

# Nutzerzentriert

## Was ist die richtige Entwicklungsstrategie für gute Apps?

Giovanni Ambrogio,  
Thomas Stüttgen

**Eine native App für ein spezifisches Smartphone oder einen Tablet-PC oder doch lieber eine mobile Web-App für den mobilen Gebrauch optimierter Websites? Vielleicht aber auch lieber eine hybride App? Wer Anwendungen für mobile Geräte entwickelt, muss sich entscheiden. Die Wahl der besten Entwicklungsumgebung hängt letztlich von der strategischen Zielsetzung ab. Steht diese fest, ist die Frage nach der technischen Plattform in der Regel schon gelöst.**

Der Trend ist eindeutig: Immer mehr Menschen wickeln viele ihrer Aufgaben – gleich ob privat oder beruflich – über Smartphones und Tablets ab. Allein in Deutschland wurden 2011 laut Bitkom rund 11,8 Mio. Smartphones verkauft. Hinzu kommen über 2 Mio. Tablet-PCs. Es ist nur noch eine Frage weniger Monate, bis weltweit die Zahl der Nutzer, die über mobile Systeme auf Internetservices zugreifen, diejenige übertrifft, die den traditionellen Weg über einen PC wählen.

In Deutschland fand dieser Wechsel bereits im vergangenen Jahr statt. Smartphones & Co. führen auf Anwenderseite zu einer neuen Erfahrung in Umgang und Haptik. An die Stelle der (über)füllten grafischen Oberfläche des stationären PC tritt die berührungs- und gestenbasierte Steuerung. Anwendungen werden konsequent aus Nutzerperspektive und nicht mehr prozesszentriert designet. In letzter Konsequenz führt das veränderte Nutzerverhalten zu neuen Anforderungen, wie vor- und nachgelagerte Business-Anwendungen entwickelt und genutzt werden.

Carrier und Content Provider, aber auch Unternehmen allgemein stehen aus diesem Grund zwangsläufig in der Pflicht, sich Gedanken über die „Mobilisierung“ ihrer Services zu machen. Sie müssen einen Weg finden, eigene mobile Apps von der Idee über die Entwicklung und Erprobung bis hin zur Veröffentlichung schnell und zuverlässig her- bzw. bereitzustellen. Mit einer einfachen Übertragung traditioneller Designkonzepte für Websites bzw. Programme ist es nicht getan, da zu den Anforderungen an eine mobile Applikation gerade auch die Einsetzbarkeit in unterschiedlichsten Umgebungen und Situationen gehört. Bedauerlicherweise lässt sich aber auch keine allgemeingültige eindeutige Antwort auf die Frage nach dem perfekten technischen Ansatz und nach

der perfekten Strategie für Apps finden.

Mit der Antwort eng verbunden ist die Frage, ob eine native App oder eine mobile Web-App der bessere Ansatz ist. Die mitunter heiß geführte Diskussion setzt allerdings am falschen Ende an. Die Entscheidung über den Weg der Realisierung hängt zuvorderst von der Zielsetzung ab, die ein Unternehmen mit Hilfe einer App umsetzen möchte. Als erstes sollte daher die Frage des geschäftlichen Wertes beantwortet werden. Welche Zielgruppe soll erreicht werden? Wer profitiert wie von der App? Ist jemand bereit, für die App zu bezahlen? An welchen Stellen ist die App relevant für die eigenen geschäftlichen Ziele? Gibt es Vorteile, die unabhängig von Einnahmen für eine Realisierung sprechen?

Im Rahmen dieser Überlegungen klärt sich quasi automatisch die Frage sowohl nach der Geräteplattform als auch nach dem Weg, wie die Zielgruppe am besten zu erreichen ist. Wenn z.B. der Nutzerkreis hauptsächlich aus Blackberry-Anwendern besteht, sollte eine Applikation nur für diese Zielgruppe entwickelt werden. Ist dagegen offen, welches Endgerät die Zielgruppe nutzt, stellt sich die Frage, ob die wichtigsten Plattformen mit jeweils eigenen nativen Anwendungen unterstützt werden sollen oder ob auf Plattformunabhängigkeit in der Entwicklung gesetzt wird?

Stellt das Unternehmen selbst eine starke Marke dar, muss es sich um die Vermarktung der eigenen App wenig Gedanken machen. Für kleinere Firmen führt dagegen kein Weg daran vorbei, Präsenz auf einem der Marktplätze zu zeigen, um eine gewisse Sichtbarkeit zu erlangen. Diese und weitere Fragen sind in der Konzeptionierung, dem Design und der Umsetzung der mobilen Applikationen zu berücksichtigen.

