

# MOBILIZE ME! APPS FÜR MOBILGERÄTE ODER MOBILE WEB APPS – TECHNIKEN, WERKZEUGE, BEWERTUNGEN

Der Trend zum Mobile Web ist ungebrochen. Immer mehr Menschen wickeln ihr berufliches wie ihr privates Dasein über Smartphones und Tablets ab. Darum haben Tools für die Kundenansprache im Web Hochkonjunktur. Mit ihnen lassen sich Botschaften streuen und Bedürfnisse wecken. Dieses Whitepaper gibt einen Überblick zu den wichtigsten mobile Web-Techniken.

## WARUM IST DAS MOBILE WEB SO EN VOGUE?

Im mobilen Internet tobt der Kampf um die Aufmerksamkeit der Nutzer. Carrier, Contentprovider, Consumer und nicht zuletzt die großen Communitybetreiber wie Google, Amazon, Facebook & Co buhlen - jeder auf seine Weise - um die Gunst der Webcrowd. Gleichzeitig bilden sich aus Teilen dieser Player interessante Newcomer. Insgesamt geht es im neuen Big Deal um Soft Skills: Aufmerksamkeit, Bewertung und Empfehlung. Harte Dollars verdient nur, wer diese weichen Faktoren zu organisieren weiß. Ob etablierter Player oder Newcomer, Erfolg haben allein solche Unternehmen, die mit Tools und Taktik die kritische Größe überschreiten und damit Relevanz im Werbemarkt erreichen.

Das kann sowohl der Schwarm, als auch der Pottwal oder Riesenkrake im Profildaten-Ozean sein. Hauptsache, der Fang ist gut und die eigene Firma steht weit genug oben in der Nahrungskette. Die notwendigen Tools und Strategien hierfür liefern Spezialisten für das Ausbringen von Netzen im Netz, vulgo: Apps. Richtig konzipiert und platziert, liefern Apps ihren Anbietern, jenen Unternehmen mit Ambitionen auf Relevanz und Größe im Datenmeer, die Möglichkeit zu massiver Kundenansprache.

Derzeit konkurrieren zwei technische Ansätze: Zum einen die nativen Apps in den Stores von Apple, Google/Android, Samsung oder RIM/Blackberry. Zum anderen die für den Zugriff mit Smartphones adaptieren Websites, Fachjargon: Mobile Web Apps. Dazwischen sind hybride Lösungen angesiedelt.

## MOBILE WEB APPLIKATIONEN LASSEN SICH MEIST EINFACHER HANDHABEN

Mobile Web Apps, auch „HTML5 App“ genannt, sind herkömmliche Webseiten, die für den Zugriff durch Smartphones optimiert wurden. Nicht vorhandene oder eingeschränkte Tastaturen, kleine Bildschirme und längere Zeiten für den Seitenaufbau erfordern besondere Formen der Benutzerführung. Hierzu gehört beispielsweise die Möglichkeit, über einfache Pfeiltasten zu den Menüpunkten zu kommen. Zudem sind mobile Webseiten wesentlich einfacher aufgebaut, damit Nutzer sie schneller aufrufen und leichter bedienen können.



Im Vergleich zu den Native Apps bieten Mobile Web Apps aber vor allem eins: Unabhängigkeit. Native Apps müssen erst aufwändige und teure Zertifizierungsprozesse der Plattformbetreiber passieren, bevor sie in den Stores erscheinen. Mobile Web Apps dagegen brauchen lediglich eine Webadresse (URL) und sind damit von der Vertriebspolitik und den Einschränkungen Dritter unabhängig. Jeder App-Entwickler kennt Fälle, wo eine perfekt funktionierende App nicht angenommen wurde, nur weil sie irgendeine Richtlinie des Store-Betreibers nicht einhielt. Auch müssen erzielte Erlöse mit niemandem geteilt werden. Positiv hinzukommt, dass mögliche rechtliche Probleme aufgrund von In-App-Käufen vermieden werden. Diese basieren auf In-App Bezahlösungen, die bei Anwendern zu bösen Überraschungen führen können. Sie führen zu einer Art en-passant-Payment, bei der ein Nutzer einen Zahlvorgang meist mit einem Klick vornimmt, der auf den ersten Blick gar nicht als Bezahlvorgang wahrgenommen wird.

