorgestellt werden. Eine Evaluation ist notwendig, um dem im Rahmen der Arbeit entstehenden Regelwerk und der Leitfaden-App eine wissenschaftliche Validität zu verleihen. Zur Bildung einer theoretischen Grundlage werden primär die Begriffe und Methoden der empirischen Sozialforschung vorgestellt. Wem die Theorie dieser Thematik geläufig ist, der kann dieses Kapitel überspringen.

3.1. Grundlagen empirischer Sozialforschung

3.1.1. Grundbegriffe der empirischen Sozialforschung

Bei der empirischen Sozialforschung werden Daten über soziale Sachverhalte über geeignete Methoden erhoben und ausgewertet. Die Eignung der Methode ist dabei vom Untersuchungsziel, der konkreten Fragestellung in den verfügbaren Ressourcen abhängig. Häufig ist bei der empirischen Sozialforschung eine Triangulation empfehlenswert, bei der mehrere Methoden kombiniert werden, um dieselbe Frage zu beantworten. Dies hat den Vorteil, dass das Problem aus verschiedenen Blickwinkeln beleuchtet wird. Neben der Gewinnung von Erkenntnissen hat die empirische Sozialforschung drei Aufgaben: Deskription, Exploration und Prüfung von Hypothesen. Eine deskriptive Untersuchung hat zum Ziel, einen Sachverhalt zu beschreiben, indem die erhobenen Daten klassifiziert und visualisiert werden. Bei einer explorativen Untersuchung werden bisher unbekannte Phänomene und Zusammenhänge erforscht, um daraus Hypothesen zu generieren. In der dritten, hypothesenprüfenden Untersuchungsart wird die Validität einer Hypothese durch Prüfung von Merkmals- und Variablenzusammenhängen festgestellt.[45]

Bevor jedoch Zusammenhänge geprüft werden können, muss in einem ersten Schritt ermittelt werden, wie sich der zu untersuchende, natürliche Sachverhalt durch empirische Daten beschreiben lässt. Dieser Vorgang nennt sich *Operationalisierung* und ist notwendig, um die erhobenen Daten statistisch auswerten zu können.[46] Dazu müssen dem zu erforschenden Gegenstand, auch *Merkmalsträger* genannt, Eigenschaften zugeordnet werden. Merkmalsträger sind häufig Menschen, es kommen aber auch alle anderen Arten von physischen und nichtphysischen Objekten (z.B. Autos, Softwareprogramme) in Frage. Die zu untersuchende Eigenschaft eines Merkmalsträgers wird *Merkmal*, oder auch *Variable* genannt. Als Beispiele

[45] Vgl. Berger-Grabner, Doris (2016), S. 109 ff.
[46] Vgl. Weber, Daniela (2015), S. 123 ff.

können hier das Alter, das Baujahr oder die Anzahl der Codezeilen dienen. Die *Merkmalsausprägung* beschreibt den Wert, den ein Merkmal annehmen kann. Besitzt ein Merkmal Zahlen als Werte (z.B. Alter, Baujahr), dann wird es auch als *quantitatives Merkmal* klassifiziert. Können die Werte jedoch nur in Kategorien eingeordnet werden (z.B. Geschlecht, Bildungsabschluss), dann handelt es sich um ein *qualitatives Merkmal*. Zudem können Merkmale auch in *abhängige* und *unabhängige Variablen* unterschieden werden. Diese Unterscheidung ist bei der Aufstellung von Hypothesen relevant, in denen abhängige Variablen durch den Einfluss unabhängiger Variablen erklärt werden sollen. Wird beispielsweise die Hypothese „Rauchen verkürzt die Lebensdauer" aufgestellt, dann soll untersucht werden, inwiefern die abhängige Variable „Lebensdauer" durch die unabhängige Variable „Intensität des Rauchens" beeinflusst wird.[47,48]

Bevor nun eine Datenerhebung durchgeführt werden kann, muss festgelegt werden, welche Merkmalsträger untersucht werden sollen. Die Menge aller Merkmalsträger, die für das Forschungsziel relevante Merkmale bzw. Ausprägungen besitzen, wird *Grundgesamtheit* genannt. Die Untersuchung der kompletten Grundgesamtheit wird *Voll-* bzw. *Totalerhebung* genannt. Problematisch dabei ist jedoch, dass die Grundgesamtheit häufig sehr groß ist, wodurch eine Totalerhebung zu aufwändig, zu teuer oder organisatorisch nicht handhabbar wird. Aus diesem Grund findet oft nur eine *Teilerhebung* statt, bei der mit einer Teilmenge der Grundgesamtheit eine repräsentative *Stichprobe* untersucht wird. Die *Repräsentanz* ist die Voraussetzung dafür, dass Rückschlüsse von der Stichprobe zur Grundgesamtheit gezogen werden können. Eine Stichprobe ist dann repräsentativ, wenn die Verteilung all ihrer relevanten Merkmale mit der in der Grundgesamtheit übereinstimmt. Ist dies nicht der Fall, kommt es zu einem sogenannten Stichprobenfehler, der die Unsicherheit der Untersuchungsergebnisse mit sich zieht.[49,50]

3.1.2. Messniveaus

Bei der oben bereits genannten Operationalisierung wird ermittelt, wie sich natürliche Sachverhalte als empirische Daten ausdrücken lassen. Die Form, in der sich diese Daten erheben lassen, wird *Messniveau* genannt. Das Messniveau ist ausschlaggebend dafür, welche Methoden zur Datenerhebung geeignet sind und welche Auswertungen vorgenommen werden können. Grundsätzlich lässt sich das Messniveau in metrische und nicht-metrische Daten eintei-

[47] Vgl. Töpfer, Armin (2012), S. 228 ff.
[48] Vgl. Berger-Grabner, Doris (2016), S. 120 f.
[49] Vgl. Magerhans, Alexander (2016), S. 73 ff.
[50] Vgl. Kuß, Alfred u.a. (2018), S. 67 ff.

len. Bei ersteren handelt es sich um zahlenmäßige Daten, bei letzterem lassen sich die Merkmalsausprägungen nur in Kategorien einordnen. Das Messniveau mit dem niedrigsten Informationsgehalt wird durch die *Nominalskala* beschrieben. Einer nominalskalierten Variabel können nur nicht-metrische Daten zugeordnet werden. Die Werte lassen sich in Kategorien einordnen, wodurch eine Aussage über die Gleichheit bzw. Verschiedenheit der Werte gemacht werden kann. Als Beispiel kann hier das Merkmal „Geschlecht“ mit den Ausprägungen „männlich“ und „weiblich“ genannt werden. Variablen einer *Ordinalskala* erlauben ebenfalls nur nicht-metrische Daten, ihr Informationsgehalt ist jedoch etwas höher, da sich die Kategorien der ordinalskalierten Werte in eine Rangfolge bringen lassen. Somit lässt sich ausdrücken, dass ein Wert größer oder kleiner als ein anderer ist. So lassen sich die möglichen Werte „Realschulabschluss, Hochschulreife, Bachelor, Master, usw.“ des ordinalskalierten Merkmals „Bildungsabschluss“ in eine Rangfolge bringen. Die *Intervallskala* lässt nun mit metrischen Daten Zahlen als Werte zu. Der Informationsgehalt steigt weiter an, da die Werte nun konstante und bedeutungsvolle Abstände besitzen. Diese Eigenschaft ist Voraussetzung für verschiedene Rechenoperationen, die im Kapitel 3.3.1 Quantitative Datenauswertung vorgestellt werden. Intervallskalierte Merkmale sind beispielsweise Jahreszahlen oder Temperaturangaben in Celsius. Das Messniveau mit dem höchsten Informationsgehalt ist die metrische *Ratio*- bzw. *Verhältnisskala*. Zusätzlich zu der Bedeutung der Werteabstände besitzen verhältnisskalierte Werte einen natürlichen Nullpunkt, wodurch weitere Rechenoperationen ermöglicht werden. Variablen einer Verhältnisskala können z.B. Längen-, Flächen- und Volumenwerte oder Temperaturangaben in Kelvin annehmen.[51,52]

3.1.3. Quantitative vs. qualitative Sozialforschung

Die empirische Sozialforschung lässt sich abhängig vom Forschungsziel und von den zu untersuchenden Daten in die quantitative und qualitative Sozialforschung einteilen. Die *quantitative Sozialforschung* versucht, einen Sachverhalt mit Hilfe von metrischen, also zahlenmäßigen Daten zu untersuchen. Sie findet primär bei der Prüfung von Hypothesen Anwendung. Es wird ein stark strukturiertes und standardisiertes Vorgehen genutzt, bei dem die Datenerhebung immer unter denselben Voraussetzungen stattfindet. Dies wird durch eine einseitige Kommunikation gewährleistet, bei der ein Proband z.B. nur die vom Forscher gestellten Fragen beantwortet. Durch die starke Strukturierung sind die quantitativen Daten ohne Probleme vergleichbar, wodurch die Auswertung großer Stichproben möglich ist. Daraus folgen objek-

[51] Vgl. Töpfer, Armin (2012), S. 230 ff.
[52] Vgl. Magerhans, Alexander (2016), S. 91 ff.

tive und repräsentative Ergebnisse, die aus einer statistischen Auswertung hervorgehen. Die Ergebnisse quantitativer Forschung bestehen also aus der Beschreibung eines Sachverhaltes in Zahlen und dem Test von Hypothesen durch Prüfung statistischer Zusammenhänge.

Die *qualitative Sozialforschung* versucht hingegen, nicht-metrische, also in Worten beschriebenen Zusammenhänge zu verstehen und zu interpretieren. Sie findet hauptsächlich in der Erforschung von Ursachen und von unbekannten Sachverhalten Anwendung. Das Vorgehen ist im Vergleich zu quantitativen Methoden offen und flexibel, um die individuellen Sichtweisen und subjektiven Meinungen der befragten Personen erfassen zu können. Dabei findet eine interaktive Kommunikation statt, bei der der Forscher auf den Probanden eingeht. Durch dieses individuelle Vorgehen kann ein wesentlich tieferer Informationsgehalt erhoben werden. Gleichzeitig sorgt diese Methode aber dafür, dass jeder Erhebungsprozess unterschiedlich verläuft, wodurch die qualitativen Daten nur schwer vergleichbar sind. Die Folge sind ein hoher Aufwand und hohe Kosten, wodurch nur eine recht kleine Stichprobe bearbeitet werden kann. Die Ergebnisse sind dadurch sehr subjektiv und nicht repräsentativ. Hinzu kommt, dass der Forscher für die Interaktion mit den Probanden eine hohe, sozial-methodische Kompetenz benötigt. Weil statistische Methoden nicht auf qualitative Daten angewandt werden können, müssen diese über eine Interpretation ausgewertet werden. Ergebnisse qualitativer Forschung sind also Beschreibungen von Sachverhalten, aus deren Interpretation Hypothesen generiert werden. Diese werden anschließend häufig mittels quantitativer Methoden getestet.[53,54] Ein Überblick über quantitative und qualitative Forschung kann der Tabelle 11 im Anhang E entnommen werden.

3.1.4. Gütekriterien empirischer Sozialforschung

Die Tatsache, ob eine empirische Forschung aus wissenschaftlicher Sicht legitim verläuft, wird anhand der drei Gütekriterien Objektivität, Reliabilität und Validität gemessen. Die *Objektivität* besagt, dass eine Untersuchung frei von subjektiven Einflüssen des Forschers ist. Sie ist gegeben, wenn unterschiedliche Personen bei der Nutzung derselben Methoden auch dasselbe Ergebnis erhalten. Die *Reliabilität* hingegen beschreibt die Genauigkeit, mit der eine Messmethode den Wert einer Variabel bestimmt. Sie ist entsprechend gegeben, wenn eine Methode bei mehreren Messungen konsistent denselben Wert liefert. Die Objektivität des Methodenanwenders gilt dabei als Voraussetzung, um die Reliabilität der Methode bestimmen zu können. Die *Validität* als letzte der drei Kriterien gibt an, ob eine Methode auch wirklich

[53] Vgl. Berger-Grabner, Doris (2016), S. 117 ff.
[54] Vgl. Magerhans, Alexander (2016), S. 69 ff.

den Sachverhalt misst, den sie zu messen versucht. Voraussetzung für die Ermittlung der Validität einer Methode ist deren Reliabilität.[55,56] Die drei Gütekriterien können anhand einer Körperwaage veranschaulicht werden. Zeigt eine Waage anstatt dem Körpergewicht die Raumtemperatur an, dann ist die Messung nicht valide. Zeigt die Waage das Gewicht an, schwankt dabei jedoch immer um bis zu fünf Kilogramm, dann ist das Messergebnis nicht sehr zuverlässig (reliable). Die sich wiegende Person (Forscher) geht objektiv bei der Messung vor, wenn sie sich mit beiden Beinen auf die Waage stellt und sich nirgends abstützt.

Die drei oben genannten Gütekriterien können allerdings nur in der quantitativen Forschung gewährleistet werden. Die qualitative Forschung hingegen erlaubt vom Ansatz her bereits keine objektive Datenerhebung mit zuverlässigen Methoden. Aus diesem Grund wird der so durchaus gewollte Ansatz häufig kritisch betrachtet. Um trotzdem eine gewisse, wissenschaftliche Validität der Ergebnisse zu gewährleisten, müssen andere Gütekriterien erfüllt werden. Da der Forschungsprozess aufgrund seiner fehlenden Standardisierung kaum repliziert werden kann, muss er mitsamt seinen Entscheidungen, Problemen und Ergebnisse genauestens dokumentiert werden. Auf diese Weise besteht eine *intersubjektive Nachvollziehbarkeit*, durch welche anderen Forscher und Gutachter den Prozess mental rekonstruieren und beurteilen können. Während der Datenerhebung sollte eine *Nähe zum Untersuchungsgegenstand* aufgebaut werden, weil sich der Forscher so besser in die Gedankenwelt der befragten Person hineinversetzen und seine subjektiven Aussagen besser verstehen und interpretieren kann. Um die Ergebnisse zusätzlich abzusichern, sollte eine *kommunikative Validierung* vorgenommen werden. Dabei werden die erhobenen Daten und deren Interpretation mit dem Probanden durchgegangen und von ihm bestätigt. Zuletzt sollte der Forscher immer eine *argumentative Interpretationsabsicherung* vornehmen. Da qualitative Daten abhängig vom Vorwissen unterschiedlich interpretiert werden können, sollte die gewählte Interpretation mit gültigen Argumenten untermauert werden.[57,58]

3.2. Methoden zur empirischen Datenerhebung

Nachdem nun die Grundlagen zur empirischen Forschung geklärt sind, sollen die wichtigsten Methoden zur Erhebung von Daten vorgestellt werden. Die mit am häufigsten und dadurch relevantesten Methoden sind das Interview und die Beobachtung. Zusätzlich wird noch die Inhaltsanalyse vorgestellt, die in der quantitativen Forschung zur Erhebung, und in der qualitativen Forschung zur Auswertung von Daten genutzt wird.

---

[55] Vgl. Töpfer, Armin (2012), S. 233 ff.
[56] Vgl. Magerhans, Alexander (2016), S. 87 ff.
[57] Vgl. Berger-Grabner, Doris (2016), S. 129 f.
[58] Vgl. Weber, Daniela (2015), S. 62 ff.

3.2.1. Interviews

Interviews, auch Befragungen genannt, können abhängig vom Grad der Standardisierung in den verschiedenen Bereichen der Sozialforschung verwendet werden. Vollständig standardisierte Interviews werden bei der quantitativen, teil- und nicht-standardisierte Interviews werden bei der qualitativen Forschung verwendet. Allgemein dienen Interviews dem Zweck, Informationen von Personen zu erhalten, die vermeintlich für das Forschungsthema relevantes Wissen besitzen. Die befragte Person muss dabei nicht zwangsläufig der Untersuchungsgegenstand sein. Zudem können bei einer Gruppenerhebung auch Daten von mehreren Personen gleichzeitig erfasst werden. Quantitative bzw. voll-standardisierte Befragungen können sowohl mündlich als auch schriftlich durchgeführt werden, wohingegen qualitative, teil- oder nicht-standardisierte Befragungen in den meisten Fällen mündlich durchgeführt werden. Das liegt daran, dass bei schriftlichen Befragungen keine interaktive Kommunikation stattfinden und keine Nähe zum Untersuchungsgegenstand aufgebaut werden kann.[59] Ein Überblick über die in diesem Kapitel vorgestellten Interviewformen kann der folgenden Tabelle 1 entnommen werden:

Standardisierungsgrad		**Erhebungsmethode**
vollständig standardisiert	**mündlich**	Persönliche Befragung, Telefonische Befragung
	schriftlich	Schriftliche Befragung, Computergestützte Befragung
teil-standardisiert		Leitfadeninterview, Experteninterview
nicht-standardisiert		Narratives Interview, Ethnografisches Interview, Fokusgruppen

Tabelle 1: Formen von Interviews
Quelle: Vgl. Weber, Daniela (2015), S. 116

3.2.1.1. Vollständig standardisierte Interviews

Vollständig standardisierte Interviews können in mündliche und schriftliche Interviews unterteilt werden. Eine Form des mündlichen Interviews ist die *persönliche Befragung*. Bei dieser stehen sich Interviewer und Befragter persönlich gegenüber und füllen gemeinsam einen standardisierten Fragebogen aus. Vorteilhaft dabei ist, dass der Interviewer dem Probanden bei Unklarheiten weiterhelfen kann, wodurch Missverständnisse vermieden werden können. Hinzu kommt, dass der Interviewer die Reihenfolge der Fragen steuern kann, wodurch er mit Hilfe von Kontrollfragen widersprüchliche Antworten identifizieren kann. Beide Punkte resultieren in einer hohen Datenqualität. Die Anwesenheit des Interviewers bringt jedoch auch Nachteile mit sich, z.B. den sogenannten Interviewer-Bias. So können die Ergebnisse der Befragung verzerrt werden, weil der Interviewer unterbewusst den Befragten und seine Antworten

[59] Vgl. Weber, Daniela (2015), S. 112 ff.

beeinflussen kann. Hinzu kommt, dass der Befragte nicht anonym bleibt, wodurch dieser ggf. soziale Erwartungen erfüllen will und somit falsche Antworten gibt. Zuletzt sorgt die Anwesenheit des Interviewers noch dafür, dass eine Befragung einen hohen Aufwand bzw. hohe Kosten mit sich bringt.[60,61,62]

Eine weitere mündliche und spezielle Form der persönlichen Befragung ist die *telefonische Befragung*. Bei ihr werden die Fragen des Fragebogens über ein Telefonat gestellt und die Antworten anschließend notiert. Interviewer und Befragter stehen weiterhin in einer direkten Verbindung, haben aber keinen Blickkontakt. Eine telefonische Befragung ist gut für eine geografisch verteilte Zielgruppe geeignet, Voraussetzung ist allerdings, dass der Befragte einen Telefonanschluss besitzt. Durch die leichte Erreichbarkeit sinken der zeitliche Aufwand und die Kosten. Der Interviewer kann den Befragten weiterhin bei Unklarheiten helfen und durch Kontrollfragen widersprüchliche Antworten aufdecken. Im Vergleich zur persönlichen Befragung besitzt der Proband eine höhere Anonymität, wodurch sich der Interviewer-Bias nicht mehr so stark auswirkt. Nachteilhaft bei telefonischen Befragungen ist, dass die Probanden häufig eine geringe Bereitschaft zeigen, an der Umfrage teilzunehmen, wodurch die Repräsentanz der Ergebnisse entsprechend gering ist. Zudem sollten keine komplexen Fragen mit vielen Antwortmöglichkeiten gestellt werden, da sich der Befragte die über das Telefon übermittelten Optionen merken muss. Aus kognitiv-wissenschaftlicher Sicht werden dabei verstärkt die erste und letzte Option genannt, wodurch die Ergebnisse der Befragung verzerrt werden.[63,64,65]

Neben den mündlichen können auch *schriftliche Befragungen* durchgeführt werden. Bei diesen füllt der Proband den Fragebogen selbstständig aus. Dazu wird der Fragebogen entweder postalisch verschickt, sodass dieser zu Hause ausgefüllt werden kann, oder er wird an mehrere, in einem Raum versammelte Personen ausgeteilt und anschließend wieder eingesammelt. Wichtig ist, dass der Fragebogen transparent und verständlich gestaltet ist, weil der Interviewer hier nicht in die Befragung eingreift. Die Vorteile dabei sind, dass der Befragte anonym bleibt und keine Interviewer-Effekte auftreten. Da sich der Forscher während der Befragung anderen Tätigkeiten widmen kann, hält sich der Zeitaufwand in Grenzen. Zuletzt haben die Befragten die Freiheit, sich aussuchen zu können, wann sie den Fragebogen ausfüllen. Nach-

[60] Vgl. Berger-Grabner, Doris (2016), S. 162 f.
[61] Vgl. Kuß, Alfred u.a. (2018), S. 127 ff.
[62] Vgl. Raab, Gerhard u.a. (2018), S. 114 f.
[63] Vgl. Berger-Grabner, Doris (2016), S. 163 f.
[64] Vgl. Kuß, Alfred u.a. (2018), S. 131 ff.
[65] Vgl. Raab, Gerhard u.a. (2018), S. 118 f.

teilig bei einem postalischen Versand ist die geringe Rücklaufquote, also der Anteil der beantworteten und zurückgeschickten Fragebögen. Dadurch ist wieder nur eine geringe Repräsentanz gegeben. Bei schriftlichen Befragungen besitzt der Interviewer keine Kontrolle über den Verlauf. Es können keine Kontrollfragen gestellt werden, wodurch sich die Qualität der Daten verringert. Zudem kann nicht sichergestellt werden, ob der Fragebogen tatsächlich von der Zielperson oder einer anderen ausgefüllt wurde. Zuletzt könnte die Lesbarkeit von Handschriften auch zu einem Problem führen.[66,67,68]

Eine spezielle Form der schriftlichen Befragung ist die *computergestützte* bzw. *Online-Befragung*. Bei dieser wird ein häufig per E-Mail versendeter Fragebogen in digitaler Form an einem Computer ausgefüllt. Voraussetzung ist entsprechend, dass die befragte Person einen Computer oder ein mobiles Endgerät mit einer Internetverbindung besitzt. Auch hier zeigt der Interviewer-Bias keine Wirkung und die Probanden können sich aussuchen, wann sie den Fragebogen ausfüllen. Zwar ist der Interviewer auch bei dieser Form der Befragung nicht anwesend, komplexe Sachverhalten können jedoch durch visuelle und akustische Elemente veranschaulicht und die Reihenfolge der Fragen kann technisch gesteuert werden, was in einer hohen Datenqualität resultiert. Da die Verteilung über das Internet läuft, können in kurzer Zeit viele Probanden mit einem geringen Aufwand und Kosten befragt werden, wodurch die Stichprobe entsprechend repräsentativ ist. Zudem ist auch gleich eine automatische Auswertung möglich, da die Daten bereits in digitaler Form vorliegen. Nachteilhaft ist auch hier, dass keine Sicherheit darüber besteht, wer den Fragebogen ausfüllt. Hinzu kommt, dass falsch angegebene Informationen schwer nachvollziehbar sind. Zuletzt sind aufgrund der Voraussetzungen bestimmte Zielgruppen, wie z.B. ältere Personen, nur schlecht erreichbar.[69,70,71]

Ein Überblick über vollständig standardisierte Befragungen kann dem Anhang F entnommen werden.

3.2.1.2. Teil-Standardisierte Interviews

Die teil-standardisierten Interviews gehören zur qualitativen Sozialforschung. Das offene und flexible Vorgehen bei der Erhebung nicht-metrischer Daten durch das individuelle Eingehen auf die befragten Personen und die daraus folgende, geringere Stichprobengröße wurde bereits im Kapitel 3.1.3 Quantitative vs. qualitative Sozialforschung beschrieben. Qualitative

[66] Vgl. Berger-Grabner, Doris (2016), S. 164.
[67] Vgl. Kuß, Alfred u.a. (2018), S. 129 ff.
[68] Vgl. Raab, Gerhard u.a. (2018), S. 115 ff.
[69] Vgl. Berger-Grabner, Doris (2016), S. 165.
[70] Vgl. Kuß, Alfred u.a. (2018), S. 133 ff.
[71] Vgl. Raab, Gerhard u.a. (2018), S. 119 ff.

