1 Einleitung

Obwohl es weitaus mehr Native als HTML5 Apps gibt, nimmt die Diskusion was der bessere Ansatz ist immer mehr an Fahrt auf. Beide Wege unterscheiden sich grundlegend. Hat man einen eingeschlagen, kann man nicht ohne weiteres auf den anderen wechseln. Dies erfordert im Normalfall eine komplette Neuentwicklung.

In diesem ersten Teil wollen wir erruieren, was die Kernpunkte bei der App-Entwicklung sind. Diese sind unabhängig von der gewählten Entwicklungsmethode und begegnen einem immer wieder.

1.1 User Experience

Um eine optimale User Experience bieten zu können müssen einige Aspekte beachtet werden. Die Anwendung möglichst responsive sein. D.h., dass die Applikation geringe Antwortzeiten haben muss. Wenn der User eine Aktion wie z.B. eine Touch-Geste (entspricht einem Klick auf dem PC) ausführt dann erwartet er unmittelbar eine Antwort. Schon eine Verzögerung von einer halben Sekunde kann die App träge erscheinen lassen. Wenn längere Prozesse ausgeführt werden, so sollte dies möglichst im Hintegrund geschehen um die Benutzeroberfläche nicht zu blockieren. Ist dies nicht möglich ist es unbedingt nötig den Anwender zu informieren, was gerade geschieht und warum er warten muss.

Ein weiterer wichtiger Aspekt ist das Scrollverhalten der App. Das Design und die Implementierug können noch so gut sein, kann der Benutzer nicht flüssig scrollen so geht die Experience verloren. Das Problem stellt sich vor allem dann, wenn man Listen mit Ëndless Scrolling"hat, so wie man sie z.B. in der Mobile App von Facebook vorfindet. Die Daten müssen schnell genug geladen und verarbeitet werden, so dass wenn der User auch schnell durchscrollt es nicht zu Verzögerungen kommt.

Neben der Bedienung der App muss auch auf das Look & Feel"geachtet werden. Hier haben alle Hersteller von mobilen Betriebssystemen Styleguides an die man sich halten kann, wenn man unsicher ist. Jedoch geht es primär darum, die Bedienung möglichst intuitiv zu halten. Dies erreicht man, indem Symbole, Bedienelemente und Steuerungsgesten mit den Funktionen hinterlegen welche man auf der entsprechenden Plattform erwartet. Z.B. sollte das Pinchen"(Abb. 1) immer für das Zoomen verwendet werden und nicht auf einmal um Dateien zu öffnen. Auch hier geben die Styleguides einem gute Ratschläge.

Nicht zuletzt geht es auch darum, die vorhandenen Ressourcen des Zielgerätes so gut wie möglich auszunutzen. Das bezieht sich nicht nur auf die Hardwareressourcen sondern auch auf die zur Verfügung stehenden APIs. Mit jedem grösseren Systemupdate führen die Hersteller tausende von neuen API-Features ein. Hier den Überblick zu behalten und diese optimal einzusetzen gestaltet sich nicht immer einfach. Glücklicherweise sind die Dokumentationen der mobilen

Plattformen heutzutage deutlich ausgereifter als vor der Zeit des Mobile-Booms und bieten eine Fülle an Beispielen, wie man die Geräte bis an die Grenze ausreizen kann.

Dies sind nur einige allgemeine Punkte, welche in Bezug auf die User Experience beachtet werden müssen. Die Details hängen immer von der Kategorie der App ab. Die Anforderungen an eine Newsapp z.B., unterscheiden sich stark von denen an einen Taskmanager.

Abb. 1. Pinch Geste, Quelle: http://mjanja.co.ke

1.2 Performance

Unter Performance versteht man die Menge an Daten welche eine Applikation unter verwendung einer bestimmten Menge an Zeit und Ressouren verarbeiten kann. Dabei sollte natürlich möglichst wenig von den zwei genannten in Anspruch genommen werden. Heutige mobile Applikationen müssen in der Lage sein, entliche Aufgaben parallel und in kürzester Zeit abarbeiten zu können. Um dies bestmöglich zu bewerkstelligen, müssen den Entwicklern entsprechende Werkzeuge zur Verfügung stehen um das Verhalten der Anwendung zu analysieren und zu optimieren.