## NIEDRIGERER KOSTENAUFWAND BEI MOBILE WEB APPS

Verglichen mit den Native Apps, bei denen Anbieter alle mobilen Plattformen ihrer anvisierten Zielgruppen verwalten und unterstützen müssen, haben sie es mit einer Mobile Web App in vielen Punkten einfacher. Nur eine einzige Mobile Web App entwickeln und bereitstellen zu können, erleichtert einiges. Natürlich sind nicht alle Mobilgeräte mit demselben Browser ausgestattet und nicht alle Browser sind komfortabel und qualitativ hochwertig. Auch kann das Verhalten unterschiedlicher Browser-Versionen mehr oder minder stark voneinander abweichen. Dennoch ist bei geeigneter Planung und Entwicklung der Aufwand für eine browserübergreifende Anwendung wesentlich geringer als die Verwaltung einer eigenen Native App für jede einzelne mobile Plattform. Im Vergleich mit der nicht-mobilen Welt ist das kein Novum.

## HERAUSFORDERUNG HAPTİK UND USAGE

Einer der größten Nachteile einer Mobile Web App liegt in ihrer Benutzeroberfläche, auf der die sonst aus der Desktop-Welt gewohnten Bedienelemente fehlen. Dabei sollten Anbieter die Macht der Gewohnheit nicht unterschätzen. Die Art und Weise, wie ein Nutzer seine Anwendung erlebt, ist eng mit der gewohnten Benutzeroberfläche im stationären, also über PC- oder Laptop-Bildschirm und Tastatur genutzten Web verbunden. Anwender sollten also keinen zu steilen Lernkurven ausgesetzt werden. Auch auf die Bequemlichkeit, die ein Nutzer benötigt, um sich mit der neuen App anzufreunden, darf auf keinen Fall verzichtet werden.

Fehlender Zugang der Mobile Web App zu den meisten Hardware-Elementen des Geräts ist ein weiterer Punkt, der berücksichtigt werden muss. In der Regel kann die App zwar auf die Geolokationsfunktion und den lokalen Speicher des Mobilgeräts zugreifen, nicht jedoch auf die Kamera, den Beschleunigungsmesser oder die Kontaktliste. Je nach Umfang der App kommt der Anwender aber auch ohne diese Funktionen aus.

## TECHNIKEN FÜR MOBILE KLONE STATIONÄRER WEBSITES ERLEICHTERN DIE HERSTELLUNG

Neben dem Abwägen von Vor- und Nachteilen aus geschäftlicher Perspektive müssen die technischen Gegebenheiten bei der Einführung einer Mobile Web App Berücksichtigung finden.

So bestehen nur geringe Unterschiede zwischen der Entwicklung einer mobilen Web-Applikation und einer für den Desktop. Für beide werden HTML, CSS und Javascript benötigt. Bei Verwendung bestimmter Gestaltungs- und Entwicklungstechniken und dem Einsatz von ad-hoc entwickelten Frameworks reicht auch die Realisierung einer einzigen Web App, die sich elegant auf eine mobile Version abspecken lässt. Der Einsatz von Media Queries macht dies möglich.

Media Queries bestehen aus einer Reihe von Programmierbefehlen, die der Browser auswerten kann. Je nach Ergebnis der Auswertung stellt der Browser dieselben Inhalte unterschiedlich dar. Media Queries wurden nicht speziell für die Anpassung von Inhalten an mobile Anforderungen entwickelt, sondern ganz allgemein dafür, dieselben Inhalte an die Anforderungen unterschiedlicher Medien anzupassen. Neben Media Queries sind weitere Techniken, Werkzeuge und Frameworks erhältlich, die jedoch alle mehr oder minder auf demselben Konzept basieren: der Anpassung von Inhalt an das jeweils verwendete Medium. Diese Technik ist unter der Bezeichnung „Responsive Web Design“ (reagierende Web-Gestaltung) bekannt.

## DIE RICHTIGE WAHL DER TOOLS MACHT DEN MEISTER

Die Debugging-Strategie für Mobile Web Apps ähnelt stark der für Desktops. So lässt sich unter Verwendung von Tools wie FireBug oder WebKit Inspector ein Code auf jedem modernen Desktop-Browser testen. Mittlerweise gibt es auch einige Debugging-Tools für die Remote-Verwendung. Mit diesen Lösungen lässt sich das Debugging der Apps durchführen, während sie auf dem Mobilgerät laufen. Die Vorteile liegen auf der Hand. Wenn die App auf dem Mobilgerät läuft, bleiben Look & Feel erhalten. Auch das Erkennen und Beheben möglicher Pannen und Leistungsprobleme aufgrund der knappen Ressourcen des Mobilgeräts selbst, ist damit wesentlich einfacher.