Interviews verwenden zwar im Vergleich zu den vollständig standardisierten Interviews keine streng strukturierten Fragebögen, durch Leitfäden werden jedoch grob die inhaltlichen Themen des Gespräches vorgegeben. Hierbei soll keine Befragung mit abwechselnder Frage und Antwort, sondern eine möglichst natürliche Gesprächssituation entstehen.[72]

Eine sehr geläufige Art des teil-standardisierten Interviews ist das *Leitfadeninterview*. Wie es der Namen bereits vermuten lässt, wird bei diesem Interview ein Leitfaden verwendet, der anhand einer Literaturrecherche und theoretischen Voranalyse des Forschungsgegenstandes erstellt wird. Der Leitfaden gibt dabei die Fragen vor, nicht jedoch die Reihenfolge der Beantwortung, da ansonsten kein natürliches Gespräch zustande kommen würde. Durch die daraus resultierende, leichte Strukturierung der Daten sind etwas größere Stichproben als bei den nicht-standardisierten Interviews möglich.[73] Eine spezielle Form des Leitfadeninterviews ist das *Experteninterview*. Bei diesem werden Experten als Repräsentanten einer bestimmten Gruppe interviewt. Der Expertenstatus ist dabei abhängig vom jeweiligen Forschungsgegenstand. So können z.B. Personen mit umfangreichen Wissen bzw. Erfahrungen in einem Gebiet oder mit Verantwortung zu einem Sachverhalt als Experten gelten. Das Ziel ist es, das Wissen der Experten möglichst umfassend zu rekonstruieren. Das Expertenwissen grenzt sich dabei vom Allgemeinwissen dahingehend ab, dass ersteres in einem langwierigen, aktiven Lernprozess erworben werden muss. Spezialisten verfügen über einen genau abgegrenzten Teil des Expertenwissens und können damit spezielle Funktionen erfüllen. Im Vergleich zu Spezialisten haben Experten einen Überblick über einen Wissensbereit und können somit den Ursachen von Problemen auf den Grund gehen. Zur Auswertung von Experteninterviews wird die später vorgestellte, qualitative Inhaltsanalyse empfohlen, deren Ergebnisse und Interpretationen über eine kommunikative Validierung mit dem Experten abgesichert werden sollen.[74]

3.2.1.3. Nicht-Standardisierte Interviews

Die nicht-standardisierten Interviews zeichnen sich in einer maximalen Offenheit und Flexibilität im Gespräch ohne Leitfaden aus. Der Vorteil dieser offenen Gesprächsführung ist, dass in einer intensiven Unterhaltung ein tiefes Verständnis über den Probanden und seiner Eindrücke aufgebaut werden kann. So wird in einem *narrativen Interview* eine in der Vergangenheit erlebte Situation besprochen, die für das Forschungsthema von Interesse ist. Das Ziel ist es, die Geschichte aus der Sicht des Befragten nachzuvollziehen und somit deren subjektive Bedeutungsstrukturen zu analysieren. Dazu muss nicht nur der Inhalt des Gespräches, sondern

[72] Vgl. Magerhans, Alexander (2016), S. 169 ff.
[73] Vgl. Berger-Grabner, Doris (2016), S. 140 ff.
[74] Vgl. Buber, Renate & Holzmüller, Hartmut H. (Hrsg.) (2009), S. 451 ff.

auch die emotionalen Regungen des Probanden in Mimik und Gestik aufgezeichnet werden. Aufgrund des ohnehin schon hohen, zeitlichen Aufwandes sind narrative Interviews ungeeignet, routinemäßige Handlungen und längere Zeitabschnitte zu besprechen.[75] Eine weitere, nicht-standardisierte Form ist das *ethnografische Interview*. Anders als beim narrativen interview dreht sich dieses nicht um ein einmaliges Ereignis, sondern um die Lebenswelt, den Alltag und die Werteorientierungen der befragten Person. So werden beispielsweise Aspekte wie die Familie, Freunde, die Arbeit oder das eigene Haus thematisiert.[76]

Bei der zuletzt vorgestellten Interviewform handelt es sich um die *Gruppendiskussion*, auch *Fokusgruppe* genannt. Bei dieser handelt es sich um eine ein- bis mehrstündige Diskussion, die von einem geschulten Moderator begleitet wird. Das Ziel der Fokusgruppe ist es, die Einstellungen, Meinungen, Gefühle, Wünsche oder Ideen der Diskussionsteilnehmer zu ermitteln. An einer Diskussion sollten durchschnittlich sechs bis zehn Personen teilnehmen, wobei die Gruppengröße kleiner sein sollte, je komplexer oder emotional geladener das Diskussionsthema ist. Die Vorteile einer Fokusgruppe liegen in den gruppendynamischen Prozessen, welche die Ängste und Hemmungen der Teilnehmer reduzieren. Zudem können spontane Reaktionen bei den Probanden beobachtet werden. Handelt es sich um eine heterogene Gruppe, können vielschichtige Ergebnisse entstehen, gleichzeitig kann es aber auch vorkommen, dass die Diskussion durch dominierende Teilnehmer sehr einseitig wird. Auch kann das Gespräch schnell vom eigentlichen Thema abweichen, wenn die Gruppe eine eigene Dynamik entwickelt. Selbst wenn die Diskussion gut verlaufen ist, benötigt der Forscher noch umfangreiche, methodische Kenntnisse und viel Erfahrung, um die Aufzeichnungen sachgerecht interpretieren und auswerten zu können. Zuletzt handelt es sich bei einer Gruppengröße von sechs bis zehn Personen um eine recht kleine Stichprobe, wodurch eine einzelne Gruppendiskussion häufig keine repräsentativen Ergebnisse liefert.[77,78,79]

3.2.2. Beobachtung

Bei Beobachtungen handelt es sich um die Erfassung, Dokumentation und Interpretation von sinnlich wahrnehmbaren Sachverhalten. Bei diesen Sachverhalten kann es sich z.B. um die Verhaltensweisen von Lebewesen und deren zeitliche Verknüpfung, um verbale Äußerungen und deren emotionale Konnotation oder um einen situativen Kontext handeln. Die Beobach-

[75] Vgl. Buber, Renate & Holzmüller, Hartmut H. (Hrsg.) (2009), S. 361 ff.
[76] Vgl. Berger-Grabner, Doris (2016), S. 137 ff.
[77] Vgl. Magerhans, Alexander (2016), S. 174 ff.
[78] Vgl. Berger-Grabner, Doris (2016), S. 143 f.
[79] Vgl. Buber, Renate & Holzmüller, Hartmut H. (Hrsg.) (2009), S. 479 ff.

tung findet dabei in dem Zeitpunkt statt, in welchem der Sachverhalt auftritt. Eine Kombination mit anderen quantitativen oder qualitativen Methoden ist möglich. Der Unterschied zwischen einer quantitativen und qualitativen Beobachtung liegt wieder in deren Grad der Standardisierung. Eine quantitative Beobachtung wird sehr systematisch durchgeführt. Dazu wird ein Beobachtungsschema oder ein Fragebogen erstellt, der während der Beobachtung ausgefüllt werden muss. Nur die zuvor im Fragebogen definierten Merkmale werden erfasst und ausgewertet. Eine qualitative Beobachtung ist im Vergleich wesentlich offener und flexibler. Zwar werden auch hier die zu beobachtenden Merkmale grob in einem Leitfaden festgelegt, es werden aber auch darüberhinausgehende Informationen erfasst und ausgewertet. Neben dem Fragebogen bzw. Leitfaden müssen zur Vorbereitung noch der Beobachter, die Beobachtungseinheit, der Beobachtungsgegenstand und das Beobachtungsfeld festgelegt werden. Bei der Beobachtungseinheit handelt es sich um den Gegenstand oder die Person, die beobachtet werden soll. Der Beobachtungsgegenstand ist die Eigenschaft bzw. das Merkmal an der Person. Das Beobachtungsfeld wird durch den Ort und Zeitpunkt der Beobachtung bestimmt.

Eine Beobachtung bringt im Vergleich zu einer Befragung einige Vorteile mit sich. Abgesehen davon, dass sie zu einigen Forschungsthemen die einzig mögliche Methode der Datenerhebung ist, tritt hier kein Interviewer-Effekt auf. Außerdem kann eine Beobachtung unabhängig von der Auskunftsfähigkeit oder -willen der Person durchgeführt werden. Nimmt der Forscher aktiv an der Beobachtung teil, kann der Beobachtungsgegenstand aus nächster Nähe betrachtet werden, gleichzeitig ist jedoch eine unmittelbare Aufzeichnung nur schwer möglich. Ist dem Proband bewusst, dass der beobachtet wird, zeigt dieser ggfs. kein natürlich Verhalten, wodurch die Daten verzerrt werden. Wird die Beobachtung ohne Wissen des Probanden durchgeführt, können ethische oder rechtliche Probleme auftreten. Ein weiterer, kritischer Punkt ist, dass eine beobachtete Situation einmalig ist, wodurch die Gütekriterien der Reliabilität und der Validität nicht mit Sicherheit gewährleistet werden können. Ein letzter, nachteiliger Punkt ist, dass die Datenerhebung über die Sinne des Beobachters stattfindet, wodurch es zwangsläufig zu einer selektiven Wahrnehmung kommt. Weil hierdurch einer Verzerrung der Daten auftritt, wird eine Ton- oder Videoaufnahme empfohlen, durch die verpasste Informationen trotzdem verarbeitet werden können.[80,81,82]

[80] Vgl. Berger-Grabner, Doris (2016), S. 151 ff.
[81] Vgl. Magerhans, Alexander (2016), S. 122 ff.
[82] Vgl. Buber, Renate & Holzmüller, Hartmut H. (Hrsg.) (2009), S. 527 ff.

3.2.3. Inhaltsanalyse

Die Inhaltsanalyse ist eine Methode, die sowohl zur Datenerhebung als auch zur Datenauswertung genutzt werden kann. So kann sie in der qualitativen Forschung zur Auswertung qualitativer Texte genutzt werden, darauf wird jedoch genauer im Kapitel 3.3.2 Qualitative Datenauswertung eingegangen. An dieser Stelle soll es jedoch um die quantitative Inhaltsanalyse gehen, die zur Ermittlung von Informationen aus Schriftstücken und anderen Artefakten dient. Informationsquellen können z.B. Zeitungsartikel, Transkriptionen von Interviews etc. sein. Ziel ist es dabei, durch ein Zählen oder Messen quantitative Daten aus dem Text zu gewinnen. So können in einer Frequenzanalyse die Anzahl bestimmter Wörter, in einer Kontingenzanalyse die Verhältnisse bzw. die Verknüpfungen zwischen verschiedenen Wörtern oder in einer Valenzanalyse die Konnotation der verwendeten Wörter ermittelt werden. Mit diesen Daten können dann die Intentionen des Verfassers ermittelt, die Zielgruppe des Textes analysiert, der Text auf bestimmte Formulierungen hin untersucht und relevante, soziale Aspekte enthüllt werden.[83]

3.3.Methoden zur empirischen Datenauswertung

Nachdem die für die Forschung relevanten Daten erhoben wurden, müssen diese noch ausgewertet werden. Da sich die quantitative und qualitative Datenauswertung aufgrund der Art der Daten grundlegend unterscheiden, werden sie folgend in zwei separaten Kapiteln erläutert.

3.3.1. Quantitative Datenauswertung

Eine quantitative Datenauswertung findet anhand objektiver, statistischer Verfahren statt. Es wird grundlegend die deskriptive und induktive Statistik unterschieden. Bei der deskriptiven Statistik werden, wie es der Name schon vermuten lässt, die quantitativen Daten beschrieben. Es geht also darum, die metrischen Daten möglichst übersichtlich darzustellen, sodass etwaige Zusammenhänge ersichtlich werden. Bei der induktiven Statistik, auch Inferenzstatistik genannt, werden Schlussfolgerungen aus den Zahlen abgeleitet. Hierbei wird geprüft, ob die zuvor aufgestellten Hypothesen bestätigt werden können. Die Durchführbarkeit der statistischen Verfahren ist vom Messniveau der Daten abhängig. Je höher das Messniveau der Daten ist, umso höher ist der Informationsgehalt und umso mehr Auswertungen können vorgenommen werden. Abhängig von der Anzahl der gleichzeitig untersuchten Variablen können univariate (eine Variable), bivariate (zwei Variablen) oder multivariate (drei oder mehr Variablen)

[83] Vgl. Weber, Daniela (2015), S. 177 ff.

Verfahren unterschieden werden.[84,85] Letztere werden hier nicht vorgestellt, weil sie für die Untersuchungen in dieser Arbeit irrelevant sind. Zudem werden auch Berechnungsformeln nicht aufgeführt, weil diese für das Verständnis des Ergebnisses nicht nötig sind.

3.3.1.1. Univariate Verfahren

Bei univariaten Verfahren kann die einzelne, betrachtete Variable in einer Häufigkeitsverteilung dargestellt werden. Die Häufigkeiten in der Verteilung können absolut und relativ angegeben werden. Bei der *absoluten Häufigkeit* handelt es sich um die Anzahl, die eine jeweilige Ausprägung während der Erhebung erfasst wurde. Die *relative Häufigkeit* hingegen ist eine prozentuale Angabe der absoluten Häufigkeit bezogen auf die Gesamtzahl. Durch ihren rein deskriptiven Charakter beschreiben die univariaten Verfahren die quantitativen Daten nur. Die Beschreibung erfolgt dabei anhand von Lage- und Streuungsparametern.

Die *Lageparameter* beschreiben den Schwerpunkt einer Häufigkeitsverteilung. Zu allen Messniveaus kann der *Modalwert*, oder auch *Modus*, ermittelt werden. Dieser ist der häufigste Wert der Verteilung, also die Merkmalsausprägung mit der größten, absoluten Häufigkeit. Wurden die Daten auf einer Ordinalskala erfasst, so lässt sich durch die Rangfolge ein sinnvoller *Median* ermitteln. Dieser ist der Wert in der Mitte, der die Häufigkeitsverteilung in zwei gleichgroße Hälften teilt. Bei metrischen Daten kann zudem das *arithmetische Mittel* errechnet werden, das auch *Mittelwert* genannt wird. Zuletzt kann bei verhältnisskalierten Werten das *geometrische Mittel* aus der n-ten Wurzel des Produktes aller Ausprägungen errechnet werden, wobei n die Anzahl der Ausprägungen ist. Dieses Mittel wird zur weiteren Rechnung mit Verhältniszahlen genutzt.

Streuungsparameter hingegen beschreiben die Variabilität der Messwerte. Diese können nur bei metrischen Daten errechnet werden. Die *Spannweite* gibt die Differenz zwischen den größten und kleinsten Wert an. Die *Varianz* ist ein weiterer Parameter, der sich aus der mittleren, quadrierten Abweichung der Messwerte vom arithmetischen Mittel berechnet. Die Varianz selbst besitzt keine direkte Bedeutung, aus ihrer positiven Wurzel lässt sich jedoch die *Standardabweichung* ermitteln. Durch das Ziehen der Wurzel wird der Wert wieder auf die Einheit der Ausgangsdaten zurückgeführt, wodurch er die durchschnittliche Entfernung aller Werte vom Mittelwert angibt. Je größer die Standardabweichung ist, desto höher ist also die Streuung der Werte einer Variablen. Der *Variationskoeffizient* ist zuletzt eine prozentuale Verhältniszahl, die sich aus dem Quotienten von Standardabweichung und Mittelwert ergibt.

[84] Vgl. Berger-Grabner, Doris (2016), S. 171.
[85] Vgl. Magerhans, Alexander (2016), S. 134.

Der Variationskoeffizient kann dazu genutzt werden, um die Streuungen mehrerer Merkmale zu vergleichen.[86,87,88]

Ein Überblick über die Anwendbarkeit der univariaten Verfahren abhängig vom Messniveau der Daten kann der folgenden Tabelle 2 entnommen werden:

	nicht metrisch		**metrische**	
	Nominal	**Ordinal**	**Intervall**	**Verhältnis**
Lageparameter	Modus	Median	arithmetisches Mittel	geometrisches Mittel
Streuungsparameter			Spannweite, Varianz, Standardabweichung	Variationskoeffizient

Tabelle 2: Univariate Verfahren nach Messniveau
Quelle: Vgl. Töpfer, Armin (2012), S. 261 & Kuß, Alfred u.a. (2018), S. 242

3.3.1.2. Bivariate Verfahren

Im Vergleich zu univariaten Verfahren untersuchen die bivariaten den Zusammenhang zwischen zwei Variablen. Eine übersichtliche Darstellung der beiden Variablen kann in Form einer Kreuztabelle erfolgen. Bei dieser handelt es sich um eine zweidimensionale Matrix, die die absoluten und relativen Häufigkeiten getrennt anhand der zwei Merkmale aufführt.

Eine bivariate Auswertung, die mit Daten aller Messniveaus durchgeführt werden kann, ist die *Kontingenzanalyse*. Bei dieser wird ermittelt, ob ein Zusammenhang zwischen zwei Merkmalen besteht, nicht aber die Stärke oder die Richtung des Zusammenhangs. Dazu wird ein Kontingenzkoeffizient errechnet, der Werte von 0,0 bis 1,0 annehmen kann. Je größer der Koeffizient ist, umso eher besteht ein Zusammenhang zwischen beiden Variablen.[89]

Ein weiteres, bivariates Verfahren, das auf ordinalskalierten Daten angewandt werden kann, ist die *Korrelationsanalyse*. Mit ihr wird die Stärke bzw. die Wahrscheinlichkeit eines Zusammenhangs von zwei Variablen geprüft. Dazu wird der Korrelationskoeffizient berechnet, der als Maßzahl für diesen Zusammenhang gilt. Dieser Koeffizient kann Werte von -1,0 bis +1,0 annehmen, wobei der Zusammenhang stärker ist, umso größer der Betrag des Wertes ist. Durch das Vorzeichen wird das Wirkungsverhalten zwischen den beiden Variablen bestimmt. Ist der Korrelationskoeffizient negativ, verhalten sich beide Variablen gegenläufig. Zur Berechnung des Korrelationskoeffizienten nach Pearson müssen die Daten metrisch skaliert sein. Für ordinalskalierte Werte kann die Rangkorrelation nach Spearman genutzt werden. Dabei werden zu den Werten einfach Ränge gebildet, mit denen dann die metrischen Berechnungen

[86] Vgl. Töpfer, Armin (2012), S. 261 ff.
[87] Vgl. Berger-Grabner, Doris (2016), S. 171 ff.
[88] Vgl. Magerhans, Alexander (2016), S. 134 ff.
[89] Vgl. Töpfer, Armin (2012), S. 264 ff.

durchgeführt werden. Bei nominalskalierten Daten ist jedoch auch das nicht möglich. Bei nominalskalierten Daten sind jedoch auch die Ansätze von Pearson und Spearman nicht anwendbar. An dieser Stelle muss über eine andere Formel der punktbiseriale bzw. punktpolyseriale Korrelationskoeffizient berechnet werden. [90,91]

Das zuletzt vorgestellte, bivariate Verfahren ist die *Regressionsanalyse*. Bei dieser wird der Zusammenhang zwischen zwei metrischen Variablen mit eindeutiger Wirkungsrichtung geprüft. Die Wirkungsrichtung ist dadurch gegeben, dass eine abhängige und eine unabhängige Variable untersucht werden. Ziel der Regressionsanalyse ist es, eine sogenannte Regressionsgerade aufzustellen. Diese entspricht dem zweidimensionalen Mittelwert beider Variablen, wobei die Abstände zwischen den Ausprägungen und der Gerade minimiert werden sollen. Die Regressionsgerade wird mathematisch als lineare Funktion abgebildet, wobei der Anstieg dieser Funktion durch den Regressionskoeffizienten bestimmt wird. Über diese Funktion kann dann der Wert der abhängigen Variablen über den Wert der unabhängigen prognostiziert werden.[92]

Ein Überblick über die Anwendbarkeit der bivariaten Verfahren abhängig vom Messniveau der Daten kann der folgenden Tabelle 3 entnommen werden:

nicht-metrisch		**metrisch**	
Nominal	**Ordinal**	**Intervall**	**Verhältnis**
Kontingenzanalyse punktbiseriale Korrelation	Korrelationsanalyse	Regressionsanalyse	

Tabelle 3: Bivariate Verfahren nach Messniveau
Quelle: Vgl. Töpfer, Armin (2012), S. 264

3.3.2. Qualitative Datenauswertung

Im Vergleich zur quantitativen Datenauswertung nutzt die qualitative keine objektiven, statistischen sondern offene und flexible Verfahren. *Offenheit* ist in dem Rahmen so zu verstehen, dass der Forscher offen gegenüber dem Unterwarteten oder nicht unmittelbar erklärbaren Ergebnissen ist. Dazu ist eine unvoreingenommene Herangehensweise von Nöten, um alle relevanten Aspekte des untersuchten Sachverhaltes identifizieren zu können. Um diese Offenheit zu gewährleisten, muss eine stetige *Selbstreflexion* vorgenommen werden. Da die Auswertung bereits mit der Datenerhebung beginnt und diese zudem vom Forscher beeinflusst wird, müssen sowohl der Befragte als auch der Fragende ihre eigenen Aussagen reflektieren und deren

[90] Vgl. Berger-Grabner, Doris (2016), S. 174 ff.
[91] Vgl. Dormann, Carsten (2017), S. 104.
[92] Vgl. Magerhans, Alexander (2016), S. 139 ff.

Einfluss und Wirkung kritisch hinterfragen. Eine Absicherung der Interpretationen im Sinne der kommunikativen Validierung trägt dabei zur Offenheit und Reflexivität bei. Zuletzt muss bei der Auswertung qualitativer Daten immer eine *Kontextualisierung* stattfinden, da die Daten abhängig vom Kontext eine neue Bedeutung erhalten können und somit andere Interpretationen möglich sind.[93]

Ein sehr häufig genutztes Verfahren zur qualitativen Datenauswertung ist die *qualitative Inhaltsanalyse*. Die quantitative Inhaltsanalyse zur Erhebung quantitativer Daten wurde bereits im Kapitel 3.2.3 Inhaltsanalyse vorgestellt, bei der qualitativen Inhaltsanalyse werden jedoch Daten in Form von Texten, Audio oder Video ausgewertet. Häufig handelt es sich bei diesen Daten um Transkriptionen von qualitativen Interviews. Das Ziel ist es, das Datenmaterial zu reduzieren und im Hinblick auf die Beantwortung der Forschungsfrage zu untersuchen. Neben dem oben bereits erwähnten Kontext müssen noch weitere Aspekte bei der Analyse beachtet werden. Dazu zählen u.a. die latenten Sinnstrukturen von Wörtern, deren Bedeutung nicht immer objektiv bestimmt werden kann. Ein weiterer Punkt der qualitativen Inhaltsanalyse ist, dass die Relevanz einer Aussage nicht mit ihrer Häufigkeit zusammenhängt. Ein Einzelfall, der bei einer quantitativen Betrachtung als Ausreißer gesehen wird, kann aus qualitativer Sicht sehr wichtig für das Forschungsthema sein. Zuletzt beinhaltet nicht nur die Präsenz, sondern auch die Abwesenheit von Informationen eine Bedeutung. So kann z.B. ein offensichtliches Thema, das vom Befragten bewusst nicht abgesprochen wird, peinlich für diesen sein. Unter Beachtung der Aspekte können bei einer qualitativen Inhaltsanalyse drei Tätigkeiten durchgeführt werden: eine Zusammenfassung, eine Explikation und eine Strukturierung. Bei einer Zusammenfassung wird das Textmaterial mit dem Ziel einer besseren Überschaubarkeit verdichtet, wobei der wesentliche Inhalt erhalten bleiben soll. Bei einer Explikation werden ggfs. erklärungsbedürftige Textstellen mit zusätzlichen Informationen angereichert, sodass diese für Außenstehende verständlicher werden. Zuletzt können die Daten während einer Strukturierung nach bestimmten Ordnungskriterien gruppiert werden. Das Ziel hierbei ist es, ein Kategoriensystem aufzubauen, indem verschiedene Merkmale aus dem Text identifiziert und ihnen Ausprägungen anhand vorgegebener Codier-Regeln zugewiesen werden. Auf diese Weise werden nicht-metrische Daten aus dem Text gewonnen, die dann weiter ausgewertet werden können.[94,95,96]

[93] Vgl. Naderer, Gabriele & Balzer, Eva (Hrsg.) (2011), S. 407 ff.
[94] Vgl. Berger-Grabner, Doris (2016), S. 144 ff.
[95] Vgl. Naderer, Gabriele & Balzer, Eva (Hrsg.) (2011), S. 418 ff.
[96] Vgl. Weber, Daniela (2015), S. 417 ff.

Neben der qualitativen Inhaltsanalyse existieren noch viele weitere Methoden, die hier jedoch nur kurz vorgestellt werden, weil sie im Folgenden nicht weiter relevant sind. So kann bei einer *computergestützten Analyse* die Datenauswertungen anhand sogenannter „Qualitative Data Analysis Tools" vereinfacht werden. Diese Tools unterstützen dabei lediglich die Analyse und können sie nicht vollautomatisiert vornehmen. Voraussetzung ist, dass die qualitativen Daten in digitaler Form vorliegen. Konkret werden Funktionen zum Management und zur Exploration von Daten, zum Text Mining, zur Bildung eines Kategoriensystems, zur Visualisierung der Daten anhand verschiedener Diagramm- und Strukturdarstellungen und zur Themenanalyse durch Zusammenstellung kategorisierter Textstellen angeboten.[97,98]

Mit der *narrativen Analyse* werden die bereits vorgestellten narrativen Interviews untersucht. Hierbei kann deduktiv oder induktiv vorgegangen werden. Eine deduktive, narrative Analyse setzt Prinzipien bzw. Regeln voraus, nach denen die Bedeutungen und Sinnstrukturen der Erzählung erklärt werden sollen. Eine induktive, narrative Analyse hingegen versucht den Effekt der Erzählung anhand der Identifikation derer kontextabhängigen Einheiten zu rekonstruieren.[99] Bei der narrativen Analyse wird bereits gefordert, dass der Forscher bei seiner Analyse gewisse Blickwinkel beachtet, die *hermeneutische Analyse* geht dabei noch einen Schritt weiter. Unter Hermeneutik wird eine Kunstlehre verstanden, die einen radikalen Perspektivenwechsel einfordert, um sich in der Analyse neue Interpretationswege zu eröffnen. Voraussetzung für den hermeneutischen Ansatz ist das Ablegen des gesamten Vorwissens und jeglicher Vorurteile, sodass ein möglichst unbefangener Eindruck erlangt und die stereotypischen Interpretationen kritisch hinterfragt werden können.[100]

Zuletzt können mittels *Konversations- und Diskursanalysen* noch Gruppendiskussionen ausgewertet werden. Bei ersterer werden unter Berücksichtigung von Inhalt, Sequenz und Kontext des Gespräches die zwischenmenschlichen Interaktionen untersucht. Konkret interessieren neben den Gefühlen und Intentionen der Teilnehmer auch die Ordnungsprinzipien einer Konversation und deren Funktionen. Das Ziel ist das Finden von Lösungen für Probleme auf sozialer Ebene. Dazu findet einer Analyse und Interpretation der Konflikte statt, die der Interaktion in der Gruppe zugrunde liegen.[101,102] Bei der *Diskursanalyse* wird nicht der Stil, sondern der Inhalt einer Unterhaltung untersucht. Mit Diskurs ist also der thematische Verlauf

[97] Vgl. Buber, Renate & Holzmüller, Hartmut H. (Hrsg.) (2009), S. 715 ff.
[98] Vgl. Naderer, Gabriele & Balzer, Eva (Hrsg.) (2011), S. 430 f.
[99] Vgl. Buber, Renate & Holzmüller, Hartmut H. (Hrsg.) (2009), S. 367 ff.
[100] Vgl. Naderer, Gabriele & Balzer, Eva (Hrsg.) (2011), S. 422 ff.
[101] Vgl. ebenda, S. 424 f.
[102] Vgl. Weber, Daniela (2015), S. 406 ff.

einer Konversation gemeint, der in einer Diskursanalyse rekonstruiert und interpretiert werden soll. Daraus ergeben sich Erkenntnisse zu den Strukturen, nach denen Meinungsbildung abläuft. Werden mehrere Diskurse verglichen, können zudem noch Erkenntnisse über die Handhabung desselben Themas von verschiedenen Gruppen entstehen.[103]

3.4.Peer Reviews

Zuletzt soll im Rahmen des Kapitels zu den Evaluationsmethodiken noch das Peer Review vorgestellt werden. Dieses Verfahren wird nicht direkt der empirischen Sozialforschung zugeordnet, da bei diesem weder Daten erhoben noch in einem statistischen oder qualitativen Sinne analysiert werden. Das Peer Review dient jedoch trotzdem der Evaluation, und zwar der von wissenschaftlichen Arbeiten.

