

Mobile User Experience Engineering – Mit Methodik und der richtigen Technologie zu erfolgreichen Apps



Mischa Demarmels
Zühlke Engineering AG
Wiesenstrasse 10a
8952 Schlieren (Zürich), Schweiz
Mischa.Demarmels@zuehlke.com

Katja Neumann
Zühlke Engineering AG
Wiesenstrasse 10a
8952 Schlieren (Zürich), Schweiz
Katja.Neumann@zuehlke.com

Dr. Thomas Memmel
Zühlke Engineering AG
Wiesenstrasse 10a
8952 Schlieren (Zürich), Schweiz
Thomas.Memmel@zuehlke.com

Abstract

Die rasante Verbreitung von Mobile Apps auf immer besserer Hardware hat auch zu einem steigenden Anspruch der Anwender in Bezug auf die gebotene Funktionalität und die Benutzerfreundlichkeit geführt [Vgl. 1]. Schnelle Datennetze und Cloud Dienstleistungen verstärken die Erwartungen der Benutzer nach visuell ansprechenden und interaktiven Apps, die ortsunabhängig funktionieren und überall den Zugriff auf Daten ermöglichen. In diesem Kontext wird die Bedeutung von User Experience (UX) im Entstehungsprozess einer App immer größer. Das Erreichen einer hohen UX ist dabei eine Herausforderung, die von vielen Faktoren beeinflusst wird. Design und gute Bedienkonzepte sind ebenso wichtig wie die enge Verzahnung mit End-to-End Software Engineering. In diesem Beitrag wird das Engineering von Mobile Apps anhand von abgeschlossenen Praxisprojekten beleuchtet. Die gesammelten Erfahrungen werden im Kontext von UX diskutiert.

Keywords:

/// Mobile, User Experience
/// Usability
/// Software Engineering
/// Praxisbericht

1. User Experience und Mobile Technologien

Wegen des rasanten Erfolgs von mobilen Endgeräten und der angeschlossenen Ökosysteme haben sehr viele Unternehmen in den letzten Jahren mit Hochdruck eigene Apps veröffentlicht. Wer UX nicht von Anfang an berücksichtigt hat, wurde in vielen Fällen von schlechten Rezensionen oder ausbleibendem Interesse überrascht. Die Benutzer-Feedbacks in den öffentlichen App Stores und die damit verbundenen Risiken für die Reputation eines Unternehmens forcieren die Einbindung von UX Expertise in den Entwicklungsprozess. Kunden und Fachabteilungen rufen inzwischen sogar selbst den Bedarf an UX Design und Methodik aus. Wer hätte das vor wenigen Jahren für möglich gehalten? UX-Experten hatten in der Vergangenheit oft einen eingeschränkten Wirkungskreis und konnten ihre Methoden nicht nachhaltig in die Wertschöpfungskette integrieren. Für unser Fachgebiet sind Apps eine fantastische Entwicklung. Endlich ist das Thema UX bei allen Entscheidungsträgern angekommen. Und auch im Volksmund ist

es zum Synonym der ganzheitlichen Erfahrung mit einem Produkt oder einer Applikation geworden. Man könnte von einem Zeitalter der Mobile UX sprechen, die der Thematik einen neuen Schub verliehen hat – zum Vorteil der Anwender und Unternehmen. Letztere haben verstanden, dass nicht nur die Usability einer App Auswirkung auf deren Wahrnehmung und Erfolg beim Benutzer hat. Auch der Einfluss vieler weiterer Faktoren, wie beispielsweise die Qualität der Hardware-Materialien, die vom Hersteller kommunizierte Werte oder das in der App reflektierte Image eines Unternehmens [2] spielen eine wichtige Rolle.

Das iPhone war das erste Smartphone, das für viele UX-Aspekte, darunter sein Design, seine Benutzerführung und seine Performance, sehr gelobt wurde. Zudem erzielte das iPhone eine Benutzer- und Markenakzeptanz, die es so zuvor nicht gegeben hat [3; 4]. Andere Plattformen haben aber zwischenzeitlich stark aufgeholt. Die Android-Plattform hat iOS weltweit bereits als meist genutztes Operating System (OS) für Smartphones abgelöst, während Windows Phone noch nicht zu den acht

meistgenutzten Betriebssystemen der Welt zählt [5; 6].