*Giovanni Ambrogio ist Managing Director, Thomas Stüttgen ist Senior Manager - Head of Value Added Service – bei der Live Reply GmbH in Düsseldorf*

## Native oder Web ...

Eine Stärke nativer Apps ist unstrittig: Sie können direkt auf dem Betriebssystem der jeweiligen mobilen Plattform aufsetzen und so zumindest theoretisch alle technischen Feinheiten der Hardware nutzen. Jede Plattform stellt hierzu unterschiedliche Entwicklungskits, Benutzeroberflächen und Programmiersprachen bereit. Sollen mehrere Plattformen unterstützt werden, ist für jede Plattform eine native App zu entwickeln.

Mobile Web-Apps punkten dagegen mit dem Argument „Write once, run everywhere“. Sie stellen im Grunde eine für den mobilen Gebrauch optimierte Website dar, die wie ihre stationären Vorbilder programmiert werden. Die Zeit- und Kostenvorteile sind offensichtlich, wenn lediglich eine mobile Web-App für alle Plattformen zu entwickeln, zu betreiben und zu pflegen ist. Der Zugang zur Gerätehardware ist allerdings deutlich eingeschränkt. Meist kann eine Web-App auf die Ortungsfunktion und den lokalen Speicher des mobilen Geräts zugreifen, während Kamera oder Beschleunigungsmesser jedoch für sie noch unerreichbar bleiben. Mit dem Vorteil der Plattformunabhängigkeit ist zudem oft der Nachteil verbunden, dass ihr standardisiertes Antlitz nicht vollständig der nativen Benutzeroberfläche entspricht. Eine Reihe oftmals frei zugänglicher Werkzeuge oder Frameworks helfen glücklicherweise, diese Lücke zur nativen Anwendungsentwicklung zumindest ein wenig zu schließen. Dabei unterstützen Frameworks wie The-M-Project/Panacoda oder Sencha Touch/Sencha im Prinzip den gesamten Entwicklungsprozess – von der Benutzeroberfläche bis zur Anwendungslogik. Andere wie jQuery Mobile konzentrieren sich ausschließlich auf den User-Interface-Part.

Vielleicht noch bedeutender als die Frage der Programmierumgebung sind die Aspekte Betrieb und Vermarktung. Eine native App lässt sich ausschließlich über die einschlägigen Marktplätze der jeweiligen Systemplattform vermarkten und unterliegt damit einem zum Teil harschen Abnahmeprozess des Plattformbetrei-

bers. Einer mobilen Web-App ist dieser Vermarktungsweg verwehrt, allerdings lässt sie sich ohne Auflagen von Dritten auf gewöhnlichen Webservern hosten. Der Anwender muss lediglich die richtige URL im Browser des mobilen Gerätes aufrufen und online sein, um die Anwendung zu laden. Während eine direkt installierte native App auch offline auf die Gerätere Ressourcen

und -inhalte zugreifen kann, verlangen Web-Apps bis auf wenige im Cache zwischengelagerte Informationen den permanenten Internetzugang.

### ... oder doch hybrid?

Als dritter Weg steht die Realisierung sog. Hybrid-Apps zur Option. Sie stellen einen mehr oder minder gelungenen Kompromiss zwischen den beiden technischen Hauptströmungen dar. Typischerweise wird dabei eine mobile Web-App in einen Container „verpackt“, der auf dem mobilen Gerät als native App arbeitet. Über diese Middleware erhält die App Zugang zu den spezifischen Funktionen des mo-

bilien Gerätes, die einer gewöhnlichen Web-App verwehrt sind. Im Kern bleibt der Code für alle Plattformen unverändert. Der Vorteil: Die mobile

Feature	Web-App	hybride App	native App
Nutzererfahrung	☑☑☐	☑☑☐	☑☑☑
einfache Entwicklung	☑☑☑	☑☑☐	☑☐☐
Funktionsreichtum	☑☑☐	☑☑☑	☑☑☑
Kosten	☑☐☐	☑☑☐	☑☑☑
Zugangsgeräte-APIs	☐☐☐	☑☑☐	☑☑☑
Offline-Verfügbarkeit	☑☐☐	☑☑☐	☑☑☑
Sicherheit	☑☐☐	☑☐☐	☑☑☑
Plattform-Range	☑☑☑	☑☑☐	☐☐☐
einfaches Upgrade	☑☑☑	☑☑☐	☑☐☐

Keine App liefert alle Vorteile auf einmal

(Quelle: Reply)

Web-App wird vom Anwender genau wie eine native App wahrgenommen, während die meisten Vorteile einer mobilen Web-App erhalten bleiben.