Peer Review ist ein Begriff aus der englischen Sprache. *Peer* bezeichnet dabei eine Person, die in irgendeiner Form gleich bzw. gleichgestellt ist. Diese Gleichheit kann sich auf verschiedene Merkmale beziehen, wie z.B. das Alter, die Fähigkeiten oder den sozialen Status. Ein *Review*, auch Inspektion genannt, bezeichnet im Allgemeinen eine Sichtung, Prüfung und Bewertung eines Sachverhaltes oder eines Verfahrens. Das Ziel des Reviews ist es, dass die durchführende Person einen Überblick erlangt und Verbesserungen vornehmen kann. Nach der Kombination beider Definitionen ist ein Peer Review also eine Prüfung und Bewertung eines Sachverhaltes durch einen Gleichgestellten, wobei der Gleichgestellte im professionellen bzw. wissenschaftlichen Kontext ein Kollege vom selben Fach ist. Zwar gehören Peer Reviews wie eingangs erwähnt nicht direkt zur empirischen Sozialforschung, sie können jedoch in der Evaluationsforschung als externes Verfahren zur Fremdevaluation verwendet werden. In diesem Rahmen werden die Peer Reviews auch „Peerevaluation“ genannt. Das Ziel ist es, mit Hilfe der Beratung durch einen Experten die Qualität des zu prüfenden Sachverhaltes zu verbessern.[104]

Peer Reviews werden jedoch nicht ausschließlich zur Evaluationsforschung, sondern primär zur Erkennung von gut durchdachter und gut durchgeführter Forschung genutzt. In der Industrie gilt dieses Verfahren als Standard zur Sicherstellung der wissenschaftlichen Grundprinzipien in der Forschung, zur Prüfung korrekt genutzter Methodiken und Dokumentationen und zur Verifizierung einer validen Interpretation auf Basis einer soliden, empirischen Datengrundlage. Da das Peer Review ein sozialer Vorgang unter Wissenschaftlern ist, ist dieses unvermeidbar subjektiv. Zwar besitzt die Subjektivität die Schwäche, dass die Forschungser-

[103] Vgl. Naderer, Gabriele & Balzer, Eva (Hrsg.) (2011), S. 425 ff.
[104] Vgl. Gutknecht-Gmeiner, Maria (2008), S. 37 ff.

gebnisse entsprechend nicht objektiv bewertet werden, ohne sie wären jedoch ein innovatives Potenzial und weitere wissenschaftliche Aspekte nur schwer erfassbar. Die Grenzen des Verfahrens liegen darin, dass ein Peer Review nur auf Basis der vorliegenden Informationen erfolgen kann. Betrug und Fehlverhalten sind dadurch quasi nicht nachweisbar.[105]

Abhängig von den Rahmenbedingungen der Durchführung können verschiedene Variationen des Peer Reviews unterschieden werden. Bei einer *einfachen, blinden Begutachtung* kennt der Gutachter den Autor der Arbeit, bleibt jedoch selbst anonym. Der Vorteil darin liegt, dass der Gutachter seine Kommentare frei und ohne Einschränkungen äußern kann, gleichzeitig kann ihn die Anonymität dazu verleiten, unfaire oder abfällige Bemerkungen zu machen. Zudem kann der Gutachter aufgrund des Geschlechtes, der Hierarchie oder der Herkunft des Autors voreingenommen sein. Diese negativen Aspekte sollen durch eine *doppelte, blinde Begutachtung* beseitigt werden. Bei dieser Form bleiben Autor und Gutachter einander unbekannt, wodurch der Gutachter weiterhin freie Kommentare geben kann, diese jedoch gleichzeitig durch die Anonymität des Autors unvoreingenommen bleiben. Die Fairness gegenüber dem unbekannten Autor sorgt dabei für eine bessere Qualität des Reviews. Problematisch bei dieser Begutachtung ist allerdings, dass die Anonymität des Autors nur schwer sichergestellt werden kann, da sich dieser z.B. durch Spezialisten in Expertenkreisen einfach identifizieren lässt. Die *offene Begutachtung* versucht hingegen durch ein transparentes Verfahren, bei dem Autor und Gutachter einander bekannt sind, Vertraulichkeit und Fairness zu schaffen. Durch die Transparenz und Bekanntheit der Beteiligten sollen unfaire und nicht-informative Begutachtungen verhindert werden. Dazu wird teilweise auch das Gutachten mit der ursprünglichen Arbeit veröffentlicht, sodass der intellektuelle Beitrag des Gutachters ersichtlich wird und ebenfalls kritisch hinterfragt werden kann.[106]

[105] Vgl. Starck, Matthias J. (2018), S. 11 ff.
[106] Vgl. ebenda, S. 43 ff.

4. Funktionen und Grenzen von Progressive Web Apps

Nachdem PWAs in der Theorie und für die folgenden Schritte potenziell nutzbare Evaluationsmethodiken vorgestellt wurden, sollen nun die Funktionen und somit auch die Grenzen von PWAs untersucht werden. Aus diesen Erkenntnissen wird dann ein Regelwerk erstellt, welches zuletzt einer empirischen Evaluation unterzogen werden soll. Die Funktionen von PWAs können aus den Funktionen herkömmlicher Web Apps abgeleitet werden, ihre Grenzen ergeben sich jedoch erst, wenn bekannt ist, welche Funktionalitäten nativer Apps nicht durch Web Apps umsetzen werden können. Die im Kapitel 2.1.3 vorgestellten hybriden Apps werden hierbei nicht untersucht, weil sich deren Funktionen aus denen der nativen und Web Apps ergeben.

4.1. Hardware-Komponenten von Endgeräten

Es werden zunächst die Funktionen nativer Apps betrachtet. Wie den Architekturmodellen aus Kapitel 2.1 Typen von Apps (Vgl. z.B. Abbildung 1: Architektur von Nativen Apps) entnommen werden kann, bauen das Betriebssystem und alle Anwendungen auf die Hardware des Endgerätes auf. Alle Funktionen von nativen Apps, Web Apps und PWAs sind also von der vorhandenen Hardware abhängig.

Die Hardware-Komponenten unterscheiden sich zwar je Endgerät teilweise sehr stark, werden die Geräte jedoch in Desktop-Computer und mobile Geräte unterteilt, besitzen die Endgeräte je Gruppe eine recht ähnliche Ausstattung. *Desktop-PCs* nutzen zur Eingabe von Informationen eine Maus und eine Tastatur. Eine Aufnahme von Video oder Audio kann durch externe Geräte, wie z.B. Webcam oder Mikrofon, ermöglicht werden. Zur Ausgabe werden ein Monitor und Lautsprecher verwendet. Für die Verarbeitung der Daten werden Komponenten wie z.B. die Festplatte, CPU, RAM oder Grafikkarte benötigt. Zuletzt bietet ein Desktop-Computer mit einem CD- bzw. DVD-Laufwerk, USB-Ports oder der Netzwerkkarte verschiedene Schnittstellen zu anderen Geräten. Ein Netzteil versorgt die Komponenten mit Strom.[107]

Die Hardware *mobiler Endgeräte* unterscheidet sich von Desktop-PCs soweit, dass sie auch tatsächlich mobil genutzt werden können. Grundlegend wurde die Hardware dabei so kompakt und leicht wie möglich gehalten. Maus und Tastatur zur Eingabe wurden durch einen Touchscreen ersetzt. Mikrofon und Kamera, die bei Desktop-PCs noch externe Geräte waren, sind in den meisten mobilen Geräten standardmäßig integriert. Der Touchscreen wird zusätzlich neben den Lautsprechern auch zur Ausgabe von Informationen verwendet. Neben der

[107] Vgl. Egewardt, Rainer (2002), S. 16 ff.

visuellen und auditiven Ausgabe besitzen mobile Geräte häufig auch einen Vibrationsaktor, über den ein haptisches Feedback gegeben werden kann. Die Komponenten zur Verarbeitung von Informationen unterscheiden sich grundlegend nicht von denen der Desktop-PCs, sie sind jedoch auf Grund ihrer Größe prinzipiell etwas leistungsschwächer. Der einzige Unterschied ist, dass mobile Geräte anstelle des Netzteils einen Akku zur Gewährleistung der Mobilität besitzen. Neben dem oben bereits erwähnten USB-Port verfügen mobile Geräte häufig über viele weitere Schnittstellen, wie z.B. eine Antenne zur Kommunikation im Mobilfunknetz inkl. Karte für das Subscriber Identity Module (SIM), eine GPS-Antenne oder Chips für WLAN, Bluetooth oder Near Field Communication (NFC). Zuletzt besitzen viele mobile Endgeräte weitere Sensoren zur Ermittlung von Informationen aus deren Umfeld. Diese Sensoren werden im folgenden Kapitel 4.2.1 Hardwarebasierte Funktionen näher erläutert.[108]

4.2. Funktionen von nativen Apps

Auf den Hardware-Komponenten bauen nun die Funktion der nativen Apps auf. Die Untersuchung dieser beschränkt sich, wie bereits im Kapitel 2.1.1 Native Apps erwähnt, auf die Plattformen Android und iOS. Die für die Untersuchung verwendeten Quellen werden auf Grund ihrer Menge nicht im Literaturverzeichnis, sondern in einem separaten Dokument aufgeführt.[109]

4.2.1. Hardwarebasierte Funktionen

Während der Recherche hat sich herausgestellt, dass sich die Funktionen nativer Apps grob in hardware- und softwarebasierte Funktionen unterteilen lassen. Mit hardwarebasierten Funktionen sind solche gemeint, die im direkten Zusammenhang mit der Hardware des Endgerätes stehen. Da sich die vielen Gerätetypen in ihrer Ausstattung unterscheiden, ist das Vorhandensein der benötigten Hardware-Komponenten immer Voraussetzung für die Nutzung der jeweiligen Funktion.

Die erste Gruppe umfasst Funktionen für die *Aufnahme von Medien* wie Bilder, Audio und Video. Sowohl in Android als auch in iOS kann die Aufnahme von Bildern oder Videos durch Aufruf der vorinstallierten Kamera-App durchgeführt werden. Der Nutzer muss hierbei selbstständig Einstellungen vornehmen und das Bild bzw. Video aufnehmen. Eine vorinstallierte App zur Audioaufnahme existiert in beiden Systemen nicht. Ein weiterer Weg besteht jedoch darin, aus der nativen App heraus direkt auf die Kamera oder das Mikrofon zuzugreifen. Dadurch können programmatisch grundlegende Einstellungen zu Dateiformat, Blitzlicht,

[108] Vgl. Dembowski, Klaus (2017), S. 3 ff. & S. 116 ff.
[109] Siehe Anhang G, Quellenverzeichnis zum PWA Regelwerk.

Zoom und Fokus, aber auch fortgeschrittene Konfigurationen wie Filtereffekte oder Bildstabilisation vorgenommen werden. Grundsätzlich können Kamera und Mikrofon auf beiden Plattformen in ihrer Gänze kontrolliert werden.

Neben der Aufnahme von Medien ist auch deren *Darstellung* bzw. *Wiedergabe* möglich. Die Darstellung von Rastergrafiken, z.B. in den Formaten PNG oder JPEG, ist über verschiedene Oberflächenelemente in beiden Systemen möglich. Während die Darstellung von Vektorgrafiken im SVG-Format in Android problemlos möglich ist, müssen diese in iOS über einen Workaround durch Nutzung einer WebView angezeigt werden. Ein fortgeschrittenes Rendering ist wieder in beiden Systemen über die „OpenGL ES API“ möglich. Für die Wiedergabe von Audio und Video besitzen beide Plattformen Player-Klassen, über die die Wiedergabe gesteuert und bspw. Lautstärke, Panoramaregelung, Abspieltempo oder die Wiedergabe in einer Schleife eingestellt werden können. Zuletzt können über den Vibrationsaktor entsprechend Vibrationen im Gerät erzeugt werden. Während iOS lediglich die Erzeugung einer einfachen Vibration ermöglicht, können bei Android die Länge und Stärke der Vibration konfiguriert und Vibrationsmuster definiert werden.

Eine weitere Gruppe von Funktionen betrifft die *Persistenz* von Daten. So kann z.B. in Android und iOS vollständig auf das Dateisystem des Gerätes zugegriffen werden, es können also Dateien gelesen und geschrieben werden. Da dies eine recht simple Form der Persistenz ist, wird sie primär für statische Ressourcen verwendet. Dynamische Daten können hingegen auf beiden Plattformen in Form von Schlüssel-Wert-Paaren gesichert werden, um z.B. Nutzerpräferenzen zu speichern. Für komplexe Daten wird eine strukturierte Datenbank angeboten. iOS bietet sogar das Ablegen von sensitiven Daten, wie z.B. Passwörtern, in einer verschlüsselten Keychain an.

Aufgrund des mobilen Kontexts dreht sich eine Vielzahl der hardwarebasierten Funktionen um die *Konnektivität* und *Kommunikation* mit anderen Geräten. Zu den mobilfunkbasierten Funktionen zählen die Telefonie und das Schreiben von SMS. Für beide wird entsprechend eine Mobilfunkantenne und eine SIM-Karte benötigt. Sowohl bei Android als auch bei iOS kann ein Anruf über die vorinstallierte Telefon-App vorgenommen werden. Dabei wird die Telefon-App mit der vorausgefüllten Nummer geöffnet und der Nutzer muss den Anruf nur noch bestätigen. Android bietet darüber hinaus an, ein Telefonat in einer App detailliert über eine Schnittstelle zu steuern. Das Schreiben von SMS verläuft analog. Beide Plattformen bieten an, eine vorausgefüllte SMS in der vorinstallierten Nachrichten-App zu öffnen und Android ermöglicht zusätzlich das Versenden einer SMS im Hintergrund über eine API.

Eine *Kommunikation mit dem Internet* erfolgt entweder mobilfunkbasiert über die Mobilfunkantenne oder über den WLAN-Chip mit einem lokalen WLAN-Netzwerk. Android bietet hierzu an, den Status der Netzwerkverbindung abzufragen und zu ermittelt, ob eine Verbindung besteht und wenn ja, welcher Verbindungstyp vorliegt. Bei iOS scheint die Ermittlung des Netzwerkstatus nicht bzw. nur umständlich möglich zu sein. Unabhängig davon können auf beiden Systemen Informationen bei vorhandener Verbindung mittels HTTP-Anfragen über das Internet verschickt werden.

Neben Mobilfunk und Internet sind jedoch noch *anderen Kommunikationskanäle* vorhanden. So ermöglicht die GPS-Antenne auf beiden Plattformen eine einmalige oder kontinuierliche Ermittlung der aktuellen Position des Gerätes über das GPS. Zudem kann über Bluetooth eine Verbindung zu einem Gerät aufgebaut werden, um z.B. Musik kabellos an Kopfhörer zu übertragen. Mittels NFC-Chip können iOS-Apps sehr nahe NFC-Tags auslesen und schreiben. Android kann darüber hinaus eine direkte Verbindung zu anderen NFC-Geräten aufbauen und sich sogar selbst als NFC-Tag ausgeben. Zuletzt können beide Systeme mit den über USB angeschlossenen Geräten Daten austauschen.

Bei der letzten Gruppe hardwarebasierter Funktionen handelt es sich um die *Sensoren* in mobilen Endgeräten. Android erlaubt dabei einen vollen Zugriff auf alle vorhandenen Sensoren an. So können über die beiden Bewegungssensoren Accelerometer und Gyroscope die Beschleunigung des Gerätes in m/s^2 und dessen Rotation in rad/s ermittelt werden. Anhand der vorhandenen Umgebungssensoren kann die Umgebungstemperatur in °C, das Umgebungslicht in lx (Lux), das geomagnetische Feld in µT (Mikrotesla), die Nähe zu einem Objekt in cm, der Luftdruck in hPa (Hektopascal) oder mbar (Millibar) und zuletzt die Luftfeuchtigkeit in % gemessen werden. Der Zugriff auf Sensoren in iOS ist etwas eingeschränkt. So sind Sensoren für Temperatur und Luftfeuchtigkeit bisher nicht in Apple-Geräten vorhanden. Ein Lichtsensor ist zwar eingebaut, es kann jedoch nicht programmatisch darauf zugegriffen werden und zuletzt kann anhand des Näherungssensors nur ermittelt werden, ob ein Objekt dicht am Endgerät ist, nicht jedoch den genauen Abstand. Beide Plattformen bieten anhand dieser hardwarebasierten Sensoren noch weitere softwarebasierte an. So existiert in iOS mit dem Pedometer ein anhand der Bewegungssensoren implementierter Schrittzähler.

Ein detaillierter Überblick über die hardwarebasierten Funktionen nativer Apps kann dem Anhang H entnommen werden.

4.2.2. Softwarebasierte Funktionen

Softwarebasierte Funktionen benötigen zwar trotzdem gewisse Hardware-Komponenten, sie greifen jedoch nicht direkt, sondern eher über hardwarebasierte Funktionen auf diese zu. Da diese softwarebasierten Funktionen bei vorhandener Hardware also in den meisten Fällen eigenständig implementiert werden können, handelt es sich hierbei um Komfortfunktionen. Weil Aspekte, wie der Aufwand oder die Kosten, für die Entscheidung zur Nutzung einer PWA relevant sind, werden diese Komfortfunktionen trotzdem beachtet.

Eine *softwarebasierte Kommunikation* über das Internet kann bspw. anhand von E-Mails erfolgen. Sowohl Android als auch iOS bieten dabei an, eine vorausgefüllte Mail in eine vorinstallierte App zu öffnen. Dabei können Betreff, Empfänger, CC, BCC, Inhalt und Anhang spezifiziert werden. Eine weitere Form der softwarebasierten Kommunikation bieten die im Kapitel 2.3.2.2 beschriebenen Push Notifications. Abhängig von der Plattform können diese mit dem Apple Push Notification Service oder Google Firebase als Push Service umgesetzt werden.

Neben diesen Funktionen sind auch die *Installation* und der *Offlinebetrieb* von nativen Apps für die Betrachtungen sehr relevant. Bei der Installation werden die App-Daten im Dateisystem des Gerätes abgelegt. In den meisten Fällen werden dabei Einrichtungen vorgenommen, durch die sich die App in das Betriebssystem integrieren kann. Die Installation ist häufig Voraussetzung für die Ausführung einer App. Da sich die App-Daten lokal auf dem Gerät befinden, ist ein grundlegender Offlinebetrieb problemlos möglich.

Eine weitere Reihe von Funktionen betrifft die *persönliche Organisation* des Nutzers. Als erstes Beispiel wird hier die in den meisten Fällen vorinstallierte Wecker-App genannt. Bei Android können programmatisch Weckrufe mitsamt Datum, Uhrzeit, Weckton und Vibration konfiguriert werden. iOS erlaubt keinen direkten Zugriff auf die Wecker-App, alternativ kann jedoch eine Erinnerung in der Kalender-App erstellt werden. Diese dient der Organisation von zeitlichen Ereignissen. Auf beiden Plattformen können diese Ereignisse erstellt, bearbeitet und gelöscht werden, wobei je Ereignis Titel, Beschreibung, Start- und Endzeit, Ort und Erinnerungen konfiguriert werden können. Neben den zeitlichen Ereignissen können in einer Kontaktlisten-App auch eine Reihe von Kontaktdaten verwaltet werden. Auf beiden Systemen können Kontakte erstellt, bearbeitet und gelöscht werden, wobei je Kontakt eine Menge von nicht fest definierten Daten angegeben werden können. Sowohl Kalender- als auch Kontaktlisten-App werden dabei mit den cloudbasierten Services der jeweiligen Plattform synchronisiert. Eine letzte Funktion zur persönlichen Organisation bietet die Notizen-App, die stan-

dardmäßig nur bei Android vorhanden ist. Bei dieser können programmatisch einfache Notizen mit Name und Inhalt angelegt werden.

Einen inzwischen sehr häufig genutzten Dienst bieten *Karten-Anwendungen* an. Über diese kann z.B. die über GPS ermittelte Position auf einer Karte angezeigt werden. Zudem bieten Karten-Apps auch ein Geocoding an, bei dem die durch das GPS zurückgegebenen Längen- und Breitengrade in eine vom Menschen lesbare Adresse umgewandelt werden. Zuletzt ist auch ein Geofencing möglich, bei dem interessante Orte in der Nähe der Nutzers identifiziert und diesem vorgeschlagen werden. Diese Karten-Funktionalitäten werden auf beiden Plattformen unterstützt.

Aus wirtschaftlicher Sicht ist der *Vertrieb* einer Anwendung interessant. Native Apps bieten dabei zwei Möglichkeiten: Zum einen kann die App selbst über einen App-Store verkauft werden. Abhängig von der Plattform können dazu der Google Play Store oder der Apple App Store verwendet werden. Zum anderen können digitale Inhalte in Form von einmaligen Leistungen oder Abonnements durch sogenannte In-App Käufe vertrieben werden. Diese In-App Käufe sind auch auf beiden Systemen möglich und müssen beim jeweiligen App-Store genau spezifiziert werden.

Zuletzt bieten beide Plattformen mit der *Verarbeitung von Sprache* sehr fortgeschrittene Funktionen an. Sowohl bei Android als auch bei iOS können mittels Sprachsynthese Text in Sprache und mittels Spracherkennung Sprache in Text umgewandelt werden.

Ein detaillierter Überblick über die softwarebasierten Funktionen nativer Apps kann dem Anhang I entnommen werden.

4.3.Funktionen von Web Apps

Nachdem nun die Möglichkeiten nativer Apps untersucht wurden, soll geprüft werden, welche dieser Funktionen auch in Web Apps genutzt werden können. Die Betrachtungen werden hierbei auf die Browser Chrome, Firefox, Safari und Internet Explorer beschränkt, da diese die größten Marktanteile besitzen.[110] Der Edge Browser wird als offizieller Nachfolger des Internet Explorer zusätzlich mit einbezogen. Da PWAs im Grunde genommen auch Web Apps sind, werden letztere dahingehend abgegrenzt, dass sie die im Kapitel 2.3.2 Progressive Web App Technologien aufgeführten Technologien nicht unterstützen. Die für die Untersuchung verwendeten Quellen sind wieder in einem separaten Dokument aufgeführt.[111]

[110] Vgl. StatCounter (2018a), Abschnitt „Browser Market Share Germany“.
[111] Siehe Anhang G, Quellenverzeichnis zum PWA Regelwerk.

4.3.1. Hardwarebasierte Funktionen

Die *Aufnahme von Medien* ist im Web so wie bei nativen Anwendungen möglich. Zur Aufnahme von Bildern kann einerseits ein `<input>` Element mit `type="file"` verwendet werden, das so konfiguriert wird, dass es nur Bilder akzeptiert. Abhängig vom Betriebssystem wird ein Dateiauswahldialog angezeigt oder eine App zur Aufnahme von Bildern geöffnet. Alternativ kann die Media Capture API verwendet werden. Diese erlaubt zwar nur die Aufnahme von Videos, es kann jedoch ein Standbild aus dem Video ermittelt und in eine Grafik umgewandelt werden. Die Aufnahme von Audio und Video ist analog zur Aufnahme von Bildern möglich. Entweder wird ein `<input>` Element so konfiguriert, dass es nur Audio- oder Videodaten akzeptiert, oder die Media Capture API wird zur Aufnahme verwendet.

Die Darstellung bzw. Wiedergabe von Medien ist im Web ebenfalls möglich. Das Anzeigen von Bildern kann über das `<img>` Element umgesetzt werden. Unterstützt werden Rastergrafiken in den Formaten JPEG, GIF und PNG und Vektorgrafikgen im SVG-Format. Ein fortgeschrittenes Rendering mittels OpenGL ES kann über die WebGL API umgesetzt werden. Das Abspielen von Audio ist über das `<audio>` Element möglich. Hierbei können die Lautstärke, ein Vorladen und das Abspielen in einer Schleife konfiguriert werden. Eine fortgeschrittene Tonverarbeitung bietet die Web Audio API an. Neben den obigen Konfigurationen können Töne über einen Oszillator erzeugt, und Tonhöhe, Panorama und verschiedene Filter eingestellt werden. Die Wiedergabe von Videos ist über ein `<video>` Element möglich. Bei diesem können die Auflösung des Wiedergabebereiches, ein Vorschaubild, ein Vorladen und das Abspielen in einer Schleife konfiguriert werden. Zuletzt können im Web auch mittels einer Vibration API Vibrationen erzeugt werden. Bei dieser können Länge der Vibration und Vibrationsmuster eingestellt und sich wiederholende Vibrationen erzeugt werden.

In Bezug auf die *Persistenz* bietet das Web auch einige Möglichkeiten. Neben der inzwischen veralteten Möglichkeit der Ablage von sogenannten Cookies können Daten als Schlüssel-Wert-Paaren mit Hilfe der Web Storage API abgelegt werden. Dabei können die Daten bis Ende der Sitzung oder darüber hinaus aufbewahrt werden. Komplexe Daten können mittels der IndexedDB API gesichert werden, ein direkter Zugriff auf das Dateisystem, wie es die nativen Apps erlauben, ist im Web jedoch nicht möglich. Ausnahme ist hier das <input> Element vom `type="file"`, bei dem der Nutzer jedoch aktiv die Datei angeben muss, die er hochladen möchte.

Eine *mobilfunkbasierte Kommunikation* ist im Web nur auf indirektem Wege möglich. Ein programmatisches Absetzen eines Telefonates oder einer SMS ist nicht möglich, über Links

mit bestimmten Schemata können jedoch die dafür vorgesehenen, vorinstallierten Apps aufgerufen werden. Diese Schemata sind in den Requests For Comment (RFC) 3966 und 5724 definiert. Für ein Telefonat muss ein Link im tel-Schema und für eine SMS ein Link im sms-Schema angegeben werden.

Die *internetbasierte Kommunikation* bildet den Kern einer jeden Webanwendung, die Abfrage des Netzwerkstatus wird jedoch nur partiell unterstützt. Zwar existiert zu diesem Zweck eine Network Information API, deren Spezifikation wird jedoch vom World Wide Web Consortium (W3C) nicht weitergeführt. Somit ist es nur in den Browsern Chrome und Firefox möglich, die Downloadrate und den Verbindungstyp zu ermitteln. Eine allgemeine Kommunikation über das Internet ist hingegen in jedem Browser möglich. Eine Funktion, die jedoch nicht immer vorhanden war, besteht darin, Ressourcen nachträglich zu laden. Diese Funktion wird Asynchronous JavaScript and XML genannt und kann über sogenannte XML Http Requests umgesetzt werden.