Die Fragmentierung des Mobile-Marktes ist Fluch und Segen zugleich. Für die Anwender ist der Wettbewerb erfreulich, da UX im Kampf um die Gunst der Kunden einen hohen Stellenwert einnimmt. Unternehmen stehen jedoch vor der Entscheidung, für welche Plattform(en) sie eine App entwickeln sollen. Eine Rolle spielt hier auch, dass das Vorhalten notwendigen Know-hows und die Betriebskosten für mehrere verschiedene Plattformen Budget verschlingt.

[Abb. 1]

Während anfänglich der Druck mit Apps auf den Markt zu kommen die Investitionsbudgets sprudeln ließ, verändern sich die Mobile-Strategien nach und nach in Richtung Cross-Plattform Entwicklung. Dieser Ansatz stellt die Verwendung des gleichen Codes für verschiedene Plattformen in Aussicht. Bei diesem Szenario gibt es jedoch starke Abhängigkeiten zum UX Design, denn trotz gleichen Codes erwarten die Anwender weiterhin natives Look & Feel und eine hohe Performance. Um eine App auf mehrere Plattformen zu

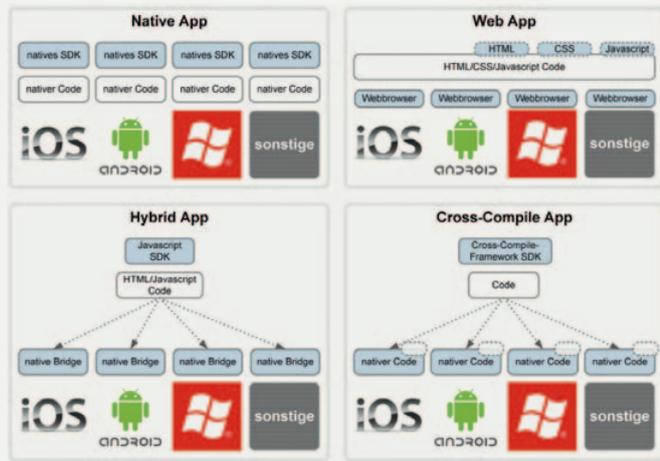


Abb. 1. Vier verschiedene Techniken Apps auf mehrere Mobile-Plattformen zu bringen: Native Apps, Web-Apps, Hybrid Apps, Cross-Compile Apps

bringen, kann man sich somit verschiedener Techniken bedienen. Zum besseren Verständnis für das Zusammenspiel von App-Ansatz und UX werden nachfolgend die vier wichtigsten Techniken vorgestellt: Die Entwicklung von nativen, Web- und Hybrid-Applikationen sowie Cross-Compile Ansätze (siehe Abb. 1).

Wenn man Apps nativ entwickeln möchte, bedeutet dies eine separate Umsetzung für jede Plattform. Vorteil dieses Vorgehens ist, dass die maximale UX erreicht werden kann, da native Apps auf alle System-Features uneingeschränkt zugreifen können und die App ganz auf das Interaktionskonzept und das Look & Feel der Plattform zugeschnitten werden kann. Für eine native App-Strategie bedarf es demnach Spezialisten für jede Plattform. Dass die komplette Applikation für Anzahl x unterstützte Plattformen auch x-mal entwickelt wird, treibt letztendlich auch die Entwicklungskosten in die Höhe.

Klassische Web-Applikationen können durch diverse CSS und Javascript Frameworks (z. B. jQuery Mobile [7], Sencha Touch [8]) für Smartphones und Tablets optimiert werden. Mit dieser Technik können alle mobilen Plattformen auf einen Streich angebunden werden, da die App direkt im mobilen Webbrowser über eine URL gestartet wird. Dadurch verliert man aber die Möglichkeit die Applikation über den App-Store – ein sehr wichtiges

Marketinginstrument – zu verbreiten, wodurch diese Apps eine deutlich geringere Sichtbarkeit aufweisen. Das plattform-spezifische Interaktionskonzept und das Look & Feel bleiben bei dieser Methode notgedrungen mehr oder weniger auf der Strecke. Darüber hinaus kann mit der Web-App-Technologie nur sehr beschränkt auf System-Features zugegriffen werden (z. B. Smartphone-eigenen Kamera, Kontaktdaten, Smartphone-Sensoren, etc.).

Einige Nachteile von Web-Apps werden von Hybrid-Apps dadurch kompensiert, dass ein spezielles Framework (z. B. PhoneGap [9]) für jede Plattform einen nativen Wrapper zur Verfügung stellt. Dieser reicht die Plattform-Features weiter, wodurch diese vom eigentlichen Programmcode der App initiiert werden können, z. B. über Javascript-Methodenaufrufe. Dadurch muss der eigentliche Code der App für alle Plattformen nur einmal mit HTML/ Javascript geschrieben werden. Das hat jedoch auch wieder zur Folge, dass das native Look & Feel kaum für jede Plattform erzielt werden kann.