Daneben gilt es auch, Fehler aufzuspüren. Es ist kaum möglich, während dem Entwicklungsprozess bereits alles Fehlerfrei umzusetzen. Das Nutzerverhalten ist meist kaum vorhersehbar und kann zu Fehlern in der Applikation führen, welche man nie für möglich gehalten hätte. Aus diesem Grund ist es wichtig, auch zur Laufzeit die Applikation auf Fehlfunktionen überwachen zu können. Die Performance ist auch eng mit der User Experience verbunden und stellt das Fundament für diese dar.

1.3 Zugriff auf Gerätefunktionen

Aktuelle mobile Geräte haben eine vielzahl von Hardwarekomponenten auf engstem Raum integriert. Die zahlende Kundschaft erwartet, dass diese Komponenten vollends ausgenutzte werden um ihnen eine bestmögliche Erfahrung zu

bieten. Deshalb ist es wichtig, dass die verschiedenen Umgebungen den Entwicklern möglichst einfache APIs bereitstellt um auf die einzelnen Komponenten zuzugreiffen. Wir stellen hier die wichtigsten vor.

1.3.1 GPS Das Global Positioning System – oder kurz GPS – wurde bis vor kurzem lediglich zur Navigation eingesetzt. Mit dem durchbruch der Smartphones und Tablets haben sich jedoch diverse andere Einsatzmöglichkeiten ergeben. Von ortsabhängigen Diensten wie Foursquare oder Facebook Places über Reminder welche einen erinnern was einzukaufen ist wenn man den Laden betritt bis hin zu Augmented Reality Apps, alle bauen sie auf dem GPS auf.

1.3.2 Gyroskop / Beschleunigungssensor Das Gyroskop und der Beschleunigungssensor ermöglichen es einem zu ermitteln, wie das Gerät aktuell gehalten wird und in welche Richtung es sich mit welcher Geschwindigkeit bewegt. Somit sind diese Komponenten eine ideale Ergänzung zum GPS. Aber auch isoliert finden sie eine vielzahl von Einsatzmöglichkeiten, vor allem in der Spieleindustrie.

1.3.3 Audio- / Videoinput Kurz mal das Handy zücken um den Moemnt einzufangen und dann auch noch gleicht mit den Freunden auf Facebook teilen. Oder einen lustigen Moment auf Video festhalten und dann auch direkt auf YouTube stellen, damit der rest der Welt auch etwas zu lachen hat. Smartphones und Tablets sind mittlerweile multimediale Zentralen. Die Kamera ist kaum noch wegzudenken aus den Geräten. Deshalb ist die Zugriffsmöglichkeit für Entwickler auf diese Komponenten unverzichtbar.

1.3.4 GPU Der Grafikprozessor oder GPU (Graphics Processing Unit) ist der Grundstein für alle grafischen und multimedialen Ausgaben. Normalerweise wird die Ausgabe vom Betriebssystem geregelt und der Entwickler muss sich nicht gross darum kümmern. Es gibt aber auch Situationen, wo der Zugriff auf die Resourcen der GPU möglichst genau gesteuert werden muss. Dies ist vor allem bei Videospielen der Fall. Da diese mittlerweile einen erheblichen Teil am Markt für mobile Apps darstellen, ist eine Schnittstelle zu dieser Koponente für Entwickler sehr wichtig geworden.

1.3.5 Dateisystem Eigentlich klingt es selbstverständlich, dass eine Applikation Dateien lesen und schreiben kann. Vom PC her kennt man nichts anderes. Jedoch ist die Situation auf mobilen Geräten etwas komplizierter. Die Apps dürfen aus Sicherheitsgründen nur auf eigene Dateien zugreiffen können. Der direkte Zugriff auf Files des Systems oder von anderen Apps muss möglichst vermieden werden. Man will so die persönlichen Daten der Benutzer schützen. Deshalb unterscheidet sich die Funktionalität von der welche man vom PC her gewohnt ist.

1.4 Einnahmemöglichkeiten

Damit eine Applikation die Kunden erreicht, muss eine einfache und schnelle Beschaffung garantiert sein. Wir befinden uns in einer schnellebigen Welt, in der nur wenige Benutzer sich die Zeit nehmen, das Internet nach Apps zu durchsuchen. Ein Benutzer erwartet heute einen zentralen Anlaufpunkt, meistens sogenannte App Stores, über welche einfach und schnell neue Anwendungen heruntergeladen und installiert werden können.