Aus der Reihe der *sonstigen Kommunikationskanäle* wird das GPS im Web sehr gut über die Geolocation API unterstützt. Diese benötigt eine HTTPS-Verbindung und liefert in einer einmaligen oder kontinuierlichen Abfrage die Längen- und Breitengrade zur aktuellen Position des Endgerätes. Für die Kommunikation mittels Bluetooth hat die Web Bluetooth Community Group eine Web Bluetooth API spezifiziert. Bei dieser API handelt es sich jedoch um keinen W3C Standard, wodurch sie lediglich von Chrome bei einer HTTPS-Verbindung unterstützt wird. Die Kommunikation über NFC soll über eine Web NFC API möglich sein, die Spezifikation dazu gilt jedoch als veraltet und wird nicht mehr vom W3C weiterentwickelt, wodurch eine Nutzung von NFC-Funktionalitäten im Web nicht möglich ist. Zuletzt soll eine Kommunikation über USB durch die Web USB API ermöglicht werden, die zugehörige Spezifikation ist jedoch noch kein W3C Standard und wird deshalb momentan nur von Chrome bei einer HTTPS-Verbindung unterstützt.

Das Auslesen von *Sensoren* im letzten Bereich der hardwarebasierten Funktionen ist im Web nur sehr sporadisch möglich. Das Auslesen einiger Sensorinformationen ist über verschiedene Ereignisse möglich. So kann der Bewegungssensor über das `devicemotion`-Event, der Rotationssensor über das `deviceoriantation`-Event, der Umgebungslichtsensor über das `devicelight`-Event und der Näherungssensor über das `userproximity`-Event ausgelesen werden. Diese Ereignisse werden jedoch nur partiell und teilweise auch nur bei bestimmten Konfigurationen von den Browsern unterstützt. Eine Alternative bietet sich mit dem aktuell entwickelnden Standard der Generic Sensor API an. Sie definiert allgemeine Informationen

zu Sensoren und kann über weitere APIs spezifiziert werden. So existieren für den Bewegungssensor die Accelerometer API, für den Rotationssensor die Gyroscope API, für den Umgebungslichtsensor eine Ambient Light Sensor API, für den Magnetfeldsensor eine Magnetometer API und für den Näherungssensor eine Proximity Sensor API. Aufgrund der Neuheit dieser Technologien werden sie aktuell kaum bis gar nicht von den Browsern unterstützt. Hinzu kommt, dass Temperatur-, Luftdruck- und Luftfeuchtigkeitssensor weder über ein Ereignis noch über eine API ausgelesen werden können.

Ein detaillierter Überblick über die hardwarebasierten Funktionen von Web Apps kann dem Anhang J entnommen werden.

4.3.2. Softwarebasierte Funktionen

Eine *softwarebasierte Kommunikation* ist im Web nur indirekt möglich. So kann zum Versenden von E-Mails wieder ein Link im mailto-Schema verwendet werden, da im RFC 6068 definiert ist. Nach diesem Schema kann eine vorausgefüllte Mail mit Empfänger, CC, BCC, Betreff und Nachrichteninhalt in einer installierten Mail-Anwendung geöffnet werden. Das Empfangen von Push Notifications wird erst durch die neue Service Worker Technologie ermöglicht und ist somit in herkömmlichen Web Apps nicht möglich.

Da eine *Installation* erst durch das neue Web App Manifest ermöglicht wird, ist diese bei herkömmlichen Webanwendungen ebenfalls nicht möglich. Im Gegensatz dazu gab es mit dem Application Cache bereits vor dem Service Worker Bemühungen, einen Offlinebetrieb im Web zu ermöglichen. Da der Application Cache jedoch verschiedene Nachteile mit sich brachte, wird er zugunsten der Service Worker nach und nach aus dem Web entfernt.

Die Unterstützung der Funktionen zur *persönlichen Organisation* unterscheidet sich teilweise recht stark. Gerade weil es sich um plattformspezifische Services handelt, ist der Browsersupport stark von den Schnittstellen abhängig, die der jeweilige Hersteller anbietet. So existiert z.B. keine API, über die auf die Wecker- oder Notizen-Apps des jeweiligen Endgerätes zugegriffen werden kann. Ein integrierter Zugriff auf den Kalender, so wie es bei nativen Apps möglich ist, kann im Web nicht genutzt werden. Jedoch bietet Google bspw. für Google Calendar eine Calendar API an, über die die Kalenderdaten in die eigene Webseite integriert werden können. Apple bietet hingegen für seinen iCloud Calendar keine öffentliche API an. Ähnlich sieht es bei der Kontaktliste aus. Google Contacts können über die People API in eine Web App integriert werden, während für iCloud Contacts keine öffentliche API existiert.

Karten-Anwendungen, unabhängig davon ob Google oder iCloud Maps, können und werden heutzutage bereits recht häufig in Webseiten integriert. Dabei ist das Anzeigen der über die Geolocation API ermittelten Position auf der Karte, das Geocoding und das Geofencing möglich. Da diese Funktionen über eine externe API integriert werden, sollten sie auch in allen Browsern lauffähig sein.

Im Bereich des *Vertriebs* unterscheiden sich jedoch die Möglichkeiten zwischen nativen und Web Apps. Ein direkter Verkauf über App-Stores ist bei Web Apps nicht möglich, da diese frei über das Internet aufgerufen werden können. Web Apps müssen hierbei auf die In-App Käufe zurückgreifen, die z.B. über die Payment Request API umgesetzt werden können. Dies ist eine recht neue API, die nur von modernen Browsern unterstützt wird und einen schnellen, einfachen, barrierefreien und sicheren Zahlungsprozess gewährleistet. Wird die Payment Request API nicht unterstützt, können Webseiten noch über den üblichen Checkout von Online-Shops zu Geld kommen.

Zuletzt bietet das Web zur *Sprachverarbeitung* eine Web Speech API an, über die sowohl Sprachsynthese als auch Spracherkennung umgesetzt werden können. Da es sich hierbei um sehr fortgeschrittene Technologien handelt, wird die API nur partiell unterstützt. Alternativ kann die Sprachverarbeitung auch wieder über verschiedene Webservices umgesetzt werden, wodurch diese Funktionalität dann von allen Browsern unterstützt wird. Als ein Beispiel unter vielen seien hier Googles cloudbasierte Text-To-Speech und Speech-To-Text Services genannt.

Ein detaillierter Überblick über die softwarebasierten Funktionen von Web Apps kann dem Anhang K entnommen werden.

4.4. Folgerung der Funktionen von Progressive Web Apps

Nachdem nun klargestellt wurde, welche Möglichkeiten native Apps bieten und welche davon auch in Web Apps nutzbar sind, können die Funktionen von PWAs abgeleitet werden. Im Kapitel 2.3 Progressive Web Apps wurde bereits festgestellt, dass PWAs in ihrem Kern einfache Webanwendungen sind, was bedeutet, dass sie prinzipiell dieselben Funktionen wie Web Apps besitzen. Hinzu kommen an dieser Stelle noch die Funktionen, die durch die im Kapitel 2.3.2 beschriebenen Progressive Web App Technologien ermöglicht werden. Dabei handelt es sich konkret um den Offlinebetrieb, die Push Notifications und die Installation der PWA. Ersterer wird über eine Kombination von Service Worker als clientseitiger Proxy und Caching ermöglicht und wird von den Browsern Chrome ab Version 40, Firefox ab Version 44, Safari ab Version 11.1 und Edge ab Version 17 und vom Internet Explorer nicht unterstützt. Für die

Umsetzung der Push Notifications wird neben den Service Workern noch die vom W3C spezifizierte Push API benötigt. Diese wird von den Browsern Chrome ab Version 40, Firefox ab Version 44 und Edge ab Version 17 unterstützt. Safari bietet hierzu nicht die vom W3C vorgegebene, sondern einer eigene Push API an. Zuletzt wird die Installation von PWAs über das Web App Manifest ermöglicht und von den Browsern Chrome ab Version 39, Firefox ab Version 47, Safari ab Version 11.3 und Edge ab Version 17 unterstützt. Hierbei muss noch ergänzt werden, dass die Installation über Firefox und Safari nur auf mobilen Endgeräten funktioniert.

Die im Anhang K aufgeführte Tabelle zu den softwarebasierten Funktionen von Web Apps kann für PWAs übernommen und wie in Tabelle 4 angepasst werden:

Funktionalitäten	**Chrome**	**Firefox**	**Safari**	**IE**	**Edge**	**Gesamt**
Push Notifications	40	44	~[1]	-	17	~
Installation	39	47[2]	11.3[2]	10[3]	17	~
Offlinebetrieb	40	44	11.1	-	17	~

Legende und Fußnoten

+	Funktion vorhanden	~	Funktion teilweise vorhanden	-	Funktion nicht vorhanden

1. Für Safari muss eine von Apple vorgegebene Push API genutzt werden.
2. Diese Funktion ist nur auf mobilen Endgeräten verfügbar.
3. Der Offlinebetrieb ist im Internet Explorer nur über den veralteten Application Cache möglich.

Tabelle 4: Ergänzung der Progressive Web App Funktionen
Quelle: Vgl. Anhang G, Quellenverzeichnis zum PWA Regelwerk

5. Regelwerk zur Nutzung von Progressive Web Apps

5.1. Erstellung des Regelwerks

Nachdem nun die Funktionen von Web Apps mit denen der PWAs ergänzt wurden, sind alle Informationen zur Erstellung des Regelwerks vorhanden. Dieses beginnt inhaltlich mit einem *einleitenden Kapitel*, das den Leser auf die Nutzung des Regelwerks vorbereiten soll. Falls jemand durch Zufall oder ohne Vorwissen auf das Regelwerk stößt, wird in einem Abschnitt zunächst der Zweck des Dokuments erläutert. Für Leser, die kein Vorwissen zu PWAs besitzen, wird zudem die Definition von Progressive Web Apps aus dem Kapitel 2.3.3 aufgeführt. Zuletzt wird in einem Abschnitt beschrieben, wie das Regelwerk zu nutzen ist, falls ein Leser dieses das erste Mal verwendet.

Nach dem einleitenden Kapitel folgen die *einzelnen Regeln* so, wie sie in den Anhängen H und I strukturiert wurden. Dabei ist zu jeder Regel eine knappe, fachliche Beschreibung gegeben, sodass die Anwendung der Funktion ersichtlich wird. Falls die Funktion durch eine offizielle Spezifikation beschrieben wird, ist auch der Link dazu aufgeführt. Zuletzt wird je Funktion noch eine Tabelle mit dem Browser-Support für die Browser Chrome, Firefox, Safari, Internet Explorer und Edge aufgeführt. Wenn eine fachliche Funktion mit mehreren Technologien umgesetzt werden kann, werden alle Technologien beschrieben. An dieser Stelle könnten noch nützliche Links zu Tutorials etc. je Funktion aufgeführt werden, diese wurden jedoch aus mehreren Gründen nicht in das Regelwerk aufgenommen. Zum einen würden diese Links das Regelwerk, das ohnehin schon viel Text enthält, weiter aufblähen. Zum anderen könnten die Links mit der Zeit veralten, wodurch sich der Aufwand für die Pflege des Regelwerks weiter erhöhen würde. Entwickler erhalten aber für eine Websuche geeignete Stichwörter.

Im vorletzten Kapitel „*Gutachten und Korrektur*“ wird die Leistung der Experten anerkannt, die durch ein Interview zur Korrektur und Evaluation des Regelwerks beigetragen haben. Abgeschlossen wird das Regelwerk mit einer *Versionshistorie*, in der die Änderungen beschrieben werden, die zwischen den einzelnen Versionen vorgenommen wurden. Diese Metainformationen dienen eher der Nachvollziehbarkeit im Rahmen dieser Masterarbeit und können bei Veröffentlichung des Regelwerks entfernt werden. Das Regelwerk wurde der digitalen Version dieser Masterarbeit beigefügt.[112]

[112] Siehe Anhang L, Regelwerk- und Interview-Dokumente.

5.2.Evaluation des Regelwerks

5.2.1. Auswahl und Begründung der Methodik

Da das Regelwerk bisher lediglich von einer einzelnen Person erstellt wurde, ist es aus wissenschaftlicher Sicht nicht belastbar und sollte evaluiert werden. Das Ziel der Evaluation ist es, die Korrektheit und Vollständigkeit des Regelwerks zu ermitteln und zu verbessern. Mit den Begriffen der empirischen Sozialforschung ausgedrückt ist das Regelwerk also der Merkmalsträger mit den Merkmalen „Korrektheit“ und „Vollständigkeit“. Die Korrektheit des gesamten Regelwerks ist dabei ein komplexes Merkmal, das sich aus der Korrektheit der einzelnen Regeln ergibt. Der Einfachheit halber wird allen Regeln dieselbe Bedeutung beigemessen, wodurch sich die Korrektheit des Regelwerks dann aus dem Durchschnitt der Korrektheit aller Regeln errechnen lässt. In der Theorie lässt sich der Grad der Korrektheit als metrische Zahl, also über eine intervall- oder verhältnisskalierte Variable beschreiben. Anders sieht es mit dem Merkmal der Vollständigkeit aus, das die Aspekte aufdecken soll, die für den Entscheidungsprozess zur Nutzung einer PWA relevant sind und noch nicht im Regelwerk aufgeführt wurden. Die Vollständigkeit des Regelwerks ergibt sich dabei anhand des Anteils der nicht beachteten Aspekte an der Gesamtheit der Regeln. Da bei diesem Merkmal unbekannte Ausprägungen erhoben werden sollen, muss eine offene Frage genutzt werden, wodurch wiederum qualitative Daten erhoben werden. Zuletzt können nicht nur die Korrektheit und Vollständigkeit, sondern auch noch andere, unbekannte Aspekte für das Regelwerk relevant sein. Diese sonstigen Eigenschaften sollen in einem weiteren Merkmal erfasst werden.

Mit diesen Vorüberlegungen soll nun die empirische Erhebungsmethode ausgewählt werden. Eine der im Kapitel 3.2 aufgeführten Methoden zur empirischen Datenerhebung ist die *Beobachtung*. Diese ist für die Evaluation des Regelwerks ungeeignet. Das Dokument kann zwar ausgedruckt oder auf einem Bildschirm betrachtet werden, es wird jedoch ohne äußere Einflüsse kein beobachtbares Verhalten zeigen. Selbst mit äußeren Einflüssen sind die beobachtbaren Eigenschaften für die Korrektheit und Vollständigkeit des Regelwerks irrelevant. Hinzu kommt, dass es sich bei der Korrektheit und Vollständigkeit um „innere“ Merkmale handelt, die durch eine äußerliche Betrachtung nicht erfassbar sind. Eine weitere Erhebungsmethode ist die *Inhaltsanalyse*, die theoretisch eine valide Form der Evaluation wäre. Durch eine Analyse verschiedener Quellen kann geprüft werden, ob sich die Inhalte des Regelwerks mit denen der Quellen decken, was Rückschlüsse auf die Korrektheit zulässt. Zudem kann geprüft werden, ob diese Quellen weitere Aspekte enthalten, wodurch die Vollständigkeit geprüft werden kann. Praktisch gesehen ist dieses Evaluationsverfahren jedoch ungeeignet, weil

durch eine Recherche verschiedener Quellen bereits die Informationen ermittelt wurden, anhand denen das Regelwerk erstellt wurde. Indirekt würden also die Inhalte derselben Quellen verglichen werden. Sollten an dieser Stelle also Inkonsistenzen erscheinen, lässt das eher Rückschlüsse auf die Qualität der Recherche aber nicht auf die Korrektheit und Vollständigkeit des Regelwerks zu. Die einzigen, praktisch nutzbaren Erhebungsmethoden sind also die Interviews bzw. Befragungen.

Da durch die offenen Fragen zwangsläufig qualitative Daten erhoben werden, fallen *vollständig standardisierte Interviews* weg. Die Inhalte von *nicht-standardisierten Interviews*, wie z.B. das narrative bzw. ethnografische Interview oder die Fokusgruppe, sind vergangene Erlebnisse oder eher emotionale Konstrukte, wie Werteorientierungen, Meinungen, Gefühle usw. Auf Grund der inhaltlichen Thematik sind diese Interviewformen nicht zur Evaluation des Regelwerks geeignet. Übrig bleiben noch die *teil-standardisierten Leitfadeninterviews*, bei denen zuvor recherchierte Inhalte durch die subjektive Sicht der Befragten ergänzt werden sollen. Die recherchierten Inhalte sind hier mit den einzelnen Regeln gleichzusetzen, die im Regelwerk zusammengefasst wurden. Der Leitfaden eines solchen Interviews ergibt sich hierbei grob aus den beiden zu untersuchenden Merkmalen und der Struktur der einzelnen Regeln. Besonders geeignet ist an dieser Stelle die Spezialform des Experteninterviews, bei dem Experten zum Thema der PWAs befragt werden. Durch das Wissen, das die Experten als solche auszeichnet, ist deren Meinung zur Korrektheit und Vollständigkeit aus wissenschaftlicher Sicht besonders relevant und trägt im hohen Maße zur Validität des Regelwerks bei.

Wie im Kapitel 3.2.1.2 Teil-Standardisierte Interviews bereits erwähnt wurde, ist die Qualität solcher Experteninterviews stark von der Auswahl der Experten abhängig. Aus diesem Grund sollen in diesem Rahmen nur solche Personen als Experten erachtet werden, die

1. entweder an der Konzeption oder Umsetzung der im Regelwerk aufgeführten Funktionen in einem Browser beteiligt waren, oder
2. langjährige Erfahrung im Bereich der Webentwicklung besitzen und zum aktuellen Zeitpunkt noch in diesem Bereich aktiv sind.

Zum ersten Punkt zählen Webentwickler, die an der Programmierung eines Browsers wie z.B. Chrome, Firefox oder Safari beteiligt sind. Daraus ergibt sich, dass an die Unternehmen Google, Mozilla, Apple und Microsoft Anfragen für ein Interview zum Thema PWAs geschickt wird. Zum zweiten Punkt zählen z.B. Webentwickler mit langjähriger Erfahrung (Senior Web Developer) oder Webentwickler in führenden Positionen (Chief Information Officer).

Nachdem nun die Erhebungsmethode feststeht, kann die Form der erhobenen Daten konkretisiert werden. Die Korrektheit einer Regel soll dadurch ermittelt werden, wie sehr ihr ein Experte zustimmt. Dieser Sachverhalt soll anhand einer Zustimmungsskala gemessen werden. Wie bereits erwähnt können hierbei metrische Werte erfasst werden, der Einfachheit halber soll jedoch eine Ordinalskala mit den folgenden Ausprägungen verwendet werden: „stimme vollständig zu, stimme größtenteils zu, stimme teilweise zu, stimme größtenteils nicht zu, stimme gar nicht zu“. Zusätzlich soll noch die Option „weiß nicht“ angeboten werden, falls ein Experte zu einer Regel keine Aussage treffen kann. Durch diese Zustimmungsskala wird jedoch nur der Grad der Korrektheit ermittelt, nicht jedoch, welcher konkrete Aspekt der Regel inkorrekt ist. Da dieser Aspekt vor der Datenerhebung nicht bestimmbar ist, muss hierzu eine offene Frage gestellt werden, die in etwa wie folgt lauten könnte: „Wenn Sie dieser Regel nicht vollständig zustimmen, spezifizieren Sie bitte, warum nicht?“. Die Vollständigkeit des Regelwerks soll über eine offene Frage ermittelt werden, die in etwa wie folgt lauten könnte: „Fallen Ihnen weitere Funktionen ein, die für die Nutzbarkeit von Progressive Web Apps relevant sind und noch nicht im Regelwerk aufgeführt wurden?“. Alle anderen, für das Regelwerk relevanten Aspekte sollen über eine offene Frage erfasst werden, die wie folgt lauten könnte: „Haben Sie noch weitere Anmerkungen oder Kritik zum Progressive Web App Regelwerk?“. Zuletzt sollen für eine gute Verwertbarkeit der Ergebnisse noch die soziodemographischen Daten der Experten aufgenommen werden. Die Experten sind hierbei der Merkmalsträger, die die folgenden Tabelle 5 aufgeführten Merkmale besitzen. Über einige dieser Merkmale ist dann auch die Eignung der befragten Person als Experte erkennbar.

Merkmal	**Ausprägungen**	**Skalenniveau**
Geschlecht	männlich, weiblich	Nominal
Alter	Zeit in Jahren	Verhältnis
Höchster Bildungsabschluss	Hauptschulabschluss, Realschulabschluss, (Fach)Hochschulreife, Bachelor, Master / Diplom, Promotion	Ordinal
Berufliche Tätigkeit	Berufsbezeichnung, z.B. CEO, Webentwickler usw.	Nominal
Erfahrung im Beruf	Zeit in Jahren	Verhältnis

Tabelle 5: Merkmale der soziodemographischen Daten

Zuletzt muss noch beachtet werden, dass sich abgesehen von den soziodemographischen Daten alle Fragen auf das Regelwerk beziehen. Um diese Fragen beantworten zu können, muss der Experte zuvor noch ein *Peer Review* bzw. eine *Peerevaluation* dazu durchführen. Aus organisatorischen Gründen findet dieses Peer Review in Form einer offenen Begutachtung statt, was bedeutet, dass der Autor des Regelwerks und der Experte einander bekannt sind.

Die Nachteile einer solchen offenen Begutachtung, wie z.B. die eventuelle Voreingenommenheit der Experten gegenüber einen Studenten, sollten dadurch reduziert werden, dass beide Parteien nach dem Review direkt aufeinandertreffen, wobei der Experte seine Meinung begründet darlegen muss. Zudem soll das Interview aufgezeichnet, transkribiert und dieser Masterarbeit beigefügt werden, wodurch die Ergebnisse des Peer Reviews wie bei einer offenen Begutachtung mit der geprüften Arbeit veröffentlicht werden.

5.2.2. Durchführung der Evaluation

Vor der eigentlichen Durchführung der Experteninterviews wurden mehrere Anfragen für ein Interview zum PWA-Thema an relevante Unternehmen und Experten geschickt. Dabei wurde in einem ersten Schritt versucht, Kontakt zu den Browserherstellern aufzubauen. Der Versuch, sowohl telefonisch als auch per Mail Kontakt zur Google herzustellen, war nicht erfolgreich. Grund ist, dass das vollständig automatisierte Supportsystem solche Forschungsanfragen nicht abdeckt. Über andere Kanäle war Google ebenfalls nicht erreichbar, weil es keine weiteren Kontaktmöglichkeiten für die Öffentlichkeit anbietet. An Mozilla, dem Unternehmen hinter dem Firefox Browser, wurde eine Mail an die extra für Forschungsanfragen vorgesehen Adresse research_requests@mozilla.com gesendet. Zum aktuellen Zeitpunkt liegt noch keine Antwort vor und es wird auch keine mehr erwartet. Die Anfrage an Apple wurde über ein Online-Formular in deren Entwicklerbereich gestellt. Der Apple Kundenservice hat zwar geantwortet, lehnt ein Interview jedoch auf Grund der Vertraulichkeit und Geheimhaltung ihrer Technologien ab. Zuletzt wurde eine Anfrage an Microsoft zunächst über deren Online-Support gestellt. Da es sich hierbei um eine sehr spezifische Anfrage handelt, hat der Online-Support auf die Mailadresse kunden@microsoft.com verwiesen. Eine Anfrage an diese Adresse hatte zur Folge, dass ein Service Request Ticket erstellt wurde, welches jedoch ohne Antwort wieder geschlossen wurde. Ein Interview mit einem der Browserhersteller war somit ohne interne Kontakte nicht möglich. Abgesehen von diesen vier Unternehmen wurde eine Anfrage an sechs weiteren Experten geschickt, von denen sich drei zum Interview bereiterklärt haben. Diese drei Experten sind konkret Achim Gosse, Chief Information Officer des IT-Unternehmens digitalnoise, Kevin Taron, Geschäftsführer der Gutwerker Digitalagentur und Peter Kröner, ein selbstständiger Spezialist für Webtechnologien. Zu keinem der drei Experten bestand vor dem Interview ein persönlicher Kontakt.

Mit der Anfrage wurden das PWA-Regelwerk und ein Interviewleitfaden[113] geschickt. Das Interview wurde abhängig von der räumlichen Entfernung im direkten Gespräch oder über ein Skype-Telefonat durchgeführt. Das Interview wurde aufgezeichnet, wobei vorher eine Einwilligung in die Audioaufzeichnung mündlich eingeholt wurde. Die Interviews mit Herrn Gosse und Herrn Kröner haben sich recht stark am Leitfaden orientiert. Herr Taron hatte bereits vor dem Interview seine Ergebnisse schriftlich übermittelt. Weil in diesen Ergebnissen bereits ersichtlich war, dass Herr Taron dem größten Teil der Regeln vollständig zustimmte, wurden im Interview nur noch die offenen Punkte besprochen.

In der anschließenden Nachbereitung wurden die Interviews anhand der Aufzeichnungen transkribiert. Auf Basis der Transkriptionen wurden die Interviews mit Hilfe einer qualitativen Inhaltsanalyse ausgewertet. Dabei wurden bereits in der Transkription durch eine Explikation erklärungsbedürftige Begriffe in Fußnoten erläutert. Anschließend wurde das Interview in einem separaten Dokument zusammengefasst und gekürzt, wobei die für die Evaluation relevanten Daten erhalten wurden. Zuletzt wurden die Daten anhand der im vorherigen Kapitel 5.2.1 beschriebenen Merkmale strukturiert, wobei die Korrektheit der Regeln und die soziodemographischen Daten in die vorgegebenen Skalen codiert wurden. Problematisch bei dieser Codierung war es, die Aussage „Mir ist nichts weiter bekannt“ auf die Frage „Funktion XYZ wird im Web nicht unterstützt“ zu deuten. Die Antwort ist in diesem Kontext mehrdeutig und kann einerseits so gedeutet werden, dass der Experte keine Aussage dazu machen kann, was der Option „weiß nicht“ entspricht, oder es kann so gedeutet werden, dass der Experte zustimmt, weil er ansonsten von einer Möglichkeit zur Umsetzung der Funktion gewusst hätte. Diese Mehrdeutigkeit wurde dadurch aufgelöst, dass sowohl die Transkription als auch die Auswertung anschließend zum jeweiligen Experten geschickt wurde, welcher diese dann bestätigt oder ggfs. korrigiert hat. In vielen Fällen hat der Experte dabei eine Mischung aus „weiß nicht“ und Zustimmung angegeben. Auf diese Weise wurde noch eine kommunikative Validierung vorgenommen, wie sie bei qualitativen Methoden üblich ist.