## HYBRID-APPS - FÜR WANDERER ZWISCHEN DEN PLATTFORM-WELTEN

Ein Teil der Probleme, die bei der Verwendung rein mobiler Web-Apps entstehen, lässt sich durch sogenannte Cross-Compiled oder Hybrid-Apps vermeiden, die weder „nativ“ sind, noch auf dem Internet basieren. Diese Art von App wird typischerweise als Mobile Web App entwickelt. Sie erlangt jedoch über eine fremde Middleware-Lösung Zugang zu den Hardware-Sensoren des mobilen Geräts sowie zu anderen Funktionen, die für eine normale Mobile Web App nicht zur Verfügung stehen.

Auch eine solche App besteht aus HTML-, Javascript- und CSS-Code, der dann in eine native Applikation verpackt wird. Diese führt dann den Code in der nativen Standardkomponente der Web-Engine aus. Je nachdem, mit welchem Tool das Verpacken vorgenommen wird, hat der Entwickler Zugriff auf weitere zusätzliche Funktionen. Doch auch bei Verwendung dieser Frameworks gibt es nur eine Möglichkeit, absolut sicher zu sein, dass die Anwendung auf jedem Gerätetyp einwandfrei läuft: Testen. Die verkürzte Entwicklungszeit wird möglicherweise durch den Testaufwand wieder aufgehoben.

## AUCH DIE VIELSEITIG VERWENDBARE KREUZUNGS-TECHNIK HAT IHRE TÜCKEN

Der Vorteil des Hybrid-Ansatzes liegt darin, dass die mobile Web-App vom Endverbraucher genau wie eine Native App wahrgenommen wird, während die meisten Vorteile einer Mobile Web App erhalten bleiben. Das heißt, dieser Ansatz wäre geradezu ideal, wenn nicht auch einige der Nachteile erhalten blieben: Ein Nachteil besteht in der Notwendigkeit, die Vertriebskanäle Dritter zu nutzen und damit Genehmigungsabläufen zu unterliegen. Wie bereits erwähnt, liegt ein Vorteil der Hybrid-App in der Tatsache, dass der Kern-Code der App für alle Plattformen im Wesentlichen unverändert bestehen bleibt. Da es sich um eine Mobile Web App

handelt, die in eine native App verpackt wird, erhöht sich aber der Testaufwand für das Endprodukt. Die einwandfreie Funktion muss auf jeder einzelnen Plattform getestet werden.

## DIE WERKZEUG-AUSWAHL IST ERFOLGSKRITISCH

Auf dem Markt gibt es viele unterschiedliche Werkzeuge mit dem Ziel, die Umsetzung solch mobiler Hybrid-Apps zu vereinfachen und zu verbessern. Die bekanntesten sind offenkundig PhoneGap und Titanium Mobile. Auch wenn kürzlich von Adobe erworben, soll PhoneGap bald zur Apache Software Foundation gehören und bleibt daher (wenn auch möglicherweise unter anderem Namen) für Entwickler kostenlos. PhoneGap stellt ein Plugin für alle größeren Entwicklungskits für Mobile Software bereit, vom iPhone bis zum Android-Handy, BlackBerry und anderen. Entwickler nutzen dabei einfach die einheitliche Javascript-API des Frameworks zur Interaktion mit gerätespezifischen Funktionen. Ein Problem dabei ist die Eins-zu-eins-Beziehung zwischen Plugin und Zielplattform. Sie zwingt den Entwickler, zusammen mit dem spezifischen PhoneGap-Plugin alle unterschiedlichen SDKs (Software Entwicklungswerkzeuge) für alle Plattformen zu installieren, für die sie entwickeln möchten.