Der Cross-Compile Ansatz (z. B. Mono-Touch [10]; Titanium [11]) bietet den Entwicklern von Apps ein eigenes Framework mit einer dedizierten „Mainstream“-Programmiersprache (z. B. Javascript oder C#). Dieser Code wird vom Framework automatisch in den jeweils nativen Code der zu unterstützenden Plattformen übersetzt.

Dieser technisch sehr vielversprechende Ansatz birgt aus UX-Perspektive aber den Nachteil, dass nicht auf die plattform-spezifischen Applikations- und Interaktionskonzepte eingegangen werden kann, da die Apps auf allen Plattformen den gleichen Code teilen. Zum Teil wird diesem Problem begegnet, indem neben dem gemeinsamen Code (z. B. Business Layer und Backend-Anbindung) manuell nativer Code für das UI der einzelnen Plattformen in die Codebasis eingepflegt wird. Dies bildet dann einen Kompromiss zwischen nativer und Cross-Compile Entwicklung (siehe [10]). Erfahrungen aus entsprechenden Projekten zeigen, dass mit dieser Herangehensweise und bei geeigneter Softwarearchitektur immerhin 50% Code Reuse erzielt werden kann.

2. Projekt-Steckbriefe

In diesem Abschnitt werden Erfahrungen aus verschiedenen Mobile-Projekten aufgezeigt und diskutiert. Für den Einblick in die Projekte haben wir Interviews mit Entwicklern, UX-Experten und Projektleitern geführt. Insgesamt wurden Erkenntnisse aus sieben Projekten gesammelt, wovon an dieser Stelle drei näher beleuchtet werden sollen.

2.1. My Swisscom App (iOS, Android, Web-App)

Die Ausgangslage des Projekts war ein nur mäßig erfolgreicher Umsetzungsversuch mit einem Cross-Compile Ansatz, der neben technischen Problemen auch nur eine geringe UX aufwies und entsprechend unbefriedigendes Feedback in den App Stores erzielte. Daraufhin wurde eine völlige Neuentwicklung der My Swisscom-App als Native- (iOS, Android) und Web-App beschlossen.

Die My Swisscom App sollte ausgewählte Funktionen zur Verwaltung des persönlichen Portfolios von Swisscom-Privatkunden (z. B. Optionen- und Abo-Verwaltung, Kosten- und Rechnungskontrolle, etc.) zusätzlich zu einem bereits bestehenden Webportal für Desktop Computer als

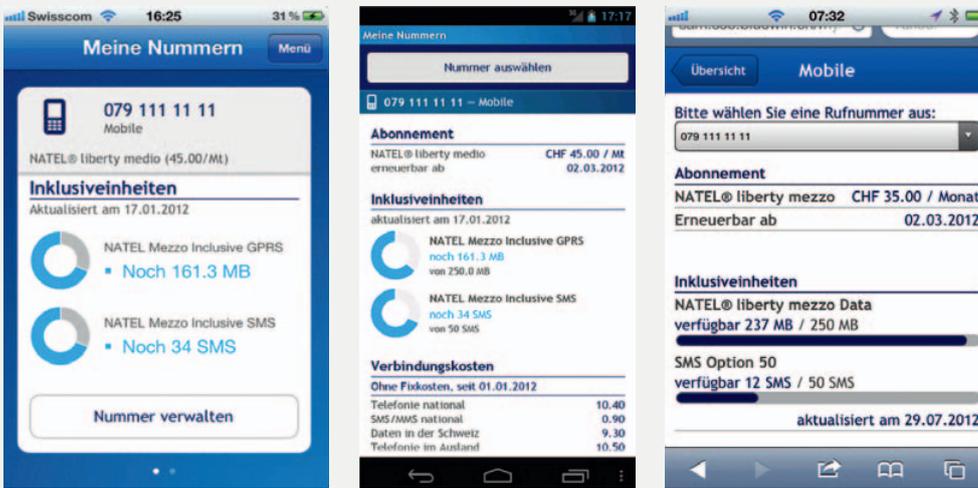


Abb. 2. My Swisscom App für iPhone (links), für Android (Mitte) und als Web-App (rechts). Die dargestellten Daten unterscheiden sich aufgrund leicht unterschiedlicher App-Konzepte etwas voneinander.