5.2.3. Ergebnisse und Interpretation der Evaluation

Zuletzt wurden die Zusammenfassungen und Codierungen der Interviews in einer Tabellenkalkulation zusammengeführt und, soweit es möglich war, mit quantitativen Methoden statistisch ausgewertet. Die Korrektheit wurde über eine Zustimmungsskala mit den Werten „stimme vollständig zu“, „stimme größtenteils zu“ usw. gemessen. Diese Werte können in eine logische Rangfolge gebracht werde und sind somit ordinalskaliert, es kann jedoch impli-

[113] Siehe Anhang L, Regelwerk- und Interview-Dokumente.

zit eine Bedeutung in die Abstände zwischen den Ausprägungen interpretiert werden. So können den Ausprägungen numerische Werte zugeordnet werden, wie der Ausprägung „stimme vollständig zu“ der Wert 1, der Ausprägung „stimme größtenteils zu“ der Wert 0,75 bis hin zur Ausprägung „stimme gar nicht zu“ mit dem Wert 0. Wurde zu einer Regel „weiß nicht“ angegeben, dann wird diese so ignoriert, dass sie keine Auswirkungen auf die Berechnungen hat. Auf diese Weise können nun mit dem Grad der Zustimmung metrische Auswertungen durchgeführt werden, die jedoch mit Vorsicht betrachtet werden sollten, weil die Zustimmung streng genommen nur ordinalskaliert ist.

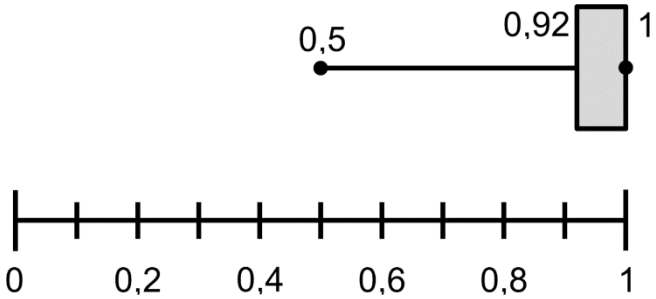

Abbildung 7: Boxplot zur Korrektheit des Regelwerks
Quelle: eigene Darstellung mit Daten aus Anhang L, Regelwerk- und Interview-Dokumente

Der in der obigen Abbildung 7 gezeigte Boxplot beschreibt die Korrektheit des gesamten Regelwerks. Der minimalste Wert, der angegeben wurde, war 0,5 (stimme teilweise zu). Der maximalste Wert war 1 (stimme vollständig zu). Das erste und dritte Quantil, durch die die linke und rechte Grenze der Box bestimmt werden, wurden anhand der Mittelwerte der Regeln über die Angaben der drei Experten bestimmt. Das erste Quantil sagt mit dem Wert 0,92 aus, dass 75% der Regeln zu mindestens 92% zugestimmt wurde. Das dritte Quantil sagt mit dem Wert 1 entsprechend aus, dass 25% der Regeln zu 100% zugestimmt wurde. Tatsächlich sind es mit 57% sogar über die Hälfte der Regeln, denen zu 100% zugestimmt wurde. Da der Median somit ebenfalls bei 1 liegt, wurde er hier nicht eingezeichnet. Der Mittelwert der Zustimmung aller Regeln liegt bei 0,95. Das bedeutet, dass das Regelwerk im Ergebnis vor den Interviews zu 95% korrekt war. Zudem wurde über die Varianz die Standardabweichung berechnet, welche nicht im Boxplot auftaucht. Der Wert der Standardabweichung ist mit 0,114 recht gering, was darauf schließen lässt, dass sich die Experten in Bezug auf die Korrektheit der Regeln recht einig waren.

Neben der Korrektheit wurden auch die verschiedenen Aussagen untersucht, die die Experten während den Interviews gemacht haben. Bei diesen Aussagen handelt es sich um rein qualitative Daten, weswegen eine qualitative Auswertung mittels Inhaltsanalyse durchgeführt wurde. Dabei wurden die Anmerkungen anhand ihrer Semantik strukturiert, sodass ermittelt werden kann, wenn mehrere Experten inhaltlich identische Anmerkungen gemacht haben. Insgesamt

haben die drei Experten 24 Anmerkungen getätigt, also durchschnittlich acht pro Experte. Keine der Anmerkungen decken sich inhaltlich, was bedeutet, dass die Experten mit jeder Anmerkung ein unterschiedliches Thema angeschnitten haben. Von diesen 24 Anmerkungen wurden mit 11 ca. 46% in das Regelwerk übernommen. Die Begründungen zur Ablehnung der restlichen Anmerkungen können dem Anhang M entnommen werden. Eine Ablehnung ist dabei nicht zwangsläufig darauf zurückzuführen, dass die Anmerkung in irgendeiner Form falsch war. Aus diesen Ablehnungen können also keine Rückschlüsse auf die Qualität der Anmerkungen oder der Experten gezogen werden.

Die Anmerkungen bilden auch den Ausgangspunkt für die Auswertung der Vollständigkeit. Den drei Experten wurde ein Regelwerk mit 37 technischen Funktionen vorgelegt. Die Experten haben 12 Anmerkungen zur Vollständigkeit gemacht, von denen fünf in das Regelwerk übernommen wurden. Vier der übernommenen Anmerkungen bezogen sich dabei auf die Vollständigkeit von vorhandenen Regeln. Weil zwei Anmerkungen dieselbe Regel betrafen, waren also drei der vorhandenen Regeln unvollständig. Aus der letzten der fünf übernommenen Anmerkungen wurde eine neue Regel generiert. Nach den Interviews existieren nun 38 Regeln, von denen 34 Regeln vor den Interviews vollständig waren. Daraus ergibt sich eine Vollständigkeit des Regelwerks von 89%. Der Fairness halber soll an dieser Stelle noch der Grad der Vollständigkeit angegeben werden, der sich ergeben würde, wenn alle Anmerkungen übernommen worden wären. In diesem Fall hätte das Regelwerk nach der Korrektur 39 Regeln, von denen 30 Regeln vollständig gewesen wären. In diesem Fall ergäbe sich also eine Vollständigkeit von 77%.

Zuletzt kann anhand der soziodemographischen Daten noch die Qualität der Experten betrachtet werden. Alle drei Experten waren männlich und im Alter von 31 bis 45. Aus beruflicher Sicht sind mit einem Chief Information Officer ein Experte in einer führenden Position im IT-Bereich eines Unternehmens, ein Geschäftsführer einer Internetagentur, und ein Trainer und Spezialist für Webtechnologien vertreten. Dies sind zwar recht heterogene Berufe, die jedoch mehr oder weniger direkt mit der Webentwicklung verzahnt sind. Die Experten weisen in der Summe 41 Jahre Erfahrung im Beruf auf, also durchschnittlich 14 Jahre pro Experte. In diesem Zeitraum kann problemlos genügend Wissen gesammelt werden, um als Experte gelten zu können. Aus soziodemographischer Sicht ist die Wahl der drei Personen als Experten also durchaus gerechtfertigt.

Zuletzt sollte diese Evaluation des Regelwerks noch mit kritischem Blick betrachtet werden. Zum einen ist die Auswertung bereits anfechtbar, weil die eigentlich ordinalskalierten Daten zur Korrektheit in metrische Werte umgewandelt wurden. Zum anderen ist eine Evaluation mit einer Stichprobe von drei Experten wahrscheinlich nicht ausreichend repräsentativ. Bereits durch einen weiteren Experten mit kritischerem Block könnte das Ergebnis wesentlich negativer ausfallen. Hinzu kommt, dass die Wahl der Experten zwar aus soziodemographischer Sicht valide war, es hätten sich jedoch um einiges genauere Ergebnisse bei Interviews mit den Browserherstellern ergeben. Zuletzt wurden anhand der Auswertung lediglich die Korrektheit und Vollständigkeit des Regelwerks vor den Interviews ermittelt, es ist aber keine sichere Aussage darüber möglich, wie korrekt und vollständig das Regelwerk im aktuellen Zustand nach den Interviews ist. Würden die Experteninterviews wiederholt werden, dass ließe sich das Problem mit dem Messniveau dadurch beheben, dass die Korrektheit über eine metrische Skala ermittelt wird. Die geringere Repräsentanz lässt sich durch mehr Interviews verbessern, wobei abhängig von der Qualität der Experten ca. fünf bis zehn Interviews ein gutes Ergebnis liefern sollten. Voraussetzung dafür ist allerdings, dass die zeitlichen und organisatorischen Rahmenbedingungen gegeben sind.

6. Leitfaden-App zur Nutzung von Progressive Web Apps

Mit einem vollständig evaluierten und korrigierten Regelwerk kann nun das zweite Teilergebnis der Masterarbeit umgesetzt werden: Die Leitfaden-App.

6.1. Entwicklung der Leitfaden-App

Da die heutige Welt zunehmend digitalisiert wird, sollte sich das Regelwerk nicht nur auf einen statischen Text beschränken, sondern auch in einer interaktiven App umgesetzt werden. Auf Grund des Themas dieser Masterarbeit bietet es sich sehr an, das Regelwerk als PWA umzusetzen. Diese App soll im Folgenden Leitfaden-App genannt werden. Um ein geeignetes Ziel zu finden, wird der Anwendungsfall der App betrachtet. Die Ausgangssituation besteht darin, dass ein Entscheidungsträger prüfen will, ob eine gegebene Reihe von Anforderungen über eine PWA umgesetzt werden kann. Würde diese Prüfung mit Hilfe des Regelwerks durchgeführt werden, müsste der Prüfer die Browserkompatibilitäten der einzelnen Funktionen händisch abgleichen. Das Ziel der Leitfaden-App soll also sein, die Entscheidungsfindung durch eine Reduktion der händischen Arbeit zu beschleunigen.

Die *konzeptionelle Umsetzung* der Leitfaden-App orientiert sich ebenfalls an dem Anwendungsfall. Analog zum Regelwerk soll einleitend ein Verständnis für die App aufgebaut werden. Dazu sollen der PWA-Begriff definiert und der Zweck und die Nutzung der Leitfaden-App erläutert werden. Die Nutzung erfolgt in zwei Schritten: Zunächst soll der Nutzer die Browser und die Versionen, die unterstützt werden sollen, eingeben. Im zweiten Schritt sollen dann anhand der Anforderungen die benötigten Funktionen angegeben werden. Hierbei werden die Namen der Technologien aber nicht deren Beschreibungen aufgeführt. Das hat den Vorteil, dass eine schnellere Eingabe durch weniger zu lesenden Text möglich ist. Gleichzeitig bringt es jedoch den Nachteil einer schlechteren Verständlichkeit mit, wenn eine Funktion unbekannt ist. Analog zum Requirements Engineering soll zu jeder Funktion angegeben werden können, ob es sich um ein Pflicht-, Kann- oder Abgrenzungskriterium handelt. Anhand der eingegebenen Daten kann die Leitfaden-App dann prüfen, ob eine PWA die gegebenen Funktionen in den Browsern und ihren Versionen unterstützt. Eine einzelne Funktion kann dabei vollständig, teilweise oder nicht unterstützt werden.

Die Eignung einer PWA über alle Funktionen ergibt sich dann wie folgt:

- Werden alle Pflicht- und Kann-Kriterien vollständig unterstützt, dann ist eine PWA geeignet.
- Wird mindestens ein Pflichtkriterium nur teilweise oder ein Kann-Kriterium teilweise oder nicht unterstützt, dann ist eine PWA vielleicht geeignet. In diesem Fall muss der Entscheider selbst prüfen, ob die Unterstützung durch die Browser ausreichend ist.
- Wird mindestens ein Pflichtkriterium nicht unterstützt, dann ist eine PWA ungeeignet.

Abgrenzungskriterien haben entsprechend keinen Einfluss auf die Auswertung. Die Ausgabe des Resultats soll so gestaltet werden, dass an erster Stelle das Gesamtergebnis dargestellt wird. Anschließend wird nach Pflicht- und Kann-Kriterien getrennt aufgeführt, welche Funktionen in welchen Browsern vollständig, nur teilweise oder gar nicht unterstützt werden. Auf diese Weise kann der Nutzer bei einem „vielleicht“ Ergebnis prüfen, welche Funktionen genau nur teilweise oder nicht unterstützt werden.

Für die *Implementierung* der Leitfaden-App sollen zunächst mit HTML, JavaScript und CSS die standardmäßigen Webtechnologien genutzt werden. Zudem sollen noch die zwei Frameworks JQuery und JQuery Mobile zum Einsatz kommen. Ersteres vereinfacht die Programmierung in JavaScript und erhöht über seine Funktionen die Kompatibilität der App zwischen verschiedenen Browsern. Letzteres unterstützt bei der Erzeugung von SPAs, wodurch eine simple, statische Webseite entwickelt werden kann, die keinen komplexen Server im Hintergrund benötigt. Zudem werden die Oberflächen für Eingaben über Touchscreens optimiert, wodurch mobile Endgeräte unterstützt werden. Diese beiden Frameworks machen die Leitfaden-App zu verschiedenen Browsern und Endgeräten kompatibel und bilden somit die Grundlage für das Progressive Enhancement.

Die *grundlegende Struktur* der SPA besteht aus drei Ansichten, der Einleitungs-, Auswahl- und Ergebnisansicht. In der ersten wird der Begriff der PWA definiert und es wird beschrieben, wofür und wie die App verwendet werden kann. Auf der Auswahlseite kann der Nutzer dann die zu unterstützenden Browser und Funktionen definieren. Auf der Ergebnisseite werden entsprechend die Resultate der Auswertung dargestellt. Es wird immer nur eine der drei Ansichten gezeigt, wobei die Ansichten über Buttons gewechselt werden können. Die Grundlage für die Inhalte der Seiten bilden die Funktionen. Diese werden analog zum Regelwerk in einer Baumstruktur abgelegt. Zusätzlich werden unter den Funktionen noch die Technologien angehangen, über die die Funktionen umgesetzt werden können. Dieser Funktionsbaum wird

in einem JSON-Objekt definiert[114]. Über die Baumstruktur kann z.B. die Aufnahme von Videos über die Media Capture API mittels der ID `hardware.recording.video.api` angesprochen werden. Das JSON-Objekt wird dann verwendet, um Ansichten zu den Funktionen auf den Auswahl- und Ergebnisseiten aufzubauen. Zudem soll die Leitfaden-App so aufgebaut werden, dass die Texte der Oberfläche anhand verschiedener IDs aus einer separaten JavaScript-Datei gelesen werden. Auf diese Weise können durch eine Internationalisierung mehrere Sprachen in der App unterstützt werden. In der ersten Ausbaustufe soll die Leitfaden-App die Sprachen Deutsch und Englisch anbieten.

Die *Logiken für die Unterstützung* der einzelnen Funktionen werden ebenfalls abhängig von der JSON-Baumstruktur definiert und abgelegt. Hier wäre es möglich, die Logiken statisch im Code zu hinterlegen. Problematisch daran ist jedoch, dass die Leitfaden-App sehr schnell veraltet, weil sich die Unterstützungen der Funktionen in den Browsern von Zeit zu Zeit ändern. Die Lösung für dieses Problem bietet die Webseite `caniuse.com`.[115] Diese Seite bietet Informationen zu Browserkompatibilitäten für eine Vielzahl von Funktionen im Web. Die Kompatibilitätsdaten sind ebenfalls öffentlich in einem GitHub Repository[116] verfügbar und werden kontinuierlich aktualisiert. Für alle Funktionen, zu denen `caniuse.com` Daten besitzt, werden die Kompatibilitäten aus dem Repository ausgelesen und aufbereitet. Davon sind 20 der 38 Funktionen im Regelwerk betroffen. Auf diese Weise ist für über die Hälfte der Funktionen keine weitere Wartung nötig. Die Kompatibilitäten eines Großteils der restlichen Funktionen wird sich in Zukunft kaum ändern, weil sie entweder bereits komplett unterstützt werden oder konzeptionell vom W3C verworfen wurden.

Zuletzt sollen noch die im Kapitel 2.3.2 aufgeführten *PWA Technologien* integriert werden, durch die dann aus der Leitfaden-App eine PWA wird. So soll nach dem Vorgehen des Progressive Enhancements ein Service Worker eingebunden werden. Das bedeutet, dass der Service Worker nur bei Unterstützung durch den Browsers registriert wird und keine für den Kern der App relevanten Funktionen besitzt. Im Code wird dabei auf die in Abbildung 5 aufgeführten Ereignisse im Lebenszyklus eines Service Worker reagiert. So werden während der Installation alle benötigten Ressourcen im Hintergrund geladen und im Cache gespeichert. Bei der Aktivierung werden alte Caches geleert. Wird eine Ressource über das `fetch` Ereignis angefordert, wird dieses nach einer Strategie verarbeitet, die der „cache then network“ Strategie ähnelt. Dabei wird die Ressource, wenn vorhanden, zunächst aus dem Cache beant-

[114] Siehe Anhang N, Ressource „Baumstruktur der Regeln als JSON“.
[115] Vgl. Alexis Deveria (2018).
[116] Vgl. Alexis Deveria (o.J.).

wortet. Während die Hauptanwendung die Ressource bereits erhalten hat und weiterarbeitet, lädt der Service Worker die aktuellste Version der Ressource über das Internet. Auf diese Weise lädt die Anwendung sehr schnell und ist beim nächsten Mal trotzdem auf dem aktuellsten Stand. Die Kompatibilitätsdaten, die aus dem `caniuse.com` GitHub Repository geladen werden, werden zusätzlich bereits im Service Worker aufbereitet. Auf diese Weise können die Kompatibilitätsdaten sofort in der Hauptanwendung genutzt und müssen dort nicht erst aufwändig umgewandelt werden. Neben dem Service Worker wird ein Web App Manifest als JSON-Datei im Header der HTML-Datei eingebunden. Diese `manifest.json` enthält alle relevanten Attribute, die aktuell vom W3C zum Web App Manifest definiert wurden.[117] Darunter befindet sich eine Vielzahl von Icons in unterschiedlichen Auflösungen, sodass sich die verschiedenen Browser ein Icon in der passenden Größe aussuchen können. Der folgenden Abbildung 8 kann entnommen werden, dass die Kombination von Service Worker und Web App Manifest die Installation der Leitfaden-App ermöglicht.

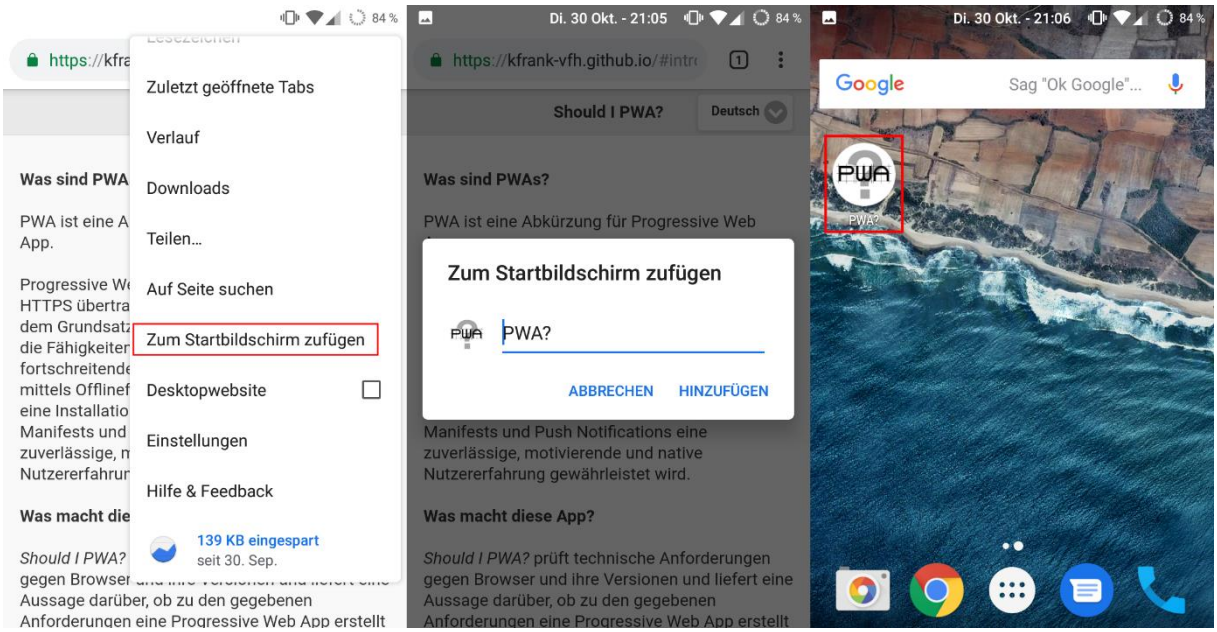

Abbildung 8: Installation der Leitfaden-App
Quelle: eigene Darstellung

Zu den Push Notifications als letzte der drei PWA Technologien existiert aktuell kein sinnvolles Einsatzszenario, weshalb diese für die Leitfaden-App nicht implementiert werden. Die App soll in Anlehnung an die „Can I Use“ Webseite auf den Namen „Should I PWA?“ (engl. für „Sollte ich PWA?“) getauft werden.

[117] Siehe Anhang N, Ressource „Manifest.json der Leitfaden-App“.

6.2.Evaluation der Leitfaden-App

Analog zum Regelwerk enthält die Leitfaden-App in ihrer ersten Version noch viel Optimierungspotenzial. Für eine wissenschaftliche Validität soll sie im Folgenden evaluiert werden.

6.2.1. Operationalisierung der Evaluation

Ziel der App war es, das Regelwerk interaktiv abzubilden, um somit dessen Nutzung und den Entscheidungsfindungsprozess zu beschleunigen. Ziel der Evaluation sollte es also sein, den tatsächlichen Nutzengewinn der Leitfaden-App gegenüber dem Regelwerk zu ermitteln. Um den zeitlichen Aspekt der beschleunigten Nutzung erfassen zu können, müssten sowohl Regelwerk als auch Leitfaden-App in einer realistischen Situation genutzt werden. Die die Probanden, die das Regelwerk nutzen, würden dabei als Kontrollgruppe dienen. Grundlage für eine solche Evaluation ist jedoch erstmal, dass die Leitfaden-App tatsächlich den Stand des Regelwerks widerspiegelt. Um das ursprüngliche Ziel der Leitfaden-App also prüfen zu können muss zunächst evaluiert werden, ob die App konform zum Regelwerk ist. Da letzteres bereits validiert wurde, kann davon ausgegangen werden, dass die App auch ein gewisses Maß an inhaltlicher Qualität besitzt, wenn sie konform zum Regelwerk ist. Eine gleichzeitige Prüfung beider Kriterien ist nicht möglich, da die Konformität den Nutzen beeinflussen würde. In einer ersten Validierung soll also zunächst die Konformität der Leitfaden-App zum Regelwerk ermittelt werden. Die Prüfung des tatsächlichen Nutzens der App muss aus zeitlichen Gründen außerhalb der Masterarbeit erfolgen.

Bei der ersten Untersuchung ist die App der Merkmalsträger, zu dem das komplexe Merkmal der Konformität geprüft werden soll. Für eine hundertprozentige Ermittlung der Konformität müssten zu allen 38 Funktionen die Kompatibilitätsdaten aller Browser geprüft werden. Der Nachteil hierbei ist der sehr hohe Aufwand, der eine sehr geringe Stichprobengröße zur Folge haben wird. Weil die Stichprobe bei der Evaluation des Regelwerks mit drei Experten bereits recht gering war, soll an dieser Stelle über eine quantitative Erhebung ein repräsentativeres Ergebnis angestrebt werden. Entsprechend soll die Konformität nicht zu 100%, sondern in einem angemessenen Maße ermittelt werden. Um während der Evaluation eine möglichst realistische Situation zu erzeugen, wird in einer User Story eine Reihe von Anforderungen vorgegeben. Der Proband soll sich in die von der Story beschriebenen Situation hineinversetzen und mit Hilfe der Leitfaden-App prüfen, ob zu den gegebenen Anforderungen eine PWA entwickelt werden kann oder nicht. Die Anforderungen der User Story werden dabei aus dem Regelwerk generiert. Kommt der Proband mit der Leitfaden-App zu demselben Ergebnis, das ausgehend vom Regelwerk zu erwarten wäre, dann ist die Konformität der App gegeben. Das

Problem hierbei ist, dass das Ergebnis von der Interpretation der Story vom Probanden abhängig ist. Dieses Problem soll dadurch vermieden werden, dass die User Story so formuliert wird, dass sie möglichst wenig Interpretationsspielraum bietet.

Um nun trotz der nicht-hundertprozentigen Abdeckung der Funktionen ein möglichst aussagekräftiges Ergebnis zu erhalten, soll zu jedem der drei Resultate, die die Leitfaden-App liefern kann, eine User Story entwickelt werden. Die Story zum Ergebnis „Eine PWA ist geeignet“ könnte z.B. wie folgt lauten:

> *Eine Videospielentwicklerin hat eine gute Idee für ein Mobile Game. Das Spiel soll eine 3D-Grafik besitzen, die über WebGL umgesetzt werden kann. Weil ihr die 2. Version noch unbekannt ist, will die Entwicklerin WebGL 1 nutzen. Der dreidimensionale Sound im Spiel soll über die Web Audio API erzeugt werden. Bei den Ressourcen des Spiels handelt es sich um komplexe Bild- und Audiodaten, die in einer IndexedDB gespeichert werden sollen. Zuletzt soll das Spiel auch über die neuen Service Worker offline lauffähig sein. Für die Android-Plattform sollen die Browser Chrome und Firefox ab Version 48 und für iOS der Safari Browser ab Version 11.1 unterstützt werden.*

In dieser Story sind mit WebGL 1, der Web Audio API, der IndexedDB in Version 1 oder 2 und dem Service Worker gleich mehrere Anforderungen enthalten. Service Worker werden ab Chrome 40, Firefox 41 und Safari 11.1 unterstützt. Die IndexedDB 2 funktioniert mit Chrome 48, Firefox 44 und Safari 10.1. Alle anderen Funktionen besitzen niedrigere Anforderungen. Die in der User Story vorgegebenen Browser sollten also die Anforderungen laut dem Regelwerk gerade unterstützen, wodurch der Proband zum Ergebnis kommen sollte, dass eine PWA geeignet ist.