Basierend auf PhoneGap gibt es mit Apparat.io, einem sich noch in der geschlossenen Beta-Phase befindenden Online-Tool, einen weiteren Erfolg versprechenden Ansatz. Mit dem Werkzeug lässt sich eine Native Mobile App aus dem Code erstellen, den der Entwickler in ein GitHub-Repository, also ein von Git kontrolliertes Verzeichnis, schiebt. Dazu braucht er die App nur mit einem Code wie bei jeder anderen Mobile Web App zu entwickeln, den Code in das GitHub-Verzeichnis zu schieben und ihn dann im Apparat.io-Dashboard in ein paar einfachen Schritten zu verknüpfen. Ergebnis ist die gewünschte Android- oder iPhone-Anwendung. Dem Vernehmen nach werden in Kürze auch BlackBerry RIM und WebOS (HP, soll demnächst Open Source verfügbar sein) unterstützt werden.

Titanium Mobile vom Anbieter Appcelerator ist eine weitere kommerzielle Plattform zum Implementieren von mobilen Hybrid-Apps. Mit seiner etwas umfangreicheren Lösung hält es für den Entwickler eine Reihe fortschrittlicher Funktionen wie native UI-Elemente, integrierte Social-Sharing-Bibliotheken und Analyse-Tools sowie eine vollständige Entwicklungsumgebung bereit.

## NATIVE APPS ALS DIE SICHERSTE, VERTRIEBLICH GESEHEN ABER DIE SCHWIERIGSTE ALTERNATIVE

Wer sich dafür entscheidet, native Applikationen zu entwickeln, muss unter Umständen viele verschiedene Plattformen unterstützen. Praktische Probleme entstehen aufgrund unterschiedlicher Entwicklungskits, unterschiedlicher Benutzeroberflächen der Geräte, der unterschiedlichen Anwendererfahrung und auch aufgrund der verwendeten

Programmiersprachen wie Objective-C/C++ auf Apple iOS, Java auf Android, C# auf Windows Phone 7, sowie C++ oder Java ME auf Symbian und Bada.

Immer im Raum steht – zusätzlich zur technischen Basis für die App selbst – auch die Frage nach den in der anvisierten Zielgruppe vornehmlich genutzten Endgeräten. Selbst innerhalb einer Gerätegruppe gibt es jede Menge Fragmentierung. Da sind zum einen die aktuellen Handys, die unter Windows Phone 7 (WP7) laufen. Bis dato ist die Familie der WP7-Geräte die am wenigsten fragmentierte Gruppe. Das liegt an den strikten Auflagen von Microsoft, nach denen bei allen Mobiltelefonen, die Windows Phone 7 verwenden, nur eine bestimmte Auflösung und ein allgemeines Mindestset an Funktionen zugelassen ist.

## JEDE ENDGERÄTE-GRUPPE HÄLT IHRE EIGENEN BESONDERHEITEN FÜR NATIVE APPS BEREIT



Die Familie der Apple iOS-Geräte ist schon wesentlich verzweigt. Die Hardware-Unterschiede in Bezug auf Display-Größe, RAM, CPU-Typ und -Geschwindigkeit sowie GPU zwischen einem iPod Touch der ersten Generation, einem iPhone 3GS, einem iPhone 4 und einem iPad 2 sind kaum zu übersehen. Selbst auf die Version des Betriebssystems kommt es an. Soll eine App auch mit älteren Geräten kompatibel sein, auf denen nicht die aktuellste OS-Version läuft, müssen auch die API-Unterschiede berücksichtigt werden. Ferner gilt es zu beachten, dass ARMv7-Code nicht auf iPhone 3G oder einem iPod Touch der zweiten Generation läuft.

Dann wäre da noch Android. Nicht nur, dass Applikationen für Android-Handys zum größten Teil aus Dalvik-Bytecode bestehen sollten, auch die unterschiedliche Hardware, die verschiedenen Größen der Displays (von einer 240x320-Auflösung der einfachen Geräte bis zu einer Auflösung ähnlich der eines Tablet-Computers) und die diverse Firmware tragen nicht dazu bei, den Entwicklern das Leben leichter zu machen. Hinzu kommen weitere Unterschiede durch neue Versionen des Betriebssystems, also mit Android 2.1 und dem gerade eingeführten Android 4.0. Außerdem werden viele Teile des Android-Betriebssystems von den Hardware-Herstellern spezifisch angepasst, wodurch sich das Verhalten manchmal geringfügig, aber für Entwickler signifikant ändert. Das gilt sogar innerhalb desselben Gerätetyps, je nachdem, ob das Gerät mit dem Markenzeichen des Mobilfunkbetreibers versehen wurde oder nicht.