mobile App zur Verfügung stellen. Der Vorteil: Kunden haben ihre Kosten jederzeit im Blick und können wichtige Änderungen an ihrem Portfolio von überall vornehmen. Des Weiteren können Anrufe im Callcenter reduziert werden. Damit alle Swisscom Privatkunden gleichermaßen von diesem Angebot profitieren, wurde das App Projekt für die beiden mobile Plattformen iOS [12] und Android [13], die in der Schweiz mit Abstand am weitesten verbreitet sind, gleichzeitig umgesetzt. Zudem wurde für Kunden mit anderen mobilen Betriebssystemen (Windows Phone, BlackBerry, Symbian, etc.) die entsprechende Funktionalität parallel als Web-App [14] entwickelt, auf die beim Besuch der vorhandenen Desktop-Webseite über eine Browsererkennung umgeleitet wird. **[Abb. 2]**

Das Ziel des Projektes war eine signifikante Steigerung der Nutzerzufriedenheit und die End-to-End-Anbindung der App an eine Vielzahl von Backend-Systemen. Dies war notwendig, um den Funktionsumfang der App ausbauen zu können. Die verschiedenen Entwickler arbeiteten zusammen in einem Team, um von erarbeiteten Lösungen gegenseitig profitieren zu können. Diese Zusammenarbeit stellte sich schnell als Vorteil heraus, denn neben der eigentlichen App-Entwicklung musste auch eine serverseitige Komponente implementiert werden, welche die verschiedenen komplex strukturierten Backend-Systeme anbindet, Autorisierungsaufgaben und

serverseitiges Caching übernimmt sowie schließlich die Daten über eine REST-Schnittstelle den App-Clients zur Verfügung stellt. Diese Serverkomponente wurde vom ganzen Team gemeinsam definiert, umgesetzt und wird von allen drei Apps gleichermaßen genutzt.

Während des Projektes hat das Team, auch aufgrund der vorausgegangenen Erfahrungen und Rezensionen von Version 1 (Cross-Compile Variante), eine größere Erwartungshaltung von iOS-Nutzern an die UX antizipiert. Insgesamt stieg der Aufwand für das UX-Design für alle drei parallelen Vorhaben relativ schnell an. So hatten die Projektbeteiligten zunächst vielfach unterschätzt, dass geeignete Interaktionskonzepte für alle drei Plattformen separat erstellt werden müssen. Die Annahme, ein Konzept auf allen drei Plattformen gleichartig umsetzen zu können, führte immer wieder zu technischen Problemen bei der Umsetzbarkeit des UI und letztendlich auch zu Einschränkungen bei der UX der Apps.

Für die Entwicklung der iOS-App bedeuteten einige UI-Designs einen hohen (und oft nicht eingeplanten) Mehraufwand, vor allem wenn diese nicht auf iOS-Standard-Komponenten basierten. Dies konnte schnell zu Verzögerungen führen. Die Android-Entwicklung verlief mit bis zu 50% kürzeren Entwicklungszeiten schneller als prognostiziert. Hier war das

UI-Design von Anfang an enger an die Android-Standards angelehnt.

Die enge Zusammenarbeit zwischen Grafik-Designern, UX-Experten, Requirements Engineers, Business-Vertretern und den App-Entwicklern war aus Sicht unserer Interviewpartner für das Projekt sehr wichtig und wird für künftige Releases noch weiter verstärkt. Damit Grafik- und UI-Designer nicht an den Möglichkeiten und Realitäten einer Plattform vorbeiarbeiten, sind ein interdisziplinäres Team und eine starke (agile) Kommunikation erfolgsentscheidend.

2.2. Delta Energy System App (Hybrid App)

Delta Energy System ist ein weltweit agierender Anbieter von innovativen Stromversorgungslösungen. Zusätzlich zu der bereits vorhandenen umfangreichen Laptop-Lösung mit Web-Schnittstelle und einem kleinen LCD-Display am Gerät, soll den Wartungsmitarbeitern eine Smartphone Applikation für die effiziente Durchführung von routinemäßigen Wartungsarbeiten und die einfache Verwaltung mehrerer Anlagen zur Verfügung gestellt werden. Das Ziel dieses Forschungsprojektes bestand darin, eine geeignete Technologie für diese Smartphone Anwendung zu evaluieren und anhand eines Prototyps deren Tauglichkeit für den Feldeinsatz zu überprüfen.