Das vermutlich bekannteste Beispiel einer solchen Hybrid-App-Umgebung ist aktuell das Open-Source-Projekt PhoneGap, dessen Weiterentwicklung unter der Bezeichnung Cordova von der Apache Software Foundation fortgeführt wird. Meist im gleichen Atemzug mit PhoneGap werden Entwicklungs-Frameworks wie Titanium/Appcelerator oder Rhodes/Rhobile (jetzt Motorola Solutions) genannt. Im Unterschied zu dem Container-Ansatz handelt es sich hier eher um Cross Compiling. Aus einer gemeinsamen Code-Basis werden native Apps für verschiedene Zielplattformen erzeugt, und zwar ohne dass eine Zusatzschicht vonnöten ist. Im Vergleich zum Container-Verfahren sind diese Anwendungen funktional reichhaltiger und leistungsfähiger, da sie direkt auf dem Betriebssystem aufsetzen. Cross-kompilierte Anwendungen sind letztlich nativen Apps gleichzusetzen, die anstelle der vom Plattformhersteller vorgesehenen Umgebung eine plattformneutrale Sprache zur Programmierung nutzen.

Das Versprechen einer Plattformneutralität können die angeführten Entwicklungswerkzeuge allerdings nur in einer idealen Technikwelt einlösen. In der Praxis behindern viele Detailfragen die problemlose Umsetzung. Schon im Fall der Web-Apps muss man sich der Tatsache stellen, dass nicht jeder Browsertyp auf allen Mo-

Wird fortgesetzt auf Seite 43

### Metro Group Marathon-App

Die Metro Group Marathon-App lässt Zuschauer und Fans hautnah miterleben, wo ihr Lieblingsläufer gerade steckt, welche Zeit er läuft und wie er sich fühlt – vorausgesetzt er postet seinen Gefühlszustand über Facebook oder Twitter. Die App sorgt für die notwendige Verbindung zu Facebook oder Twitter. Genauso stellt sie eine GPS-Verbindung her, wie auch den Zugriff auf das Internetportal, das alle Informationen rund um das Sportereignis zur Verfügung stellt.



*Fortsetzung von Seite 41*

bilgeräten verfügbar ist. Zudem weisen die Implementierungen mitunter erhebliche qualitative Unterschiede auf. Auch wird nicht jede mobile Plattform von hybriden Entwicklungs-Frameworks unterstützt. Als knifflige Angelegenheit entpuppt sich manchmal bereits die Installation, wenn Komponenten betagte Versionen der Plattform-SDKs voraussetzen. Für Code-generierende Frameworks entwickelt sich – insbesondere bei komplexeren Anwendungen – die Speicher-verwaltung zur Achillesferse. Kurz: Die einwandfreie Funktion einer App muss auf jeder einzelnen Plattform getestet werden. Dies ist bei den Entwicklungskosten einer App zu berücksichtigen,

da eine breite Palette an Browsern und Systemen zu Testzwecken angeschafft werden muss. Im Fall hybrider Anwendungen sind zudem lizenzpolitische Restriktionen der einzelnen Plattformhersteller zu beachten.

### **Zuerst eigene Ziele bestimmen**

Auf die Frage „Native App oder Mobile Web-App oder Cross Compiled App?“ gibt es keine definitive Antwort. Die Plattformfrage – gerne vorgeschoben in der Diskussion um geeignete App-Entwicklungsumgebungen – ist im Gesamtbild von nachrangiger Bedeutung. Zumal ein Großteil der Kosten, gerade im professionellen

Bereich, bei der Integration in die traditionellen Backend-Systeme und weniger am Frontend anfällt. Vor diesem Hintergrund ist es für die erfolgreiche App-Strategie eines Unternehmens weitaus wichtiger, sich zuallererst mit den eigenen Zielen und Kunden zu befassen. Dann löst sich die Frage der Plattform meist von selbst. Wenn etwa wie für die im Auftrag der Metro Group von Reply entwickelte Marathon-App (siehe *Kasten* auf Seite 41) für einen fünfstündigen Dauereinsatz ausgelegt werden muss, verbietet sich mit Blick auf die Akkuleistung die Realisierung als reine Browser-App von vornherein. Denn jede zusätzliche Ebene frisst Energie, die zum Ende des Laufes vielleicht fehlt. (bk)