Insgesamt wird damit deutlich: Wenn es schon schwierig ist, die Konsistenz einer Anwendung innerhalb derselben Produktfamilie sicherzustellen, ist es im zweiten Schritt noch aufwändiger, Konsistenz über alle anvisierten Plattformen hinweg zu erreichen.

## WHEN THE GOIN' GETS ROUGH - JE KOMPLIZIERTER DIE MASCHINE, DESTO GRÖßER DER ANSPORN

Die letzte Überlegung in Sachen Native Apps steht im Zusammenhang mit der Leistung. Hier trennt sich die Spreu von Weizen, denn für das Tuning von Endgeräten ist der tiefe Griff in die Trickkiste der Entwickler erforderlich. Wer leistungsintensive Applikationen wie Spiele oder Multimedia-Anwendungen entwickelt, möchte nämlich nicht nur die Vorteile der Funktionen des nativen Frameworks nutzen, sondern auch die Hardware-Fähigkeiten des mobilen Geräts. Erstrebenswerte Funktionen sind beispielsweise die von der Handy-Hardware bereitgestellte Grafikbeschleunigung, das Arbeiten mit schnellen, zusätzlich zu bestimmten CPU-Funktionen integrierten Mathematikbibliotheken und im Extremfall sogar die Nutzung des Assemblers. Keine Frage, dass Anbieter bei solchen Zaubereien für jede anvisierte Plattform eine eigene Native App bereitstellen müssen.

Für Spiele und Multimedia-intensive Anwendungen ist die Spiele-Engine Unity das derzeit am meisten unterstützte Framework. Auch wenn sie sich vor allem für Spiele eignet, kann sie den Aufwand für das Bereitstellen überzeugender 3D-Anwendungen wie etwa virtueller Kataloge auf iOS- und Android-Handys deutlich verringern. Ein optimales Endergebnis lässt aber auch hier nur durch Feineinstellungen erreichen.

## ENTSCHEIDEND BEIM EINSATZ MOBILER WEBTECHNIK IST IHRE FÄHIGKEIT, COMMUNITIES ZU BETREIBEN

Mit innovativen Apps sind Smartphones und Tablets damit in Zukunft nicht mehr nur ein neuer niedrigschwelliger Zugang zu IT, Kommunikation oder Unterhaltung, sondern auch noch jederzeit für mobile CRM nutzbar. Der Aufbau und das Gestalten von Kundenbeziehungen gelten als der Wachstumspfad per excellence im derzeit erst richtig startenden Mobile Commerce. Aus Handys wurden Devices für Community-Betreiber, zu deren Nutzung sich Konsumenten bereitwillig hinreißen lassen. Schon heute sind Google, Facebook, Kreditinstitute oder Telematik-Dienstleister in der Lage, Nutzerprofile anzulegen. Nächster Schritt ist das Analysieren von Verhaltens- und Bewegungsmustern und der Entwurf darauf abgestimmter Vermarktungsstrategien. Sind entsprechende Apps und Infrastrukturen verfügbar, steht auch allen weiteren mobilen Anwendungen wie Augmented Reality oder Extended Packaging nichts mehr im Wege. Die Kunst der Anbieter von Produkten und Services in diesen neuen Märkten wird darin bestehen, sich als vertrauenswürdiger Partner für Community-Operating positionieren zu können.

### **Autoren**

**Giovanni Ambrogio, Geschäftsführer von Live Reply**

**Omar Adobati und Enrico Pertoso, Consultants bei Live Reply**



Live Reply bietet Beratung, Entwicklung und Systemintegration für Telekommunikations- und Medienunternehmen. Das Unternehmen unterstützt seine Kunden bei der Ausarbeitung von Strategien und der Umsetzung neuer Geschäftsmodelle im Value Added Services- und BSS-Bereich. Die Einbindung in das Netzwerk des europaweit agierenden IT-Dienstleisters Reply eröffnet Live Reply den Zugriff auf das Know-how von über 3.400 IT-Experten. Das Unternehmensnetzwerk Reply erwirtschaftete 2011 in seinen Hauptniederlassungen in Italien, Deutschland und im Vereinigten Königreich einen Umsatz von über 440 Millionen Euro. Mehr Informationen unter [www.reply.de](http://www.reply.de).

Live Reply GmbH  
[www.reply.de](http://www.reply.de)