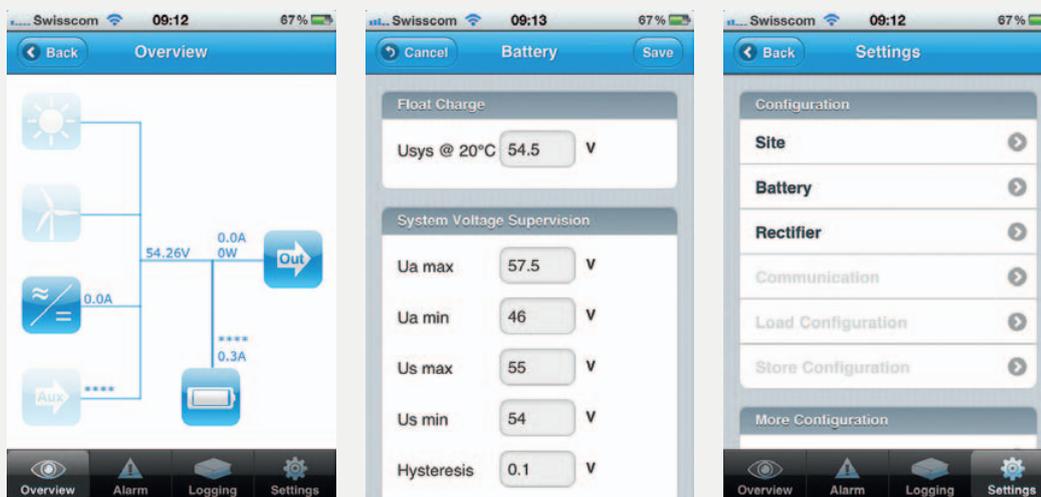


Abb. 3. Delta Energy App für das iPhone entwickelt mit einem Hybrid-Ansatz

Für den Auftraggeber war eine hohe UX ein entscheidendes Ziel. Gleichzeitig gab es Anforderungen zur Wart-, Adaptier- und Portierbarkeit der App. Auch die Verbreitung der verschiedenen Smartphones im Weltmarkt spielte eine entscheidende Rolle bei der Technologiewahl. Zudem sollte die Applikation schnell auf individuelle Anlagen anpassbar sein. Aus diesen Gründen fiel die Entscheidung auf eine Hybrid-App Lösung mit HTML5, JavaScript, jQuery Mobile und PhoneGap als Umsetzungstechnologien. So konnte eine Codebasis für alle Plattformen verwendet werden und die Entwickler von Delta konnten auf vorhandenem Web Know-how aufbauen. Der Fokus der ersten Version lag auf der Unterstützung für das iPhone. [Abb. 3]

Die Entscheidung für den Hybrid-Ansatz war letztendlich ein Kompromiss zwischen effizienter Entwicklung, Erweiterbarkeit und einer guten UX. Trotz grosser Anstrengungen, die App mit CSS3 auf iPhone Look zu trimmen, reagiert die Anwendung im Vergleich zu einer nativen Lösung langsamer. Dies ist mehr auf den Reifegrad und die Implementierung des Frameworks (hier jQuery Mobile) zurückzuführen als auf die Verwendung von HTML5/JavaScript. Bei einer Portierung der App z. B. auf Android-Geräte, müsste entweder das UI der App für diese Plattform angepasst werden oder man müsste mit dem iPhone Look & Feel auch in einer eventuellen

Android-App Vorliebe nehmen, was, wie hier, für Business-to-Employee (B2E) Apps vermutlich akzeptabler ist als für Business-to-Consumer (B2C) Apps.

Aus Sicht der Entwickler scheint klar, dass eine hohe mobile UX nur mit nativen Lösungen möglich zu sein scheint, da es mit HTML5 nur CSS3 als Möglichkeit gibt, das Screendesign zu optimieren, womit auch alle Zielplattformen gleichzeitig abgedeckt werden müssen. Die Entwickler dieses Projektes haben Zweifel, dass man mit der HTML5-Cross-Plattform Lösung (bzw. Hybrid-Apps) wirklich „schneller“ entwickelt, da sowohl iOS als auch Android jeweils eine sehr gute Tool-Suite anbieten, um Software für die entsprechende Plattform zu entwickeln, wohingegen die existierenden HTML5-Frameworks zur Zeit des Projektes zum Teil noch etwas unausgereift waren und dementsprechend häufig vorhandene Unzulänglichkeiten mit Workarounds umschiffen werden mussten. Ein Beispiel hierfür ist die Implementierung von Tabs und scrollbarem Content zugleich – jedes Interaktionselement für sich kann problemlos mit jQuery Mobile umgesetzt werden, möchte man beide allerdings zusammen einsetzen, d. h. scrollbarer Content in einer Tab-Struktur, dann ist die Implementierung zum Erreichen einer guten UX ohne Ruckeln der Tabs sehr aufwändig. Deshalb könnte man vermutlich auch eine native iPhone-App schneller entwickeln als eine Hybrid-Lösung,