1.5 Cross Plattform Entwicklung

Laut IDC (Abb. 2) sind teilen sich vor allem Android, iOS und Windows Phone den Smartphone-Markt auf. Um eine möglichst breite Masse an Kunden zu gewinnen ist es wichtig, sich auch mit dem Thema Cross Plattform Entwicklung zu beschäftigen. Eine App für jede Plattform separat zu entwickeln bedeutet hohe Entwicklungszeit, -kosten und einen erheblichen Wartungsaufwand.

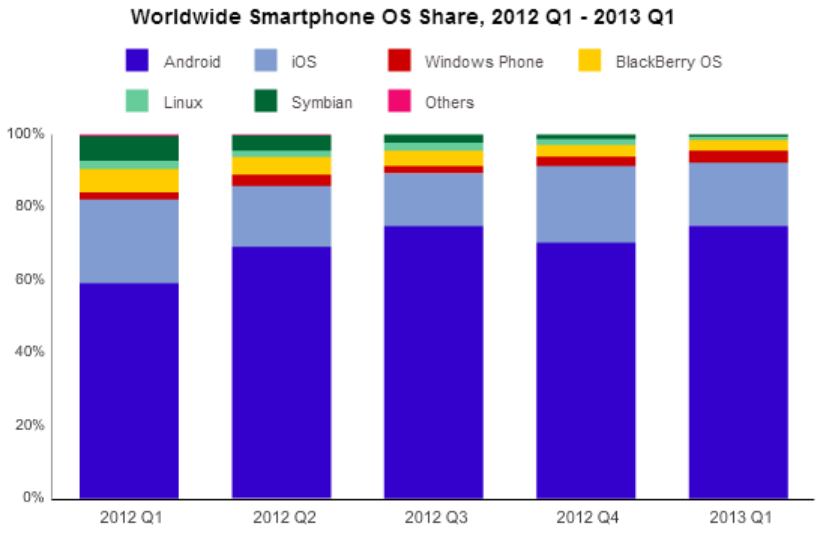

Abb. 2. Marktanteile der Smartphone Betriebssysteme, Quelle: http://www.idc.com/getdoc.jsp?containerId=prUS24108913

1.6 Fragmentierung

Verschiedene Hardware, variierende Betriebssysteme in unterschiedlichen Versionen welche alle einen Webbrowser in variierenden Ausführungen besitzen. Egal ob Web oder Native App; will man eine möglichst breite Masse erreichen muss

man sich auf die Streuung des Marktes ausrichten. Je grösser die Fragmentierung auf der Zielplattform ist, desto höher ist auch der Entwicklungsaufwand. Auch steigt mit der Fragmentierung die Anforderung an das Know-how des Entwicklers. Er muss die Eigenarten und die Feinheiten der einzelnen Versionen der Zielsysteme kennen.

1.7 Verfügbarkeit von Entwicklerresourcen

Ein wichtiges Kriterium zur erfolgreichen Etablierung einer Technik ist auch das Angebot an vorhandenen Entwicklerresourcen. Die Plattform kann nur so erfolgreich werden wie die Entwickler welche sie einsetzen. Auch für Unternehmen welche mobile Anwendungen entwickeln ist es essenziell, dass auf dem Markt genügend Entwickler vorhanden sind um den Support für ihre Anwendungen langfristig sicherstellen zu können.

1.8 Distribution

Ist die Applikation mal fertiggestellt muss dann auch irgenwie zum Verbaucher gelangen. Ein zentraler und bei den Anwendern etablierter Vertriebskanal ist da sehr hilfreich. Je besser und effektiver der Konsument erreicht wird um so höher sind auch die Einnahmen.

1.9 Security

Die Sicherheit ist in der Informatik stets ein aktuelles Thema. Gerade im Zeitalter des Internets und des Smartphones ist es umso wichtiger, ein gutes Sicherheitskonzept zu schaffen. Für viele Konsumenten ist heutzutage das Smartphone die persönliche Nachrichtenzentrale. Telefonieren, E-Mails abrufen, Nachrichten versenden, News lesen und im Internet surfen sind dabei die häufigsten Verwendungszwecke. Immer mehr Personen tätigen zudem ihre Zahlungen per Smartphone-App und durch das Aufkommen von NFC (Near Field Communication) wird auch das direkte Zahlen mit dem Smartphone bald ein Thema werden. Aus diesen Gründen ist die Sicherheit sehr wichtig, da Kontaktdaten, Chat-Verläufe und auch Zahlungsdaten eines Kunden bedroht sind.