

iPhone-Software: Die Hürden beim Entwickeln

VON ROGER BOESCH

Betrachtet man das iPhone aus Anwendersicht, so fallen erstens das schlanke, elegante Display und zweitens die schön animierte Haptik und die Usability auf. Mit dem Device und den bereits auf dem iPhone befindlichen Programmen legt Apple denn auch die Erwartungshaltung der Kunden an Applikationen von Drittherstellern ganz schön hoch. Schaut man sich die Technologie dahinter an, werden schnell weitere Hürden ersichtlich: Objective-C als neue beziehungsweise andere Programmiersprache sowie Cocoa-(Touch), ein wenig bekanntes, aber umfassendes und mächtiges Framework. Was sich also dem Anwender als geniales Gerät offenbart, stellt sich für den Entwickler aus technischer Sicht schnell als der berühmte Wolf im Schafspelz heraus.

Der Prozess

Um die Technologie dennoch in gelenkte Bahnen zu bringen, hat sich die agile Softwareentwicklung noch mehr als in allen anderen Software-Projekten und Technologien zuvor als einzig gangbarer Weg in der iPhone-Entwicklung herausgestellt. Der Markt selbst ist

IN KÜRZE

Das iPhone war und ist sicher einer der wichtigsten IT-Hypes überhaupt. Was für den Anwender ganz Apple-typisch als fast perfekte Symbiose von Hardware und Software daher kommt, ist für den Entwickler leider nicht «Business as usual». Technologien kommen und gehen, was immer bleibt, ist der Anspruch, qualitativ hochstehende Applikationen mit möglichst geringem Ressourcenaufwand zu erstellen. Und gerade dies ist beim iPhone nicht immer ganz einfach.

sehr agil und viele Kunden wissen zu Projektbeginn nicht viel mehr, als dass sie eine iPhone-Applikation möchten und wie diese ungefähr ausschauen soll. Es ist deshalb extrem wichtig, User-Interface-basiert zu arbeiten, weshalb wir schon vor der eigentlichen Entwicklung sogenannte Mockup-Karten benutzen. Diese erlauben es – ähnlich wie bei Spielkarten – den kompletten Ablauf in der zu erstellenden Software vorgängig mit dem Kunden abzustimmen und insbesondere auch die Usability zu verifizieren. Auf Basis der Mockup-Karten wird im Anschluss daran eine Kostenschätzung erstellt und auf Feature-Ebene bewertet. Die anschließende Programmierung stützt sich auf ein Basisframework und setzt darauf aufbauend das UI möglichst detailgetreu und kundenspezifisch um.

Die Kommunikation

Offene Kommunikation ist ein wichtiger Bestandteil für den Erfolg, insbesondere wenn in unterschiedlichen Zeitzonen und Sprachen

gearbeitet wird. Dies ist nicht immer einfach zu realisieren, trotzdem sind bei uns die SCRUM-ähnlichen Standup-Meetings ein Muss, wobei diese natürlich virtuell mit Skype, iChat etc. erfolgen. In diesen Meetings, die auf 15 Minuten beschränkt sind, geht es insbesondere um die Arbeiten des Tages, aber auch um aufgetauchte Fragen und Probleme. Die detaillierte Besprechung und die Lösung erfolgen dann ausserhalb in Peer Meetings. Unterstützend wirkt ein jederzeit zugängliches und flexibles Change Management Tool, das ebenfalls online verfügbar ist und den zweiten essentiellen Punkt bezüglich einer funktionierenden Kommunikation darstellt.

Die Release-Zyklen

Unsere iPhone-Projekte dauern im Durchschnitt ein bis zwei Monate, wobei der Kunde recht schnell einen funktionierenden Prototypen erwartet. Um dem Rechnung zu tragen, erstellen wir in der Regel mindestens drei Kunden-Releases. Release 1: UI Prototyp nach zirka 2 Wochen. Release 2: Nach weiteren 2 Wochen mit den fertig umgesetzten Hauptfunktionen und zu guter Letzt den AppStore Release, der 1:1 der Version entspricht, die in Apples AppStore (iTunes) hochgeladen wird und an dem allenfalls noch Bugfixes vorgenommen werden. Danach ist es essentiell zu beobachten, welche Bewertungen abgegeben werden und welche Features gut oder weniger gut beim Anwender ankommen. Da der durchschnittliche Lebenszyklus einer iApp auf den allervordersten Rängen meistens nur wenige Wochen dauert, gilt es, sie immer wieder zu verbessern und mit neuen und vom Anwender gewünschten Funktionen zu versehen. Wichtig bei dem Thema Release-Zyklen ist zudem, bereits zu Projektbeginn sicherzustellen, dass Releases automatisiert erstellt werden und auch die notwendigen Testpersonen und Zertifikate frühzeitig verfügbar sind. Im optimalen Fall wie bei uns: Mittels Continuous Integration! ■

DER AUTOR

Roger Boesch arbeitet seit nunmehr fast 20 Jahren als Software-Architekt, Technology- und Startup-Coach bei verschiedenen Firmen und Projekten. In seiner Funktion als CEO der Firma iAgentur GmbH war er seit Erscheinen des iPhones bei mehr als zwei Dutzend iPhone-Projekten verantwortlich für deren Umsetzung. Neben umfassendem technischen und langjährigem Management-Know-how verfügt er vor allem über ausgewiesene Erfahrung im Aufbau von Software-Entwicklungsteams.