insbesondere mit der neuen iOS Storyboard Technologie, mit der relativ einfach mehrere Screens einer Applikation (inkl. der Übergänge zwischen den Screens) auf einmal mit dem GUI-Designer „entwickelt“ werden können [15, 16]. Für eine weitere native Android-App bräuchte man eventuell auch nicht viel mehr Zeit (abhängig natürlich vom Umfang und Inhalt der jeweiligen App).

Letztendlich würden die Entwickler HTML5-Crossplattform Lösungen aktuell nur empfehlen, wenn bewusst gewisse Abstriche beim UI und bei der UX in Kauf genommen werden können, was beispielsweise bei gewissen B2E Applikationen durchaus akzeptabel ist.

2.3. XING App (Windows Phone)

Die Entwicklung der XING-App für das Windows Phone geschah in enger Zusammenarbeit von Zühlke, Microsoft und XING. Die Qualität und die Reichweite der Applikation standen dabei im Vordergrund. Bei einer B2C App entscheiden die Benutzer über den Erfolg der Applikation – bei nativen Apps sind dies z. B. die Bewertungen im Store. In diesem Zusammenhang waren eine hohe UX und gute Rezensionen sehr wichtig. Natürlich kommen bei einer XING App auch hohe Performance-Anforderungen und Skalierbarkeit hinzu – vom gelegentlichen



XING-Benutzer bis zum Power User mit tausenden Kontakten existiert ein breites Spektrum. Gleichzeitig war das Ziel in Kooperation mit Microsoft eine Top-App für die neue Windows Phone-Plattform zu entwickeln, die einerseits die gleichen inhaltlichen Features wie die anderen XING Apps bietet und andererseits das spezifische Windows Phone Interaktionsdesign gut umsetzt (siehe Abb. 4). Neben der Kundenseite war in diesem Projekt somit auch der Plattformhersteller involviert – eine spannende und außergewöhnliche Konstellation. [Abb. 4]

Unter den größten Herausforderungen in diesem Projekt waren die Erreichung einer hohen Performance, die Umsetzung der Metro Design Guidelines – hier insbesondere die Gestaltung des Live Tiles – sowie die Sicherstellung einer guten UX mit Hilfe von zahlreichen Reviews, Tests und der kontinuierlichen Verarbeitung von Benutzer-Feedbacks im Anschluss an die Veröffentlichung der App.

Hinsichtlich der Performance war es in diesem Projekt nützlich, nicht nur die tatsächliche Performance zu optimieren, sondern auch die wahrgenommene Performance zu berücksichtigen, da für den Benutzer letztendlich die wahrgenommene Reaktion der Anwendung zählt. Insbesondere bei einer neuen Plattform wie Windows Phone 7 ist es für eine erfolgreiche App unerlässlich, jemanden im Projekt zu haben, der sich mit den Plattformspezifitäten auskennt. Die Live Tile ist eine Kachel, die der Anwender auf dem Startbildschirm von Windows

Phone platzieren kann. Die Kachel bringt idealerweise die interessantesten Informationen aus der App an die Oberfläche und verhindert so das unnötige Starten der App. Für das Design einer Live Tile sollte ausreichend Zeit eingeplant werden, wobei schnelle und zahlreiche Design-Iterationen für ein optimales Ergebnis empfehlenswert sind (in diesem Fall waren es ca. 20 Iterationen) (siehe Abb. 5).

Der größte Nachteil dieser Plattform besteht in ihrer momentan noch sehr geringen Reichweite – sowohl in Deutschland als auch in der Schweiz macht Windows Phone weniger als 5% aller mobilen Betriebssysteme aus [6]. [Abb. 5]

3. Erkenntnisse und Empfehlungen

Die drei vorgestellten Projekte verdeutlichen, dass sich sowohl die Technologiewahl als auch das App-Konzept auf die UX auswirken – und umgekehrt. Aus den beschriebenen Projekten und den Interviews, die wir mit Entwicklern, UX-Experten, Auftraggebern und Projektleitern geführt haben, wollen wir nachfolgend einige Erkenntnisse und Empfehlungen für Mobile Projekte ableiten.