Die Story zum Resultat „Eine PWA ist vielleicht geeignet“ könnte wie folgt lauten:

> *Eine Ärztin möchte einige Prozesse in ihrer Praxis automatisieren, um ihrem Praxispersonal Arbeit abzunehmen. Dazu sollen die Werte von medizinischen Geräten automatisch in das Praxisverwaltungssystem eingelesen werden. Medizinische Geräte stellen dazu in der Regel eine Bluetooth-Schnittstelle bereit. Die Ärztin hat die volle Kontrolle darüber, welche Browser auf den Endgeräten in ihrer Praxis installiert werden. Es genügt also, wenn diese Automatisierung in einem beliebigen Browser in der aktuellsten Version umgesetzt werden kann.*

Die einzige Anforderung aus dieser User Story ist die Bluetooth-Funktion. Nach dem Regelwerk ist Bluetooth im Chrome ab Version 53 nutzbar. Voraussetzung für dessen Nutzung sind eine obligatorische HTTPS-Verbindung und die Auslösung durch einen Nutzer. Durch diese Einschränkungen ist eine PWA nicht in jedem Fall geeignet. Der Nutzer sollte also nochmal prüfen, ob diese Einschränkungen in seinem konkreten Fall vertretbar sind.

Die dritte Story zum Ergebnis „Eine PWA ist ungeeignet“ könnte wie folgt lauten:

> *Ein Kreditkartenbetrüger benötigt mal wieder ein paar Kreditkartendaten, um an etwas Geld zu kommen. Die neue Funktion zum kontaktlosen Bezahlen in vielen neuen Geldkarten kommt ihm dazu sehr gelegen. Vielen Personen ist diese Funktion nicht bekannt, wodurch sie ihre Karten nicht schützen. Seine App soll automatisch über NFC (Near Field Communication) den NFC-Chip auf der Karte auslesen, wenn sich das Smartphone in der Nähe befindet. Für diese App soll der Chrome Browser verwendet werden, der in der aktuellsten Version 70 auf seinem Smartphone installiert ist.*

Die einzige Anforderung, die sich hierbei ergibt, ist die NFC-Funktionalität. NFC ist laut dem Regelwerk im Web generell nicht nutzbar. Der Proband sollte unabhängig davon, dass die aktuellste Version von Chrome genutzt werden kann, zu dem Ergebnis kommen, dass eine PWA ungeeignet ist.

Neben der Konformität soll zudem bereits in einer einfachen Form der *Nutzen der Leitfaden-App* ermittelt werden. Dazu soll zum einen mittels der Frage „Wie nützlich finden Sie die Leitfaden-App?“ die vom Probanden persönlich empfundene Nützlichkeit der App ermittelt werden. Zum anderen soll über die Frage „Würden Sie diese App nutzen, wenn sich die Gelegenheit dazu anbietet?“ die Einsatzbereitschaft erhoben werden. Die zweite Frage ist bewusst unter der Prämisse gestellt, dass eine entsprechende Situation zur Nutzung der App existiert, um auch eine valide Antwort von Personen zu erhalten, bei denen ansonsten keine solchen Anwendungsfälle im Beruf auftreten. Zuletzt sollen mittels der Frage „Sind Ihnen Fehler in der Leitfaden-App aufgefallen? Falls ja, welche?“ technische Fehler ermittelt werden. Der Grund für die Frage ist, dass Fehler recht wahrscheinlich sind, weil aus ressourcentechnischen Gründen keine umfassende Qualitätssicherung durchgeführt werden konnte.

Letztendlich sollen auch bei dieser Umfrage soziodemographische Daten erhoben werden, die potenziell Einfluss auf die zuvor erhobenen Merkmale haben können. Dazu zählen das Alter, der höchste Bildungsabschluss, die aktuelle berufliche Tätigkeit und die Erfahrung in der

Web- oder App-Entwicklung. Diese Merkmale verhalten sich dabei analog zu den soziodemographischen Merkmalen, die bei der Evaluation des Regelwerks erhoben wurden. (Siehe Tabelle 5: Merkmale der soziodemographischen Daten)

6.2.2. Auswahl und Begründung der Methodik

Mit diesen Vorüberlegungen soll nun die empirische Erhebungsmethode ausgewählt werden. Eine Methode, die theoretisch einsetzbar wäre, ist die *Beobachtung*. Diese könnte mit einem anderen Vorgehen kombiniert werden, in welchem der beobachtete Proband die Leitfaden-App nutzt. Die Beobachtung wird an dieser Stelle jedoch aus verschiedenen Gründen als ungeeignet erachtet. So bringt diese einen recht hohen Aufwand mit sich, was wiederum eine geringe Stichprobe zur Folge hat. Dies widerspricht dem gesetzten Ziel der hohen Repräsentanz. Außerdem liegt der Fokus einer Beobachtung eher bei der Usability und UX während der Nutzung der App. Somit wäre die Beobachtung eher dazu geeignet, den Nutzen der App im zweiten Schritt zu ermitteln. Die *Inhaltsanalyse* als zweite Methode wird ebenfalls als ungeeignet eingestuft, weil diese keine Probanden einbezieht und dadurch nicht die zuvor festgelegten Merkmale ermittelt werden können. Die Evaluation zur Leitfaden-App läuft also ebenfalls auf eine *Befragung* hinaus. Weil eine quantitative Erhebung mit einer möglichst großen Stichprobe angestrebt wird, fallen die teil- und nicht-standardisierten Befragungen weg. Von der persönlichen und schriftlichen Befragung ist über letztere eine größere Stichprobe möglich, da die schriftliche Befragung asynchron und ohne Beteiligung des Forschers durchgeführt werden kann. Hinzu kommt, dass der Proband während der Befragung ohnehin die Leitfaden-App über einen Computer oder ein mobiles Endgerät nutzen muss. Passend dazu bietet es sich an, die Befragung als computergestützte Online-Umfrage durchzuführen. Diese hat den Vorteil, dass viele Probanden in kurzer Zeit mit wenig Aufwand erreicht werden können und, dass die Daten problemlos digital exportiert und in einer Tabellenkalkulation ausgewertet werden können.

6.2.3. Durchführung der Evaluation

Die computergestützte Befragung soll mit Hilfe von Google Forms durchgeführt werden, einem kostenlosen Dienst zur Erstellung von Online-Umfragen. Die Umfrage wird in sechs Abschnitte unterteilt. Im ersten Abschnitt werden der Untersuchungsgegenstand und die Daten erläutert, die in der Umfrage erhoben werden. Zudem wird ein Hinweis auf die Anonymität der Datenerhebung und auf die geschätzte Dauer für die Teilnahme an der Umfrage gegeben. Die Abschnitte zwei bis vier enthalten die drei User Stories inklusive einer Aufforde-

rung, mit der Leitfaden-App zu prüfen, ob die in der Story definierten Anforderungen in einer PWA umgesetzt werden können. Entsprechend soll noch der Link zur Leitfaden-App eingefügt werden. Der Abschnitt fünf soll unter dem Titel „Sonstige Fragen“ die Nützlichkeit, die Einsatzbereitschaft und die technischen Fehler ermitteln. Zur Erhebung der Nützlichkeit wird eine Skala von 0 bis 10 angeboten. Zur Einsatzbereitschaft können die Antworten `Ja`, `Nein` und `Unsicher` gewählt werden. Die technischen Fehler können bei Bedarf in einem Freitextfeld beschrieben werden. Im letzten, sechsten Abschnitt werden die verbleibenden soziodemographischen Daten erhoben. Das Alter kann dabei über mehrere Optionen angegeben werden, die Intervalle im Abstand von 10 Jahren enthalten. Der Grund für die Intervalle ist, dass mit dem genauen Alter theoretisch eine Entanonymisierung der Datensätze möglich wäre. Der höchste Bildungsabschluss soll analog zur Evaluation des Regelwerks über mehrere Optionen erhoben werden. Die berufliche Tätigkeit wird über ein Freitextfeld erfasst. Der Grund dafür ist, dass eine nachträgliche Gruppierung der Berufe effizienter ist, als eine Liste von allen möglichen Berufen zu erstellen. Zur Erfahrung in der Web- oder App-Entwicklung in Jahren wird ein genauer Wert über ein Freitextfeld erhoben. Die vollständige Umfrage kann der digitalen Form der Masterarbeit entnommen werden.[118]

Die *Verteilung der Umfrage* soll per E-Mail erfolgten. Dabei werden die Informationen, die auch im ersten, einleitenden Abschnitt der Online-Umfrage aufgeführt wurden, enthalten sein. Zusätzlich soll in der E-Mail angemerkt werden, dass die Umfrage in deutscher Sprache und die App in den Sprachen Deutsch und Englisch genutzt werden kann. Es sollen E-Mails an die Studenten und Mitarbeiter der Technischen Hochschule Brandenburg und an die Mitarbeiter der data experts gmbh versendet werden. Die entsprechenden E-Mails können der digitalen Version der Masterarbeit entnommen werden.[119] Die Umfrage war vom 26.10.218 bis zum 04.11.2018 aktiv. In diesem Zeitraum sind 55 Antworten eingegangen. Der Endstand der Daten wurde dann zur Evaluation der Leitfaden-App genutzt.

6.2.4. Ergebnisse und Interpretation der Evaluation

6.2.4.1. Auswertung der Konformität

Die von Google Forms exportierten CSV-Daten wurden zunächst in eine Excel-Datei importiert und anschließend in einem weiteren Tabellenblatt angepasst. Dabei wurden die Daten so formatiert und gekürzt, dass sie im Folgenden besser ausgewertet werden können. In diesem Schritt wurden z.B. Textwerte in Datums- oder Zahlwerte umgewandelt oder gleiche Berufe

[118] Siehe Anhang N, Dokument „Druckversion der Umfrage“.
[119] Siehe Anhang N, Dokument „Mailverkehr zur Umfrage“.

mit unterschiedlicher Schreibweise vereinheitlicht. Die Umwandlung von Werten, die sich etwas stärker von ihren Ausgangswerten unterscheiden, wurden durch einen Kommentar in der entsprechenden Zelle begründet.

Im dritten Tabellenblatt erfolgt nun die *Auswertung der Konformität* über die User Stories. Jeder Datensatz enthält zu jeder User Story die Antwort darauf, ob eine PWA geeignet ist oder nicht. Die Antworten wurden in die Werte `ja`, `nein` und `vielleicht` umgewandelt. Abhängig von der erwarteten Antwort zu jeder Story kann nun ermittelt werden, ob die Antwort des Probanden erwartungskonform ist oder nicht. Der Proband kann somit die Fragen zu null, eins, zwei oder allen drei Stories erwartungskonform beantworten. Die Anteile der Antworten und deren Konformitäten kann der folgenden Abbildung 9 entnommen werden.

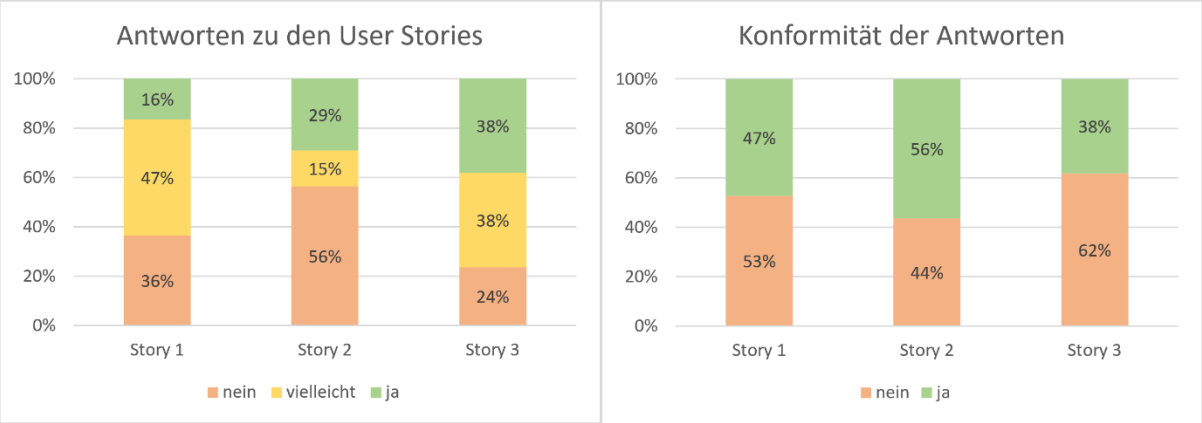

Abbildung 9: Anteile und Konformität der User Story Antworten
Quelle: Siehe Anhang N, Dokument „Auswertung der Umfrage in Excel“

Bei der ersten Story wurde die Antwort `vielleicht`, bei der zweiten Story die Antwort `nein` und bei der dritten Story die Antwort `ja` erwartet. Entsprechend sind bei den drei Stories 47%, 56% und 38% der Antworten konform. Im Durchschnitt haben die Probanden also anhand der Leitfaden-App mit 47% nicht mal zur Hälfte der Fälle das Ergebnis erhalten, was über das Regelwerk zu vermuten wäre. Die erwartete Antwort macht zwar in allen drei Fällen den größten Anteil aus, sie hebt sich jedoch nicht weit genug ab, um davon ausgehen zu können, dass die Leitfaden-App konform zum Regelwerk ist. An dieser Stelle muss also untersucht werden, wie die unerwarteten Antworten entstanden sein könnten. Da die Eingaben in die Leitfaden-App nicht erfasst wurde und auch nicht ohne weiteres dem Datensatz der Umfrage zugeordnet werden können, sind an dieser Stelle nur Vermutungen möglich.

Die *erste User Story* befasst sich mit der Ärztin, die über Bluetooth ihr Praxisverwaltungssystem an medizinische Geräte anschließen will. Das einzige Pflichtkriterium hier ist die Bluetooth-Funktionalität. In Bezug auf die Kompatibilität war nur vorgegeben, dass die PWA in einem beliebigen Browser in der aktuellsten Version funktionieren soll. Ein Test hat gezeigt,

dass die Probanden bei dieser Story mitdenken müssen und nicht einfach nur die Daten eingeben können. Grund ist, dass Bedingungen zu den Browsern und Versionen mit einem logischen `und` verknüpft sind. Wählt der Proband also alle Browser aus und wertet die Kompatibilität aus, dann erhält er zunächst die Antwort `nein`, weil Bluetooth entsprechend nur im Chrome verfügbar ist. Hier muss der Proband seine Eingaben anschließend anpassen und nur Chrome als unterstützten Browser auswählen, um das Ergebnis `vielleicht` zu erhalten. Wenn der Proband davon ausgeht, dass die Bedingungen zu den Browsern und Versionen mit einem logischen `oder` verknüpft sind, dann würde er die erste Antwort in der Umfrage eingeben, was die 36% der `nein`-Antworten erklären würde. Die restlichen 16% der `ja`-Antworten können nicht logisch nachvollzogen werden.

Die *zweite User Story* befasst sich mit dem Kreditkartenbetrüger, der mittels NFC die Daten von Kreditkarten auslesen will. Das einzige Pflichtkriterium hier ist die NFC-Funktionalität. Diese sollte mit Chrome in der Version 70 kompatibel sein. Ein Test hat gezeigt, dass die Leitfaden-App auch tatsächlich `nein` als Antwort liefert, wenn Chrome 70 als Browser und NFC als Pflichtfunktion eingegeben werden. Wird NFC jedoch nur als Kann-Kriterium eingegeben, dann liefert die Leitfaden-App `vielleicht` als Antwort, was 15% der unerwarteten Antworten erklären könnte. Die restlichen 29% der `ja`-Antworten können nicht nachvollzogen werden, weil hierzu NFC als einziges Kriterium vollkommen ignoriert werden müsste.

Die *dritte User Story* enthält die Videospielentwicklerin, die über verschiedene Funktionen ein Mobile Game entwickeln will. Die Story enthält vier Pflichtfunktionen, die von den drei vorgegebenen Browsern vollständig unterstützt werden sollten. Ein Test hat gezeigt, dass sich die Leitfaden-App hier in einem Fall tatsächlich nicht konform zum Regelwerk verhält. Ursache ist die IndexedDB, zu der nicht vorgegeben wurde, ob die Version 1 oder 2 verwendet werden sollte. Laut dem Regelwerk sollten beide Versionen in einer PWA unterstützt werden, die Leitfaden-App kommt zur IndexedDB 2 jedoch zu einem anderen Ergebnis. Grund dafür ist, dass die Kompatibilität zur IndexedDB anhand der Daten der Webseite `caniuse.com` ermittelt wird. Laut `caniuse.com` wird die IndexedDB 2 von den Browsern Firefox und Chrome in der Version 48 mit der Begründung nur partiell unterstützt, dass einige in der Spezifikation vorgegebenen Methoden fehlen.[120] Somit würden alle Probanden, die die zweite anstatt der ersten Version als Kriterium angegeben haben, zu dem Ergebnis kommen, dass eine PWA nur geeignet sein könnte, es jedoch nicht in jedem Fall ist. Dies würde 38% der `vielleicht`-Antworten erklären. Ein `nein` als Antwort wäre nur möglich, wenn andere

[120] Vgl. Alexis Deveria (2018), Suche nach „IndexedDB 2.0“.

Daten als vorgegeben für die Prüfung genutzt wurden. Somit können 24% der nicht-erwartungskonformen Antworten nicht erklärt werden.

Auf Basis dieser Überlegungen können nun einige unerwartete Antworten mit zu den erwartungskonformen Antworten gerechnet werden, wodurch sich eine korrigierte Konformität ergibt, deren Verteilung der folgenden Abbildung 10 entnommen werden kann.

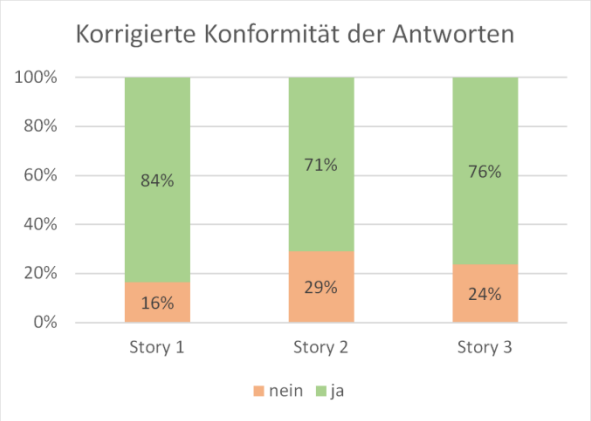

Abbildung 10: Korrigierte Konformität der Antworten
Quelle: Siehe Anhang N, Dokument „Auswertung der Umfrage in Excel“

Die korrigierte Konformität soll an dieser Stelle keine Schönrechnung der Statistik sein, sondern eher als Maß für die Antworten verstanden werden, die sich durch logische Schlussfolgerungen erklären lassen. Der Durchschnitt der korrigierten Konformität beträgt 77%. Unter Berücksichtigung der Tatsache, dass sie anhand von Vermutungen korrigiert wurde, sind weiterhin zu viele unerwartete Antworten enthalten.

6.2.4.2. Untersuchung von Zusammenhängen zur Konformität

Auf Grund der unerwarteten Antworten soll explorativ untersucht werden, ob sich diese ggfs. anhand von *Zusammenhängen der Konformität* zu anderen Variablen erklären lassen. Neben den Antworten zu den User Stories liegen noch Daten zu Datum und Zeitpunkt der Erhebung, zur Nützlichkeit, Einsatzbereitschaft und technischen Fehlern der App und soziodemographische Daten der Probanden vor. Ein potenzieller Zusammenhang könnte dabei zum Erhebungszeitpunkt oder den soziodemographischen Daten bestehen. Über verschiedene statistische Methoden, die im Kapitel 3.3.1.2 Bivariate Verfahren erläutert wurden, können nun Korrelationen zur Konformität berechnet werden.

Die errechneten Werte können der folgenden Tabelle 6 entnommen werden.

Korrelation zu	Errechnet durch	normal	korrigiert
Datum	Rangkorrelation (Spearman)	0,034	-0,146
Uhrzeit	Korrelationskoeffizient (Pearson)	-0,117	-0,261
Alter	Rangkorrelation (Spearman)	0,061	0,199
Abschluss	Rangkorrelation (Spearman)	0,076	0,275
Tätigkeit	Kontingenzkoeffizient	0,405	0,362
Erfahrung	Korrelationskoeffizient (Pearson)	0,131	-0,035

Tabelle 6: Korrelationen zur normalen und korrigierten Konformität
Quelle: Siehe Anhang N, Dokument „Auswertung der Umfrage in Excel“

Alle Werte sind so normiert, dass sie im Intervall zwischen 0 und ±1 liegen. Auf dem ersten Blick ist gut erkennbar, dass keiner der Werte besonders hoch ist. Es existiert also kein eindeutiger Zusammenhang zwischen der Konformität und einer anderen Variable. Leicht erhöhte Werte würden jedoch dafürsprechen, dass bestimmte Variablen zumindest einen geringen Einfluss auf die Konformität besitzen. Das Datum hat erwartungsgemäß keinen Einfluss auf die Konformität. Die Uhrzeit zeigt sowohl bei der normalen als auch bei der korrigierten Konformität eine leicht erhöhte, negative Korrelation. Sollte dieser Zusammenhang tatsächlich existieren, würde das bedeutet, dass mit fortschreitender Stunde immer weniger erwartungskonforme Antworten von den Probanden gegeben wurden. Das könnte ein Hinweis darauf sein, dass die Probanden vormittags mit einer größeren Konzentration an der Umfrage teilgenommen haben. Während Alter und Abschluss in Bezug auf die normale Konformität kaum einen Einfluss zeigen, besitzen sie eine leichte Korrelation mit der korrigierten Konformität. Das würde bedeuten, dass mit höherem Alter und einem höheren Abschluss zunehmend erwartungskonformere Antworten gegeben wurden. Gleichzeitig würde diese Korrelation die zuvor aufgestellten Vermutungen zur Korrektur der Konformität bestätigen. Interessant hierbei ist, dass die Konformität mit dem Alter korreliert, nicht jedoch mit der Erfahrung in der Web- und App-Entwicklung. Das könnte bedeuten, dass die Nutzbarkeit der Leitfaden-App unabhängig davon ist, ob sie von einer Person mit viel oder keiner Erfahrung in der Entwicklung genutzt wird. Der größte Zusammenhang ergibt sich zur Tätigkeit der Probanden. Da dieser zu nominal-skalierten Werten über den Kontingenzkoeffizienten berechnet wurde, wird die Konformität in der folgenden Abbildung 11 nach Tätigkeitsgruppe aufgeschlüsselt.

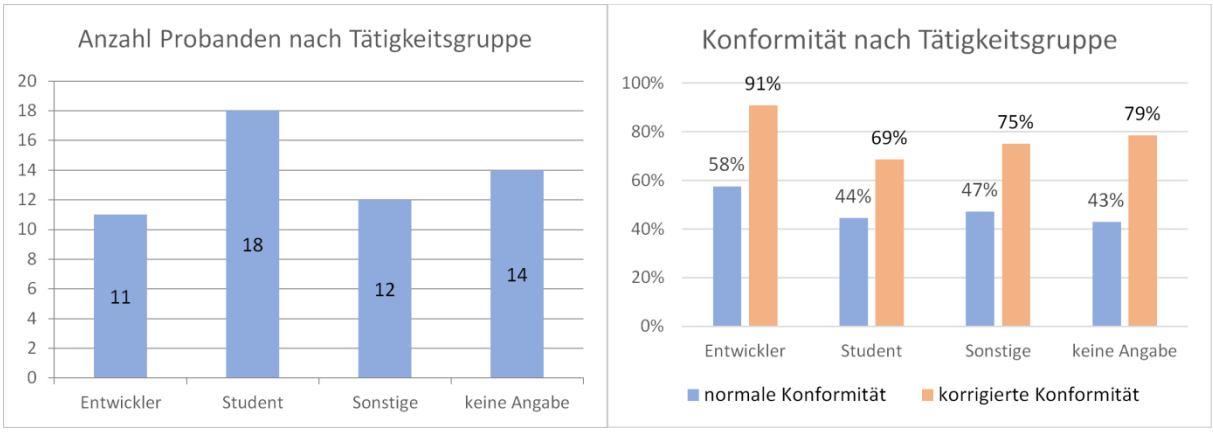

Abbildung 11: Konformität nach Tätigkeitsgruppe
Quelle: Siehe Anhang N, Dokument „Auswertung der Umfrage in Excel“

In diesem Diagramm ist erkennbar, dass die Probanden, die als Entwickler arbeiten, wesentlich erwartungskonformere Antworten geliefert haben. Während sich die restlichen Gruppen um und unter den Durchschnittswerten 47% für die normale und 77% für die korrigierte Konformität einpegeln, liegen die Entwickler bei 58% für die normale und sogar bei 91% für die korrigierte Konformität. Dieses Ergebnis steht nicht unbedingt im Widerspruch zur vorherigen Aussage, dass die Konformität unabhängig von der Erfahrung in der Web- und App-Entwicklung ist. So kann es z.B. sein, dass die Erfahrung zwar keinen Einfluss hat, wohl aber das technische Verständnis, das für einen Beruf in der Softwareentwicklung notwendig ist.

6.2.4.3. Untersuchung weiterer Zusammenhänge

Neben den bisherigen Untersuchungen können aus dem Datensatz keine weiteren Erkenntnisse zur Konformität gewonnen werden, es können aber wohl die Zusammenhänge zu weiteren Merkmalen untersucht werden. Interessant sind hier die Nützlichkeit, Einsatzbereitschaft und technischen Fehler, die neben den Antworten zu den User Stories erhoben wurden. Analog zu den bisherigen Auswertungen sollen nun die Korrelationen dieser Merkmale und in Bezug auf die soziodemographischen Daten explorativ geprüft werden. Die errechneten Werte können der Tabelle 7 auf der nächsten Seite entnommen werden.