3.1. Technologiewahl und UX

Die Entscheidung für die richtige Mobile-Technologie muss in jedem Projekt neu getroffen werden, da es verschiedene und je nach Kontext und Ziel unterschiedlich

relevante Parameter zu beachten gibt. Dazu gehören zum Beispiel Entwicklungsbudget, vorhandenes Know-how, Funktionalität, UX-Ziele und Anforderungen an das Look & Feel.

Grundsätzlich bieten sich zwei Strategien an: die native Entwicklung getrennt für alle Plattformen oder die Verwendung von Frameworks zur einfacheren Übertragung einer App auf verschiedenen Plattformen. Bei der nativen Entwicklung ist es oft nötig neues Know-how aufzubauen, da es nur wenige UX-Experten und noch weniger Entwickler gibt, die auf allen Mobile-Plattformen gleichermaßen mit Erfahrung aufwarten können. Gerade für iOS ist es noch schwierig erfahrene Entwickler zu finden. Für neue Plattformen wie dem kommenden Windows 8/Metro UI müssen sich auch UX-Experten mit einer weiteren Mobile-Plattform auseinandersetzen.

Der Cross-Compile- und der Hybrid-Ansatz versprechen gleichermaßen eine Senkung der Entwicklungskosten, da wesentliche Teile des Projektes nur einmal entwickelt werden müssen. Aber auch hier gilt: wenn das gewählte Framework neu für die Entwickler ist, brauchen sie auch hier Einarbeitungszeit. Dies wirft die Frage auf, ob in diesem Fall der Cross-Compile Ansatz wirklich in der Lage ist eine native Entwicklung zu übertrumpfen. Ein weiteres Problem besteht darin, dass es momentan viele Frameworks gibt, von denen jedes einzelne Stärken und Schwächen in gewissen Bereichen aufweist. Hier auf das richtige Pferd zu setzen ist nicht einfach



Abb. 4. Ein Ausschnitt der Panorama-Ansicht der XING App für Windows Phone 7



Abb. 5. Die Live Tile der XING App für Windows Phone 7

und erfordert viel Erfahrung und Kenntnisse von allen technischen Lösungen. Unsere Interviewpartner waren sich in einem Punkt einig: Cross-Compile-, Hybrid-App- und Web-App-Ansatz bedeuten klare Abstriche bei der UX im Vergleich zu nativ entwickelten Apps.

Technische Probleme wie Device-Fragmentierung, Versionsunterschiede des Betriebssystems und die Connectivity-Problematik haben im Mobile-Bereich viel stärkeren Einfluss auf die Arbeit des UX-Experten, von dem sehr großes Know-how und Erfahrung erwartet wird. Auch das Testen auf den verschiedenen mobilen Endgeräten spielt eine zentrale Rolle bei der Umsetzung von Multi-Plattform Projekten.

3.2. Mobile-Strategie

Funktionalität alleine reicht bei mobilen Apps nicht aus, wie ein Blick auf die Benutzerkommentare in den verschiedenen App Stores verrät. Die UX einer App spielt eine sehr wichtige Rolle, also z. B. auch, wie eine App und sogar der Anbieter hinter der App von den Benutzern wahrgenommen werden. Apps mit toller UX zu entwickeln ist deshalb weder einfach noch billig. Insbesondere bei der Entwicklung von B2C Apps ist von halbherzig umgesetzten Apps strikt abzuraten, da diese manchmal mehr (Reputations-)Schaden anrichten können als sie Nutzen bringen.

Die verschiedenen mobilen Plattformen (iOS, Android, Windows Phone, etc.) spalten ihre Benutzer oft in verschiedene Sympathie-Lager. Und so kann es auch vorkommen, dass der persönliche Geschmack von Entscheidungsträgern in Mobile-Projekten zu einer bestimmten Plattform- bzw. Technologiewahl führt. Hier ist es auch die Aufgabe von UX-Experten solche Entscheidungen zu hinterfragen und womöglich bessere Alternativen aufzuzeigen.

UX lässt sich nicht vollends vorhersehen, darum ist es wichtig, bei der Strategie von Mobile-Projekten Zeit und Geld einzuplanen, um nach dem ersten Release auf das Feedback von Benutzern reagieren und die App immer wieder verbessern und

weiterentwickeln zu können. Stetige kleine Releases, die eine App verbessern, werden von den Nutzern in der Regel positiv aufgenommen.

4. Fazit

Ein erfahrener UX-Experte mit plattform-spezifischen Kenntnissen ist in einem Mobile-Projekt unverzichtbar und sollte dort eine zentrale Rolle einnehmen. Neben der Anwendung von UX-Methoden ist es auch seine Aufgabe zwischen Fachseite und Entwicklung zu vermitteln und schon bei der Spezifikation der Anforderungen darauf zu achten, dass diese benutzergerecht erhoben werden und auf den verschiedenen Plattformen kostenbewusst umsetzbar sind. Da der UX-Experte oft die Person mit dem besten Überblick über die verschiedenen zu unterstützenden Plattformen ist, trägt er in Mobile-Projekten oft mehr Verantwortung als in anderen Projekten. Es ist auch seine Aufgabe, Business, Grafikdesigner und Entwickler auf die Eigenheiten des Mobile-Umfelds aufmerksam zu machen und entsprechend zu coachen.

Sehr gutes Know-how aller Zielplattformen, der geltenden Style Guides und der Vorgaben beim Interaktionsdesign ist essenziell bei der erfolgreichen Umsetzung von Mobile-Projekten, in denen eine sehr hohe UX angestrebt wird.

Viel mehr als in anderen Projekten muss im Mobile-Bereich der Nutzungskontext berücksichtigt werden. Anders als z. B. bei Desktop-Anwendungen, welche den Kontext des Benutzers gewissermaßen vorgeben, spielen bei mobilen Apps externe Faktoren eine entscheidende Rolle, wie zum Beispiel die Qualität der mobilen Internetverbindung, die aufwändigere Eingabe von Text via Touchscreen und die Unterbrechung der App-Nutzung, z. B. durch eingehende Anrufe.

Die Entwicklung von Apps für Smartphones und Tablets steht immer noch am Anfang. Die Geräte werden immer leistungsfähiger und viele End-to-End-Anwendungen verschieben sich von klassischen Desktop-Anwendungen zu mobilen Use

Cases. Der Stellenwert von B2C und B2E Apps wird weiter zunehmen. Die Entwicklung anspruchsvoller Apps wird auch UX-Experten weiter fordern, aber auch den Stellenwert ihrer Rolle fördern, insbesondere in einem technologisch immer stärker fragmentierten Markt und einer immer UX-bewussteren Benutzerklientel.

Quellen

1. Berkman E. & Hooper, S. (2012). Designing Mobile Interfaces. Sebastopol, CA: O'Reilly.
2. http://www.google.de/imgres?hl=de&sa=X&biw=1440&bih=775&tbm=isch&prmd=imvnsb&tbid=Z_-O6xpMsYYDzM:&imgrefurl=http://blog.procontext.com/2010/03/usability-und-user-experience-unterscheiden.html&docid=vDBaKrUEgu3Q8M&imgurl=http://blog.procontext.com/images/posts2010/user-experience-und-usability.png&w=1372&h=758&ei=snr8T9uKDoOO8wSf5ZHqBg&zoom=1 (Zugriff am 29.07.2012)
3. http://www.time.com/time/specials/2007/article/0,28804,1677329_1678542_1677891,00.html (Zugriff am 29.07.2012)
4. http://www.nytimes.com/2007/06/27/technology/circuits/27pogue.html?_r=1&pagewanted=all (Zugriff am 29.07.2012)
5. <http://www.netmarketshare.com/> (Zugriff am 29.07.2012)
6. <http://gs.statcounter.com> (Zugriff am 29.07.2012)
7. <http://jquerymobile.com> (Zugriff am 29.07.2012)
8. <http://www.sencha.com/products/touch> (Zugriff am 29.07.2012)
9. <http://phonegap.com> (Zugriff am 29.07.2012)
10. <http://xamarin.com/monotouch> (Zugriff am 29.07.2012)
11. <http://www.appcelerator.com/platform/titanium-sdk> (Zugriff am 29.07.2012)
12. <http://itunes.apple.com/ch/app/my-swisscom/id444087594> (Zugriff am 29.07.2012)
13. <https://play.google.com/store/apps/details?id=com.swisscom.myswisscom> (Zugriff am 29.07.2012)
14. <http://www.swisscom.ch/myswisscom> (Zugriff am 29.07.2012)
15. <https://developer.apple.com/technologies/ios5/> (Zugriff am 30.07.2012)
16. <http://www.raywenderlich.com/5138/beginning-storyboards-in-ios-5-part-1> (Zugriff am 30.07.2012)