Korrelation zwischen		Errechnet durch	Wert
Nützlichkeit	Einsatzbereitschaft	Rangkorrelation (Spearman)	0,565
Nützlichkeit	Anmerkung	Punktbiseriale Korrelation	-0,044
Nützlichkeit	Fehler	Punktbiseriale Korrelation	-0,206
Nützlichkeit	kein Fehler	Punktbiseriale Korrelation	0,086
Einsatzbereitschaft	Anmerkung	Kontingenzkoeffizient	0,199
Einsatzbereitschaft	Fehler	Kontingenzkoeffizient	0,284
Einsatzbereitschaft	kein Fehler	Kontingenzkoeffizient	0,087
Nützlichkeit	Alter	Rangkorrelation (Spearman)	-0,207
Einsatzbereitschaft	Alter	Rangkorrelation (Spearman)	0,003
Nützlichkeit	Abschluss	Rangkorrelation (Spearman)	-0,350
Einsatzbereitschaft	Abschluss	Rangkorrelation (Spearman)	-0,025
Nützlichkeit	Tätigkeit	Kontingenzkoeffizient	0,735
Einsatzbereitschaft	Tätigkeit	Kontingenzkoeffizient	0,486
Nützlichkeit	Erfahrung	Korrelationskoeffizient (Pearson)	0,193
Einsatzbereitschaft	Erfahrung	Korrelationskoeffizient (Pearson)	0,328

Tabelle 7: Korrelationen zwischen weiteren Merkmalen
Quelle: Siehe Anhang N, Dokument „Auswertung der Umfrage in Excel“

Nicht verwunderlich ist, dass eine positive Korrelation zwischen Nützlichkeit und Einsatzbereitschaft besteht. Der Wert ist entgegen der Erwartung sogar etwas klein ausgefallen, es sollte jedoch ersichtlich sein, dass Probanden, die die Leitfaden-App nützlich finden, sie auch eher einsetzen würden. Die Untersuchung der Zusammenhänge auf technische Fehler wurde aufgeteilt, weil das Feld in der Umfrage nicht nur für tatsächliche Fehler, sondern auch für Optimierungsvorschläge und sonstige Anmerkungen genutzt wurde. Aus diesem Grund enthält das Merkmal „Anmerkungen“ in der obigen Tabelle alle Eingaben im Feld, das Merkmal „Fehler“ die tatsächlichen Fehler und das Merkmal „kein Fehler“ entsprechend die sonstigen Anmerkungen. Die punktbiseriale Korrelation wird dabei anhand des Vorhandenseins einer Anmerkung berechnet. Hierbei ist erkennbar, dass die sonstigen Anmerkungen anscheinend keinen Einfluss auf die Nützlichkeit haben, zu den Fehlern jedoch mit -0,206 eine leichte, negative Korrelation besteht. Das kann so interpretiert werden, dass sich das Vorhandensein von Fehlern negativ auf die von den Probanden empfundene Nützlichkeit der Leitfaden-App ausgewirkt hat. Ebenfalls erkennbar ist, dass mit dem Wert 0,284 vor allem die Fehler unter den Anmerkungen einen Einfluss auf die Einsatzbereitschaft haben. Der auf Grund der Nominalskalierung der Daten errechnete Kontingenzkoeffizient gibt dabei nicht die Richtung des Zusammenhangs an, sondern nur dessen Eindeutigkeit. Deswegen wird die Einsatzbereitschaft in der folgenden Abbildung 12 nach den Anmerkungen aufgeschlüsselt dargestellt.

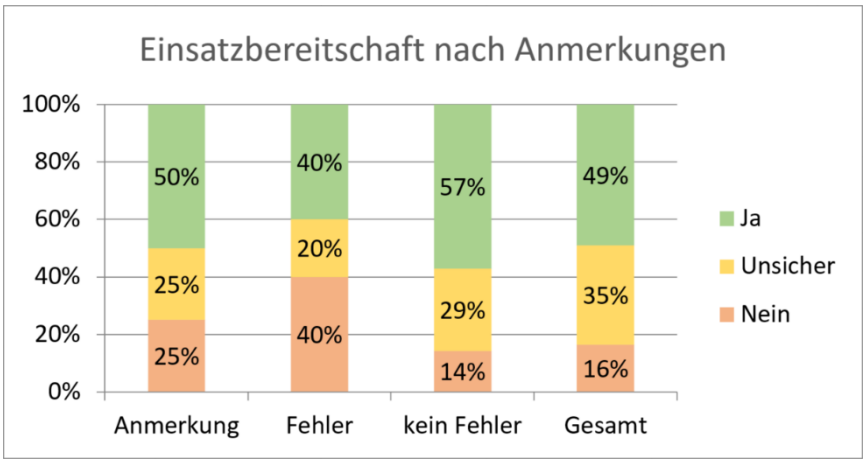

Abbildung 12: Einsatzbereitschaft nach Anmerkungen
Quelle: Siehe Anhang N, Dokument „Auswertung der Umfrage in Excel“

Der Balken rechts zum Wert „Gesamt“ gibt die Anteile der Angaben zur Einsatzbereitschaft über alle Daten wieder und bildet somit den durchschnittlichen Richtwert. Hier ist nun erkennbar, dass Probanden, die einen Fehler entdeckt haben, überdurchschnittlich häufig nicht zum Einsatz der Leitfaden-App bereit sind. Analog zur Nützlichkeit wirkt sich also das Vorhandensein von Fehlern negativ auf die Einsatzbereitschaft aus. Ebenso ist erkennbar, dass überdurchschnittlich viele Probanden, die einen Optimierungsvorschlag eingereicht haben, zur Nutzung der App bereit sind. Das kann erklärt werden, indem die Richtung von Ursache und Wirkung umgedreht wird: Probanden, die dazu bereit waren, sich mit der Umfrage und der Leitfaden-App auseinanderzusetzen, haben sich intensiver damit beschäftigt und eher Optimierungsvorschläge unterbreitet.

Da die Werte zur Einsatzbereitschaft mit 0,003 und -0,025 in Bezug auf Alter und Abschluss nahezu null sind, besteht entsprechend kein Zusammenhang. Die Nützlichkeit hingegen weist mit -0,207 und -0,35 eine leicht negative Korrelation zu Alter und Abschluss auf. Das bedeutet, dass die Leitfaden-App als unnützer empfunden wurde, je höher das Alter oder der akademische Abschluss des Probanden ist. Ein Grund dafür könnte sein, dass die App aus einem wissenschaftlichen Blickwinkel, den die Probanden mit hohem Abschluss haben, keinen hohen Nutzen mit sich bringt. Um diese Hypothese jedoch zu bestätigen und die Ursachen dahinter zu ergründen, müssten weitere Evaluationen vorgenommen werden. Mit den Werten 0,735 und 0,486 besteht die größte Korrelation wieder zwischen der Tätigkeit und der Nützlichkeit bzw. Einsatzbereitschaft. Auf Grund der Nominalskalierung werden die beiden Merkmale in der folgenden Abbildung 13 nach der Tätigkeit aufgeschlüsselt.

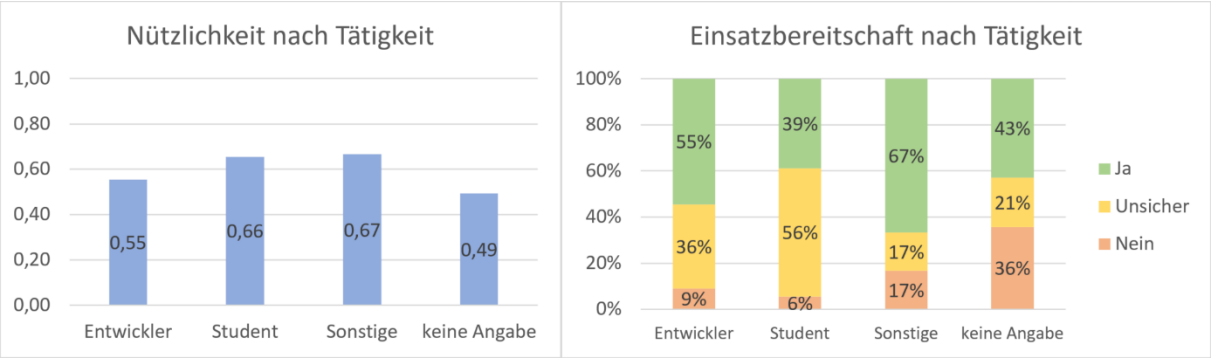

Abbildung 13: Nützlichkeit und Einsatzbereitschaft nach Tätigkeit
Quelle: Siehe Anhang N, Dokument „Auswertung der Umfrage in Excel“

Auf der linken Seite sind die durchschnittlichen Werte zur Nützlichkeit je Tätigkeitsgruppe aufgeführt. Insgesamt liegt die Nützlichkeit leicht im oberen Mittelfeld. Vor allem Studenten und die Probanden mit sonstigen Tätigkeiten finden die Leitfaden-App überdurchschnittlich nützlich. Die Entwickler hingegen scheinen einen eher pragmatischen Blick zu besitzen und schätzen den Nutzen nicht so hoch ein. In Bezug auf die Einsatzbereitschaft machen die Studenten die Gruppe aus, die sich am unsichersten sind, ob sie die App verwenden würden. Die Personen mit sonstigen Tätigkeiten hingegen würden die Leitfaden-App am ehesten einsetzen. Das könnte daran liegen, dass diese Personen auf Grund ihrer Tätigkeit kein technisches Vorwissen besitzen und entsprechend für eine solche Problemstellung bei der App einen guten Anlaufpunkt sehen. Die Probanden, die ihre Tätigkeit nicht angaben, haben sowohl die Nützlichkeit als auch die Einsatzbereitschaft am negativsten bewertet. Der Grund für diese negativen Werte kann an dieser Stelle nicht erschlossen werden, es liegt jedoch die Vermutung nahe, dass die Probanden ihre Anonymität wegen der schlechten Bewertung absichern wollten. Zuletzt bestehen mit den Werten 0,193 und 0,328 noch leichte Korrelationen zwischen Nützlichkeit bzw. Einsatzbereitschaft und der Erfahrung. Das bedeutet, dass Probanden mit mehr Erfahrung in der Web- und App-Entwicklung die Leitfaden-App nützlicher finden und eher einsetzen würden.

6.2.4.4. Bewertung der Ergebnisse und Methodik

Die Konformität der Antworten über alle drei User Stories beträgt nur 47%. Selbst anhand von Vermutungen basierend auf logischen Rückschlüssen konnten lediglich 77% der Antworten erklärt werden. Daraus lässt sich folgern, dass die Konformität von weiteren Faktoren beeinflusst wurde, die aus den untersuchten Korrelationen hervorgehen. An dieser Stelle soll betont werden, dass die Korrelationswerte überwiegend sehr gering waren, weshalb die im Folgenden aufgestellten Vermutungen mit Vorsicht zu genießen sind. Da die Antworten zu früheren Zeitpunkten eine höhere Konformität aufweisen, könnte mangelnde Konzentration

bei den Probanden Fehlerwerte erzeugt haben. Zudem hat sich gezeigt, dass ein grundlegendes, technisches Verständnis für die Nutzung der App nötig war. Die Abwesenheit solchen Wissens kann die Ursache für Fehlerwerte sein. Zuletzt scheint es auch sehr wahrscheinlich, dass die Wahl der Online-Umfrage als alleinige Erhebungsmethode suboptimal war. Der Nachteil der Online-Umfrage ist nämlich, dass nicht prüfbar ist, war die Umfrage ausgefüllt hat und ob auch tatsächlich korrekte Antworten gegeben wurden. Zudem haben einige Probanden über inoffizielle Wege das Feedback gegeben, dass sie nicht verstanden haben, was in der Umfrage zu tun ist. Es ist also möglich, dass die Teilnehmer in diesen Fällen die Fragen zu den User Stories ohne Nutzung der Leitfaden-App aus dem eigenen Wissen heraus beantwortet haben, was wiederum die Umfrageergebnisse verfälscht. Die Verwendung von User Stories zur Erzeugung einer möglichst realistischen Situation sollte in diesem Rahmen legitim gewesen sein, in einem Pretest hätte jedoch zuvor die Verständlichkeit der Umfrage geprüft werden können. Auf Grund der Verständnisprobleme hätte die Umfrage zudem mit einer Beobachtung kombiniert werden sollen. Da der Forscher in diesem Fall anwesend wäre, hätte der Proband ihm Fragen stellen können. Zudem hätte der Forscher prüfen können, ob und wie die Fragen beantwortet wurden. Dadurch ließen sich dann auch besser potenzielle Fehlerquellen erschließen. Bei der Wahl der Methodik hätte der Fokus also zugunsten der Qualität der Ergebnisse weniger auf der Repräsentanz liegen sollen.

Neben denen zur Konformität ergeben sich aus der Evaluation noch weitere Erkenntnisse, z.B. zur Zielgruppe der Leitfaden-App. Die initial vorgesehene Zielgruppe bestand aus Entscheidungsträgern, die prüfen wollen, ob zu einer Reihe von Anforderungen eine PWA entwickelt werden kann. Im Rahmen der Umfrage sind diese Entscheidungsträger am ehesten mit den Entwicklern gleichzusetzen. Diese wären auf Grund des nötigen, technischen Vorwissens gut als Zielgruppe geeignet. Die Auswertung zeigt, dass über die Hälfte der Entwickler die App nutzen würden, sie von ihnen jedoch nicht allzu nützlich eingeschätzt wird. Da die Nützlichkeit die Einsatzbereitschaft und das Vorhandensein von Fehlern beide zuvor genannten Eigenschaften beeinflusst, sollte in weiteren Schritten der Fokus auf die Korrektur von Fehlern gelegt werden. Abgesehen von den Entwicklern ergibt sich anhand der Auswertung noch eine weitere Zielgruppe. Sowohl Nützlichkeit aus auch Einsatzbereitschaft waren bei Probanden mit sonstigen, nicht-technischen Tätigkeiten überdurchschnittlich hoch. Hierbei sollte jedoch angemerkt werden, dass die Frage zur Einsatzbereitschaft unter der Prämisse gestellt wurde, dass eine Situation existiert, in der die Leitfaden-App genutzt werden kann. Weil solche Anwendungsfälle bei nicht-technischen Berufen eher selten gegeben sind, sollten diese Probanden aus der Zielgruppe exkludiert werden.

Zuletzt hat die Auswertung noch Erkenntnisse zur Usability hervorgebracht. So wurde das Feld für technische Fehler in der Umfrage für Optimierungsvorschläge zur Nutzbarkeit der App verwendet. Eine naheliegende Schlussfolgerung ist also, dass immer ein Feld für allgemeine Anmerkungen in einer Umfrage vorhanden sein sollte. Zusätzlich wird im Zusammenhang mit anderen Korrelationen deutlich, dass die Usability der App eine wichtige Rolle für die empfundene Nützlichkeit und Einsatzbereitschaft spielt. Das spricht für die bereits vorgesehene, zweite Evaluation, durch die die Nutzbarkeit der App verbessert werden soll. Auf diese Weise können dann eine höhere Einsatzbereitschaft und der Erfolg der Leitfaden-App gesichert werden.

7. Fazit und Ausblick

Ziel dieser Masterarbeit war es, den PWA-Begriff zu definieren und deren technische Grenzen zu untersuchen. Auf Basis der erlangten Erkenntnisse sollte dann ein leicht zugänglicher Leitfaden zur Nutzung von PWAs entwickelt werden. Die PWA-Definition wurde über eine Recherche sowohl in Deutsch als auch in Englisch als internationale Version erstellt. Anhand eines detaillierten Vergleichs von nativen und Web Apps wurden dann die Funktionen und Grenzen von PWAs abgeleitet und in einem schriftlichen Regelwerk festgehalten. Dessen Korrektheit und Vollständigkeit wurde mit Hilfe einer Kombination von Peer Review und Interview mit drei Experten abgesichert, die Evaluation ist jedoch nicht sehr repräsentativ. Zuletzt wurde das schriftliche Regelwerk zur Gewährleistung des leichten Zugangs und einer besseren Nutzbarkeit als PWA umgesetzt. Eine Untersuchung des Nutzens der Leitfaden-App konnte aus zeitlichen Gründen nicht im Rahmen der Masterarbeit durchgeführt werden. Die Wahl und Umsetzung der Online-Umfrage zur Evaluation der Konformität der Leitfaden-App zum Regelwerk hat sich ebenfalls als suboptimal herausgestellt. Aus diesem Grund konnte die Konformität der App nicht bestätigt, sondern lediglich Hypothesen aufgestellt werden. Da die Leitfaden-App nur eine verbesserte Form des schriftlichen Regelwerks ist, wurden die Ziele der Masterarbeit aus Sicht des Autors nichtsdestotrotz in einem zufriedenstellenden, aber nicht vollständigen Maße erfüllt.

Somit gibt es einige Tätigkeiten, die an diese Arbeit angeknüpft werden können. Da scheinbar noch keine standardisierte Definition für PWAs existiert, könnte die Definition dieser Masterarbeit etabliert werden. Auf Grund der fehlenden Prüfung der Definition müsste sie zunächst von weiteren Wissenschaftlern oder Experten validiert und anschließend z.B. durch das Deutsche Institut für Normung aufgegriffen werden. Das Regelwerk könnte zur Erhöhung der Repräsentanz des Ergebnisses durch weitere Experten geprüft werden, wie bspw. Entwickler verschiedener Browserhersteller oder anderer Unternehmen im Bereich der Webentwicklung. Zuletzt ist für eine erfolgreiche Nutzung die Verbesserung der Leitfaden-App nötig. Weil die Konformität der App nicht bestätigt werden konnte, sollten die bisher bekannten Mängel beseitigt und die Konformität anhand einer besseren Methodik ermittelt werden. Zudem ist noch der zweite Schritt offen, in dem die Leitfaden-App in Bezug auf die Usability geprüft und optimiert wird.

Literaturverzeichnis

Alexis Deveria (2018): Can I Use, veröffentlicht am 28.10.2018, zuletzt abgerufen am 03.11.2018, https://caniuse.com/

Alexis Deveria (o.J.): Can I Use GitHub Repository, kein Veröffentlichungsdatum angegeben, zuletzt abgerufen am 03.11.2018, https://github.com/fyrd/caniuse

Apache Cordova (o.J.): Architecture, kein Veröffentlichungsdatum angegeben, zuletzt abgerufen am 12.07.2018, https://cordova.apache.org/docs/en/latest/guide/overview/

Archibald, Jake (2014): The offline cookbook, veröffentlicht am 09.12.2014, zuletzt abgerufen am 20.70.2018, https://jakearchibald.com/2014/offline-cookbook/

Ates, Faruk u.a. (o.J.): Modenizr - the feature detection library for HTML5/CSS3, kein Veröffentlichungsdatum angegeben, zuletzt abgerufen am 13.07.2018, https://modernizr.com/

Berger-Grabner, Doris (2016): Wissenschaftliches Arbeiten in den Wirtschafts- und Sozialwissenschaften, 3., aktualisierte und erweiterte Auflage, Wiesbaden, 2016

Braun, Herbert (2017): Progressive Web-Apps entwickeln. In: c't-Redaktion (Hrsg.): c't Webdesign: Entwicklung - Performance - SEO - Content Management, o.O., 13.06.2017, S. 30-34.

Braun, Herbert (2018): Push-Web - Push-Nachrichten im Browser empfangen, Teil 1. In: c't-Redaktion (Hrsg.): c't - Magazin für Computertechnik / Know-how, o.O., Heft 9 / 2018, S. 172-177

Buber, Renate & Holzmüller, Hartmut H. (Hrsg.) (2009): Qualitative Marktforschung, 2., überarbeitete Auflage, Wiesbaden, 2009

Castro, Elizabeth & Hyslop, Bruce (2014): Praxiskurs HTML5 & CSS3 - Professionelle Webseiten von Anfang an, 3., aktualisierte und erweiterte Auflage, Heidelberg, 2014

Carlyle, Thomas (1893): Über Helden, Heldenverehrung und das Heldenthümliche in der Geschichte, 2. Auflage, Heidelberg, 1893

Chrome Platform Status (2018): Service Worker, veröffentlicht am 26.04.2018, zuletzt abgerufen am 19.07.2018, https://www.chromestatus.com/feature/6561526227927040

Dembowski, Klaus (2017): Smartphone- und Tablet-Hacks, 1. Auflage, Heidelberg, 2017

Dormann, Carsten (2017): Parametrische Statistik, 2., überarbeitete und erweiterte Auflage, Berlin, 2017

Egewardt, Rainer (2002): Das PC-Wissen für IT-Berufe, 2., überarbeitete und erweiterte Auflage, Wiesbaden, 2002

Fablet, Youenn (2018): Workers at Your Service, veröffentlicht am 07.02.2018, zuletzt abgerufen am 19.07.2018, https://webkit.org/blog/8090/workers-at-your-service/

G2 Crowd (o.J.): Best Mobile Development Frameworks Software, kein Veröffentlichungsdatum angegeben, zuletzt abgerufen am 12.07.2018, https://www.g2crowd.com/categories/mobile-development-frameworks

Gaunt, Matt (2018a): How Push Works, veröffentlicht am 02.07.2018, zuletzt abgerufen am 20.07.2018, https://developers.google.com/web/fundamentals/push-notifications/how-push-works

Gaunt, Matt (2018b): Service Workers - an Introduction, veröffentlicht am 29.03.2018, https://developers.google.com/web/fundamentals/primers/service-workers/

Gaunt, Matt & Kinlan, Paul (2018): The Web App Manifest, veröffentlicht am 24.07.2018, zuletzt abgerufen am 24.07.2018, https://developers.google.com/web/fundamentals/web-app-manifest/

Gasston, Peter (2014): Moderne Webentwicklung - Geräteunabhängige Entwicklung -- Techniken und Trends in HTML5, CSS3 und JavaScript, dpunkt.verlag, 2014

Google Developers (o.J.): Progressive Web Apps - A new way to deliver amazing user experiences on the web., kein Veröffentlichungsdatum angegeben, zuletzt abgerufen am 10.70.2018, https://developers.google.com/web/progressive-web-apps/

Gutknecht-Gmeiner, Maria (2008): Externe Evaluierung durch Peer Review, 1. Auflage, Wiesbaden, 2018

Hellbusch, Jan E. & Probiesch, Kertin (2011): Barrierefreiheit verstehen und umsetzen - Webstandards für ein zugängliches und nutzbares Internet, 1. Auflage, Heidelberg, 2011

IETF (2016): RFC 8030 – Generic Event Delivery Using http Push (Version 12), veröffentlicht im Dezember 2016, zuletzt abgerufen am 20.07.2018, https://datatracker.ietf.org/doc/rfc8030/

Jeromin, Holger (2018): Add to Home screen, veröffentlicht am 11.06.2018, zuletzt abgerufen am 24.07.2018, https://developer.mozilla.org/en-US/Apps/Progressive/Add_to_home_screen

Knott, Daniel (2016): Mobile App Testing, 1. Auflage, Heidelberg, 2016

Kuß, Alfred u.a. (2018): Marktforschung - Datenerhebung und Datenanalyse, 6., überarbeitete und erweiterte Auflage, Wiesbaden, 2018

LePage, Pete (2018): Add to Home Screen, veröffentlicht am 24.07.2018, zuletzt abgerufen am 24.07.2018, https://developers.google.com/web/fundamentals/app-install-banners/

Liebel, Christian (2017a): Progressive Web Apps, Teil 5: Das App-Modell der Zukunft?, veröffentlicht am 10.07.2017, zuletzt abgerufen am 11.07.2018, https://www.heise.de/developer/artikel/Progressive-Web-Apps-Teil-5-Das-App-Modell-der-Zukunft-3767383.html

Liebel, Christian (2017b): Progressive Web Apps, Teil 2: Die Macht des Service Worker, veröffentlicht am 16.06.2017, zuletzt abgerufen am 18.07.2018, https://www.heise.de/developer/artikel/Progressive-Web-Apps-Teil-2-Die-Macht-des-Service-Worker-3740464.html

Liebel, Christian (2017c): Progressive Web Apps, Teil 3: Wie die Web-App zur App-App wird, veröffentlicht am 23.06.2017, zuletzt abgerufen am 24.07.2018, https://www.heise.de/developer/artikel/Progressive-Web-Apps-Teil-3-Wie-die-Web-App-zur-App-App-wird-3464603.html

Lynch, Max (2016): What are Progressive Web Apps?, veröffentlicht am 18.05.2016, zuletzt abgerufen am 20.07.2018, https://blog.ionicframework.com/what-is-a-progressive-web-app/

Magerhans, Alexander (2016): Marktforschung, Wiesbaden, 2016

Malavolta, Ivano (2016): Beyond Native Apps - Web Technologies to the Rescue! (Keynote). In: Proceedings of the 1st International Workshop on Mobile Development, S. 1-2

Maurice, Florence (2012): Mobile Webseiten - Strategien, Techniken, Dos und Don'ts für Webentwickler, München, 2012

Medley, Joseph (2018): Web Push Notifications - Timely, Relevant, and Precise, veröffentlicht am 03.01.2018, zuletzt abgerufen am 20.70.2018, https://developers.google.com/web/fundamentals/push-notifications/

Mozilla (o.J.): Firefox 44.0 Release Notes, kein Veröffentlichungsdatum angegeben, zuletzt abgerufen am 19.07.2018, https://www.mozilla.org/en-US/firefox/44.0/releasenotes/

Naderer, Gabriele & Balzer, Eva (Hrsg.) (2011): Qualitative Marktforschung in Theorie und Praxis, 2., überarbeitete Auflage, Wiesbaden, 2011

Navara, Erika Doyle u.a. (2018): Progressive Web Apps on Windows, veröffentlicht am 19.06.2018, zuletzt abgerufen am 24.07.2018, https://docs.microsoft.com/en-us/microsoft-edge/progressive-web-apps

Pflug, Kyle & McCormick, Libby (2018): What's new in Microsoft Edge in the Windows 10 April 2018 Update, veröffentlicht am 30.04.2018, zuletzt abgerufen am 19.07.2018, https://blogs.windows.com/msedgedev/2018/04/30/edgehtml-17-april-2018-update/

Que, Peixin u.a. (2016): A Comprehensive Comparison Between Hybrid and Native App Paradigms. In: 8th Internation Conference on Computational Intelligence and Communication Networks, Beijing, S. 611-614

Raab, Gerhard u.a. (2018): Methoden der Marketing-Forschung, 3., überarbeitete und erweiterte Auflage, Wiesbaden, 2018

Russell, Alex (2015): Progressive Web Apps - Escaping Tabs Without Losing Our Soul, veröffentlicht am 15.06.2015, zuletzt abgerufen am 10.07.2018, https://infrequently.org/2015/06/progressive-apps-escaping-tabs-without-losing-our-soul/

SELFHTML (2018a): HTML/Tutorials/Trennung von Inhalt, Präsentation und Verhalten, veröffentlicht am 20.06.2018, zuletzt abgerufen am 13.07.2018, https://wiki.selfhtml.org/wiki/HTML/Tutorials/Trennung_von_Inhalt,_Pr%C3%A4sentation_und_Verhalten

SELFHTML (2018b): Progressive enhancement, veröffentlicht am 05.07.2018, zuletzt abgerufen am 13.07.2018, https://wiki.selfhtml.org/wiki/Progressive_enhancement

SELFHTML (2018c): Semantik, veröffentlicht am 05.07.2018, zuletzt abgerufen am 13.07.2018, https://wiki.selfhtml.org/wiki/Semantik

Stack Overflow Talent (2017): Developer Hiring Landsscape – Global Report, o.O., 2017, zuletzt abgerufen am 11.07.2018, https://www.stackoverflowbusiness.com/hubfs/content/2017_Global_Developer_Hiring_Landscape.pdf

Starck, Matthias J. (2018): Peer Review für wissenschaftliche Fachjournale - Strukturierung eines informativen Reviews, Wiesbaden, 2018

StatCounter (2018a): Browser Market Share Germany, im Jahr 2018, zuletzt abgerufen am 03.10.2018, http://gs.statcounter.com/browser-market-share/all/germany/#yearly-2018-2018-bar

StatCounter (2018b): Mobile Operating System Market Share Worldwide, Zeitraum Januar bis Juli 2018, zuletzt abgerufen am 11.07.2018, http://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-201801-201807-bar

Steyer, Ralph (2017): Cordova - Entwicklung plattformneutraler Apps, Wiesbaden, 2017

Töpfer, Armin (2012): Erfolgreich Forschen, 3., überarbeitete und erweiterte Auflage, Wiesbaden, 2012

Vollmer, Guy (2017): Mobile App Engineering, 1. Auflage, Heidelberg, 2017

W3C (2018): Web App Manifest, veröffentlicht am 19.07.2018, zuletzt abgerufen am 24.07.2018, https://www.w3.org/TR/2018/WD-appmanifest-20180719/

Weber, Daniela (2015): Wissenschaftliches Arbeiten für Wirtschaftswissenschaftler, 1. Auflage, Weinheim, 2015

Anhang

Anhang A Caching Strategien

Strategie	**Beschreibung**	**geeignet …**
cache only	Anfragen werden immer aus dem Cache beantwortet	für statische Ressourcen
network only	Anfragen werden immer über das Netzwerk geleitet	wenn es kein Offline-Äquivalent gibt (z.B. POST-Anfragen)
cache, falling back to network	priorisiertes Laden aus dem Cache, falls nicht vorhanden, dann über das Netzwerk	für Webseiten, die nach dem „Offline First“ Ansatz agieren
network, falling back to cache	priorisiertes Laden über das Netzwerk, falls nicht möglich, dann aus dem Cache	für sich regelmäßig ändernde Ressourcen
cache & network race	paralleles Laden aus Cache und Netzwerk, schnellste Antwort wird verwendet	wenn das Gerät nur langsam auf die Festplatte zugreift
cache then network	Verwendung der Ressource aus dem Cache und nachträgliches Ersetzen mit der Ressource aus dem Netzwerk	wenn Daten nicht aktuell sein müssen
generic fallback	generische Antwort, falls Ressource weder aus dem Cache noch über das Netzwerk bezogen werden kann	als Fallback immer sinnvoll

Tabelle 8: Caching Strategien
Quelle: Vgl. Archibald, Jake (2014), Abschnitt „Serving suggestions – responding to requests“

Anhang B Inhalt des Web App Manifest nach dem W3C Standard

Attribute	Beschreibung
dir	die Schreibrichtung des Inhaltes der Felder name, short_name und description
lang	die Sprache, in der das Manifest verfasst ist
name	der Name der Anwendung
short_name	eine kurze Version des Namens der Anwendung
description	eine Beschreibung des Zweckes der Anwendung
icons	eine Liste von Icons, die die Anwendung repräsentieren
screenshots	eine Liste von Bildern, die die Anwendung während der Nutzung zeigen
categories	eine Liste von Kategorien, die dieser Anwendung zugeordnet werden können
iarc_rating_id	ein Identifikator der International Age Rating Coalition, der die Altersfreigabe dieser Anwendung beschreibt
start_url	die URL, die beim Start der Anwendung aufgerufen werden soll
display	der Anzeigemodus der Anwendung, mögliche Werte sind: • fullscreen (gesamter Bildschirm wird verwendet) • standalone (öffnet die App wie einer nativen / standalone App) • minimal-ui (gewährt einen minimalen Zugriff auf Browserelemente) • browser (öffnet die App wie üblich in einem Browser)
orientation	die Orientierung des Bildschirms bei mobilen Endgeräten
theme_color	primäre Farbe, die in der Anwendung verwendet wird
background_color	Hintergrundfarbe, die in der Anwendung verwendet wird
scope	der URL Bereich, auf den sich das Manifest auswirken soll
serviceworker	eine Beschreibung des von der Anwendung verwendeten Service Worker
related_applications	Liste von Anwendung, die mit dieser Anwendung verwandt sind
prefer_ related_applications	gibt an, ob die verwandten Anwendungen dieser Anwendung vorgezogen werden soll

Tabelle 9: Inhalt des Web App Manifest nach dem W3C Standard
Quelle: W3C (2018)

Anhang C Voraussetzungen zur Anzeige des Installationsbanners

Voraussetzungen		Chrome	Firefox	Edge
Nutzungsheuristik		(Nutzung > 30 s)	?	?
HTTPS		ja	ja	ja
Service Worker		ja	?	ja
Web App Manifest		ja	ja	ja
	name / short_name	ja	ja	?
	icons	192px + 512px	192px	?
	start_url	ja	ja	?
	display	fullscreen, standalone, minimal-ui	fullscreen, standalone	?
	backgroup_color	nein	ja	?

Legende	wird benötigt	keine Angabe vorhanden	wird nicht benötigt

Tabelle 10: Voraussetzungen zur Anzeige des Installationsbanners
Quellen: Vgl. LePage, Pete (2018), Abschnitt „What are the criteria?“; Jeromin, Holger (2018), Abschnitt „How do you make an app A2HS-ready?“ & Navara, Erika Doyle u.a. (2018), Abschnitt „Requirements“

Anhang D Internationale Definition von Progressive Web Apps

> *Progressive web apps (PWA) are responsive and securely transmitted web apps using a HTTPS encryption. PWAs are developed according to the progressive enhancement[1] principle. They provide offline functionality, installation and a reliable and engaging user experience via service worker[2], push notifications and web app manifest[3].*

1. Progressive enhancement is a principle for the development of websites defined by Steven Champeon in 2003.
2. A service worker is a modern web technology defined by the W3C in the following specification: https://www.w3.org/TR/service-workers-1/
3. A web app manifest if a centralized with metadata to a web application defined by the W3c in the following specification: https://www.w3.org/TR/appmanifest/

Anhang E Überblick über quantitative und qualitative Sozialforschung

Eigenschaft	Quantitative Forschung	Qualitative Forschung
Forschungsprozess	standardisiert, unflexibel	flexibel, offen
Anwendungsfeld	Hypothesenprüfung	Ursachenforschung
Form der Daten	metrisch (quantifizierbar)	nicht metrisch (nicht quantifizierbar)
Informationsgehalt	nur Daten, die sich quantifizieren lassen	tiefe Informationen mit individuellen Meinungen und Eindrücken
Stichprobengröße / Fallzahl	hoch	gering
Aufwand / Kosten je Fall	gering	hoch
Auswertung	statistische Methoden	Interpretation
Ergebnisse	objektiv, repräsentativ	subjektiv, nicht repräsentativ
Anforderungen an den Forscher	nur methodische Kompetenz nötig	methodische und hohe soziale Kompetenz nötig

Tabelle 11: Überblick über quantitative und qualitative Sozialforschung
Quelle: Vgl. Magerhans, Alexander (2016), S. 71

Anhang F Überblick über vollständig standardisierte Befragungen

Eigenschaft	mündlich		schriftlich	
	persönliche Befragung	telefonische Befragung	schriftliche Befragung	computergestützte Befragung
Voraussetzung	keine	Telefonanschluss	keine	Internetanschluss
Erreichbarkeit der Zielgruppe	schwer	einfach	schwer	einfach
Aufwand / Kosten	hoch	gering	gering	sehr gering
Teilnahme	hoch	gering	gering	hoch
Repräsentanz	hoch	gering	gering	hoch
Beantwortung der Fragen	synchron	synchron	asynchron	asynchron
Anonymität des Befragten	gering	mittel	hoch	hoch
Datenqualität	hoch	mittel	gering	hoch
Interviewer-Bias	sehr hoch	hoch	gering	gering

Tabelle 12: Überblick über vollständig standardisierte Befragungen
Quellen: Vgl. Berger-Grabner, Doris (2016), S. 162 ff.; Kuß, Alfred u.a. (2018), S. 127 ff. & Raab, Gerhard u.a. (2018), S. 114 ff.

Anhang G Quellenverzeichnis zum PWA Regelwerk

Für die Ermittlung der hardware- und softwarebasierten Funktionen von nativen und Web Apps wurde eine Vielzahl von Quellen verwendet. Um das Literaturverzeichnis übersichtlich zu halten, wurden diese Quellen in einem Verzeichnis in einer externen Datei „Feature Quellenverzeichnis.pdf“ aufgelistet, die der digitalen Version dieser Masterarbeit beigefügt wurde.

Die Informationen wurden zum größten Teil den folgenden Seiten entnommen:

- https://developer.android.com/ (Android Developer)
- https://developer.apple.com/ (Apple Developer)
- https://www.w3.org/ (W3C)
- https://developers.google.com/ (Google Developer)
- https://developer.mozilla.org/ (Mozilla Developer Network)

Anhang H Hardwarebasierte Funktionen von nativen Apps

Funktionalitäten	benötigt Hardware	Android	iOS	Gesamt
Aufnahme von Medien		+	+	+
Aufnahme von Bildern	Kamera	+	+	+
Aufnahme von Audio	Mikrofon	+	+	+
Aufnahme von Video	Kamera, Mikrofon	+	+	+
Ausgabe von Medien und Informationen		+	~	~
Anzeigen von Bildern	Display	+	~	~
Rastergrafiken		+	+	+
Vektorgrafiken		+	~	~
fortgeschrittenes Rendering		+	+	+
Wiedergabe von Audio	Lautsprecher	+	+	+
Wiedergabe von Video	Display, Lautsprecher	+	+	+
Vibration	Vibrationsaktor	+	~	~
Vibrationslänge		+	-	~
Vibrationsstärke		+	-	~
Persistenz	Festwertspeicher	+	+	+
Schlüssel-Wert-Paare		+	+	+
Strukturierte Daten		+	+	+
Dateisystem		+	+	+

Kommunikation			+	~	~
	Mobilfunkbasierte Kommunikation	Mobilfunkantenne, SIM	+	~	~
	Telefonie		+	~	~
	SMS		+	+	+
	Internetbasierte Kommunikation	Mobilfunkantenne, WLAN-Chip	+	~	~
	Abfrage des Netzwerkstatus		+	~	~
	Kommunikation über das Internet		+	+	+
	Sonstige Kommunikation		+	~	~
	GPS	GPS-Antenne	+	+	+
	Bluetooth	Bluetooth-Chip	+	+	+
	NFC	NFC-Chip	+	~	~
	USB	USB-Port	+	+	+
Sensoren			+	~	~
	Bewegungssensoren		+	+	+
	Beschleunigung	Beschleunigungssensor	+	+	+
	Rotation	Gyroskop	+	+	+
	Umgebungssensoren		+	~	~
	Licht	Umgebungslichtsensor	+	-	~
	Temperatur	Umgebungstemperatursensor	+	-	~
	Magnetfeld	Magnetfeldsensor	+	+	+
	Näherung	Näherungssensor	+	~	~
	Luftdruck	Barometer	+	~	~
	Luftfeuchtigkeit	Hygrometer	+	-	~

Legende

+	Funktion vorhanden	~	Funktion teilweise vorhanden	-	Funktion nicht vorhanden

Tabelle 13: Hardwarebasierte Funktionen von nativen Apps
Quelle: Vgl. Anhang G, Quellenverzeichnis zum PWA Regelwerk

Anhang I Softwarebasierte Funktionen von nativen Apps

Funktionalitäten	Android	iOS	Gesamt
Softwarebasierte Kommunikation	+	+	+
E-Mails	+	+	+
Push Notifications	+	+	+
Installation	+	+	+
Offlinebetrieb	+	+	+
Apps zur persönlichen Organisation	+	~	~
Wecker	+	~	~
Kalender	+	+	+
Kontaktliste	+	+	+
Notizen	+	-	~
Karten-App	+	+	+
Positionsanzeige	+	+	+
Geocoding	+	+	+
Geofencing	+	+	+
Vertrieb	+	+	+
direkter Verkauf	+	+	+
In-App Verkäufe	+	+	+
Text- und Sprachverarbeitung	+	+	+
Sprachsynthese (Text to Speech)	+	+	+
Spracherkennung (Speech to Text)	+	+	+

Legende					
+	Funktion vorhanden	~	Funktion teilweise vorhanden	-	Funktion nicht vorhanden

Tabelle 14: Softwarebasierte Funktionen von nativen Apps
Quelle: Vgl. Anhang G, Quellenverzeichnis zum PWA Regelwerk

Anhang J Hardwarebasierte Funktionen von Web Apps

Funktionalitäten	Chrome	Firefox	Safari	IE	Edge	Gesamt
Aufnahme von Medien	~	~	~	~	~	~
Aufnahme von Bildern	~	~	~	~	~	~
Image-Input-Element	+	+	+	+	+	+
Media Capture API[1]	47	36	11	-	+	~
Aufnahme von Audio	~	~	~	~	~	~
Audio-Input-Element	+	+	+	+	+	+
Media Capture API	47	36	11	-	+	~
Aufnahme von Video	~	~	~	~	~	~
Video-Input-Element	+	+	+	+	+	+
Media Capture API	47	36	11	-	+	~
Ausgabe von Medien und Informationen	~	~	~	~	~	~
Anzeigen von Bildern	+	+	~	~	~	~
Rastergrafiken (Image-Element)	+	+	+	+	+	+
Vektorgrafiken (Image-Element)	+	+	+	+	+	+
WebGL 1 (fortgeschr. Rendering)	9	4	5.1	11	+	+
WebGL 2 (fortgeschr. Rendering)	56	51	-	-	-	~
Wiedergabe von Audio	+	+	+	~	+	~
Audio-Element	3	3.5	3.1	9	+	+
Web Audio API	14	23	6	-	+	~
Wiedergabe von Videos	3	3.5	3.1	9	+	+
Vibration	~	~	-	-	-	~
Vibrationslänge	32	16	-	-	-	~
Vibrationsstärke	-	-	-	-	-	-
Persistenz	~	~	~	~	~	~
Schlüssel-Wert-Paare	4	3.5	4	8	+	+
Strukturierte Daten	23	10	7.1	10	+	+
Dateisystem	-	-	-	-	-	-
Kommunikation	~	~	~	~	~	~
Mobilfunkbasierte Kommunikation	+	+	+	+	+	+
Telefonie	+	+	+	+	+	+
SMS	+	+	+	+	+	+
Internetbasierte Kommunikation	+	~	~	~	~	~
Abfrage des Netzwerkstatus	61	31[2]	-	-	-	~
Kommunikation über das Internet	+	+	1.2	7	+	+

Sonstige Kommunikation		~	~	~	~	~	~
	GPS	5	3.5	5	9	+	+
	Bluetooth	53[3]	-	-	-	-	~
	NFC	-	-	-	-	-	-
	USB	61[3]	-	-	-	-	~
Sensoren		~	~	~	-	~	~
Bewegungssensoren		+	+	~	-	+	~
	Beschleunigung	+	~	~	~	~	~
	devicemotion Event	+	6	4.2[2]	11[4]	+	~
	Accelerometer API	67	-	-	-	-	~
	Rotation	+	~	~	~	~	~
	deviceorientation Event	7	6	4.2[2]	11[4]	+	~
	Gyroscope API	67	-	-	-	-	~
Umgebungssensoren		~	~	-	-	~	~
	Licht	~	~	-	-	~	~
	devicelight Event	-	62[5]	-	-	+	~
	Ambient Light Sensor API	54	-	-	-	-	~
	Temperatur	-	-	-	-	-	-
	Magnetfeld (Magnetometer API)	69	-	-	-	-	~
	Näherung	-	~	-	-	-	~
	userproximity Event	-	62[5]	-	-	-	~
	Proximity Sensor API	-	-	-	-	-	-
	Luftdruck	-	-	-	-	-	-
	Luftfeuchtigkeit	-	-	-	-	-	-

Legende und Fußnoten

+	Funktion vorhanden	~	Funktion teilweise vorhanden	-	Funktion nicht vorhanden

1. Die Aufnahme von Bildern ist nur über ein Workaround möglich.
2. Diese Funktion ist nur auf mobilen Endgeräten verfügbar.
3. Diese Funktion ist nur bei einer HTTPS-Verbindung verfügbar.
4. Diese Funktion ist nur auf Windows 8.1 und höher verfügbar.
5. Zur Nutzung dieser Funktion muss eine Konfiguration aktiviert werden.

Tabelle 15: Hardwarebasierte Funktionen von Web Apps
Quelle: Vgl. Anhang G, Quellenverzeichnis zum PWA Regelwerk

Anhang K Softwarebasierte Funktionen von Web Apps

Funktionalitäten	Chrome	Firefox	Safari	IE	Edge	Gesamt
Softwarebasierte Kommunikation	~	~	~	~	~	~
E-Mails	+	+	+	+	+	+
Push Notifications	-	-	-	-	-	-
Installation	-	-	-	-	-	-
Offlinebetrieb[1]	4	3.5	4	10	~	~
Apps zur persönlichen Organisation	~	~	~	~	~	~
Wecker	-	-	-	-	-	-
Kalender	~	~	~	~	~	~
Google Calendar	+	+	+	+	+	+
iCloud Calendar	-	-	-	-	-	-
Kontaktliste	~	~	~	~	~	~
Google Contacts	+	+	+	+	+	+
iCloud Contacts	-	-	-	-	-	-
Notizen	-	-	-	-	-	-
Karten-App (Google + iCloud Maps)	+	+	+	+	+	+
Positionsanzeige	+	+	+	+	+	+
Geocoding	+	+	+	+	+	+
Geofencing	+	+	+	+	+	+
Vertrieb	~	~	~	~	~	~
direkter Verkauf	-	-	-	-	-	-
In-App Verkäufe	+	~	+	~	+	~
Payment Request API	61	55[2]	11.1	-	+	~
Online-Shop Checkout	+	+	+	+	+	+
Text- und Sprachverarbeitung	+	~	~	~	~	~
Sprachsynthese (Text to Speech)	+	+	+	~	+	~
Web Speech API	33	49	7	-	+	~
Text-To-Speech Webservice	+	+	+	+	+	+
Spracherkennung (Speech to Text)	+	~	+	~	+	~
Web Speech API	25	44[2]	7	-	+	~
Speech-To-Text Webservice	+	+	+	+	+	+

Legende und Fußnoten

+	Funktion vorhanden	~	Funktion teilweise vorhanden	-	Funktion nicht vorhanden

1. Partieller Support, weil der Application Cache veraltet ist und zunehmend vom Web entfernt wird.
2. Zur Nutzung dieser Funktion muss eine Konfiguration aktiviert werden.

Tabelle 16: Softwarebasierte Funktionen von Web Apps
Quelle: Vgl. Anhang G, Quellenverzeichnis zum PWA Regelwerk

Anhang L Regelwerk- und Interview-Dokumente

Das PWA-Regelwerk und die durchgeführten Experteninterviews wurden in einer Reihe von externen Dokumenten erfasst, die der digitalen Version dieser Masterarbeit beigefügt wurden. Diese externen Dokumente wurden unter den folgenden Pfaden abgelegt:

Dokument	**Dateipfad**
PWA-Regelwerk	`\Regelwerk\PWA Regelwerk v*.pdf`
Interviewleitfaden	`\Regelwerk\Interviewleitfaden.pdf`
Interview-Aufzeichnungen	`\Interviews\Aufnahmen\*`
Interview-Transkriptionen	`\Interviews\Transkription\*`
Interview-Auswertungen	`\Interviews\Auswertungen\*`

Tabelle 17: Dateipfade zu Regelwerk- und Interview-Dokumente

Anhang M Anmerkungen der Experten im Interview

Anmerkung	**Übernommen? Falls nein, Begründung**
Anmerkungen zur Korrektheit	
Ein Zugriff auf das Dateisystem kann im Web über Java- oder Flashanwendungen erfolgen.	Nicht übernommen, weil Java- und Flash-anwendungen veraltet sind und aus Sicherheits-gründen nicht aufgeführt werden sollten.
Der Link zur Spezifikation in den Abschnitten zu Beschleunigungs- und Rotationssensor ist falsch.	Übernommen, Link korrigiert.
Die Beschreibung im Abschnitt Vertrieb ist aus fachlicher Sicht falsch.	Übernommen, Beschreibung angepasst.
Eine Wecker-Funktion kann im Web über die Task Scheduler API erzeugt werden.	Nicht übernommen, weil die API vom W3C nicht als Standard empfohlen und in keinem Browser unterstützt wird.
Die Web Audio API ist je nach Use Case ggfs. nicht praktisch einsatzbar.	Nicht übernommen, weil im Regelwerk nicht alle möglichen Use Cases abgedeckt werden können.
Die Network Information API ist in den meisten Fällen praktisch nicht einsetzbar.	Nicht übernommen, weil im Regelwerk nicht alle möglichen Use Cases abgedeckt werden können.
Positionsbestimmung ist auch ohne GPS über WLAN möglich.	Übernommen, Abschnitt angepasst.
Die Web USB API ist in den meisten Fällen praktisch nicht einsetzbar.	Nicht übernommen, weil im Regelwerk nicht alle möglichen Use Cases abgedeckt werden können.
Der Safari unterstützt Service Worker und Web App Manifest nicht wie spezifiziert.	Übernommen, Anmerkung ergänzt und Kompatibilität angepasst.

Anmerkungen zur Vollständigkeit	
Das canvas-Element kann auch ohne WebGL über seinen 2D Kontext genutzt werden.	Übernommen, 2D Kontext hinzugefügt.
Wiedergabe von Videos ist auch über den Flash-Player möglich.	Nicht übernommen, weil der Flash-Player veraltet ist und aus Sicherheitsgründen nicht aufgeführt werden sollte.
Die Rotation kann auch im CSS über die orientation Media Query ermittelt werden.	Nicht übernommen, weil diese Media Query lediglich die Seitenverhältnisse vergleicht und keine Rotationsdaten bereitstellt.
Die Web Share API sollte noch im Regelwerk aufgenommen werden.	Als neue Regel übernommen.
Die Information, dass man einen Vollbildmodus über das Web App Manifest erzeugen kann, geht nicht aus dem Regelwerk hervor.	Übernommen, Vollbildmodus im Abschnitt „Installation und Offlinebetrieb“ aufgeführt.
Die Vor- und Nachteile von PWAs sollten in der Einleitung beschrieben werden.	Die Vor- und Nachteile befinden sich im gesamten Entscheidungsprozess vor der Nutzung des Regelwerks und sollten somit in diesem nicht mehr nötig sein.
Nicht unterstützte Bildformate können über das canvas-Element dargestellt werden.	Übernommen, Beschreibung ergänzt
Nicht unterstützte Videoformate können über das canvas-Element dargestellt werden.	Nicht übernommen, weil es ein sehr unschönes Workaround ist.
Das Umgebungslicht kann auch im CSS über die light-level Media Query ermittelt werden.	Nicht übernommen, weil die Media Query von keinem Browser überstützt wird.
Vibrationen können auch indirekt über die Notification API erzeugt werden.	Nicht übernommen, weil es nur eine Alternative ohne Mehrwert ist.
OpenStreetMap sollte neben Google und Apple Maps als Kartendienst ergänzt werden.	Übernommen, OpenStreetMap ergänzt.
Neben den technischen spielen noch viele „softe“ Entscheidungskriterien eine Rolle.	Nicht übernommen, weil sich das Regelwerk auf technische Anforderungen beschränkt und softe Kriterien schwer abgebildet werden können.
Sonstige Anmerkungen	
Samsung Internet, der native Android Browser und Opera sollten in den Kompatibilitätstabellen aufgeführt werden.	Aus zeitlichen Gründen nicht übernommen, wird jedoch als sinnvoll erachtet.
Eine fehlende Funktion im Web sollte nicht als Grund gesehen werden, eine PWA nicht zu nutzen, sondern eher als Anlass genutzt werden, eine kreative Lösung zum Problem zu finden.	Übernommen, Kerngedanke diese Anmerkung in der Einleitung ergänzt.

Tabelle 18: Anmerkungen der Experten im Interview
Quelle (Anmerkungen): siehe Anhang L, Regelwerk- und Interview-Dokumente
Quelle (Begründungen): siehe Anhang G, Quellenverzeichnis zum PWA Regelwerk

Anhang N Dokumente und Ressourcen zur Leitfaden-App

Die Leitfaden-App an sich kann unter der folgenden URL aufgerufen werden: https://kfrank-vfh.github.io.

Die Ressourcen der Leitfaden-App sind online in folgendem GitHub-Repository verfügbar: https://github.com/kfrank-vfh/Should_I_PWA.

Zudem wurden die Ressourcen der Leitfaden-App der digitalen Version dieser Masterarbeit beigefügt. Diese wurden unter den folgenden Pfaden abgelegt:

Ressource	**Dateipfad**
Git Repository	`\Leitfaden-App\`
Baumstruktur der Regeln als JSON	`\Leitfaden-App\js\rules.js`
Manifest.json der Leitfaden-App	`\Leitfaden-App\manifest.json`

Neben diesen App-Ressourcen wurden auch die Dokumente zur Evaluation der Leitfaden-App der digitalen Version beigefügt. Diese befinden sich unter den folgenden Pfaden:

Dokument	**Dateipfad**
Druckversion der Umfrage	`\Umfrage\Progressive Web App Umfrage.pdf`
Mailverkehr zur Umfrage	`\Umfrage\*.eml`
Rohdaten der Umfrage im CSV-Format	`\Umfrage\Umfrage Rohdaten.csv`
Auswertung der Umfrage in Excel	`\Umfrage\Umfrage Auswertung.xlsx`

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides Statt,

dass ich die vorliegende Masterarbeit selbstständig angefertigt habe,

dass ich die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet habe,

dass ich die vorliegende Masterarbeit bei keiner anderen Prüfung vorgelegt habe.

Alle Quellen, die dem World Wide Web entnommen oder in einer sonstigen digitalen Form verwendet wurden, sind der Arbeit beigefügt.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Ort, Datum Unterschrift