



DEN MOBILE-TREND NICHT VERSCHLAFEN

Smartphones und Tablets lösen immer häufiger den PC ab. Hersteller von klassischen Web- oder Desktop-Softwareprodukten müssen auf diesen Trend reagieren, um auch weiterhin den Bedürfnissen und Erwartungen ihrer Kunden gerecht zu werden.

Der Absatz von PCs und Laptops ist seit Jahren rückläufig. 2013 brachen die Absatzzahlen im Vergleich zum Vorjahr um weitere 17,8 % ein, meldet Gartner. Grund für den rückläufigen Absatz ist der Boom von Smartphones und Tablets, die in immer mehr Bereichen des Geschäfts- und Privatalltags eine praktische und preiswerte Alternative zum PC sind. Die Geräte sind handlich und leicht, sofort einsatzbereit, einfach zu bedienen und ermöglichen es, jederzeit und von jedem Ort aus auf Daten und Dienstleistungen zuzugreifen.

„Der Trend zu mobilen Endgeräten stellt Hersteller von klassischen Web- oder Desktop-Softwareprodukten vor eine große Herausforderung“, sagt Christian Moser, User Experience Designer bei Zühlke, „denn die bestehenden Desktop-Lösungen funktionieren nicht auf den mobilen Geräten. Zudem eignen sich viele Webseiten schlecht für die verschiedenen Auflösungen der Bildschirme oder sind nicht für Touch-Funktionen optimiert. Wer heute keine mobile Lösung bietet, läuft Gefahr, Kunden zu verlieren – denn sie erwarten von einem Anbieter, dass er alle Plattformen abdeckt.“

Was die Umstellung auf mobile Endgeräte für ein Unternehmen bedeutet, zeigt das Beispiel eines Herstellers von Zeiterfassungssoftware. Sein bestehendes Produktportfolio umfasst eine Desktop-Lösung, basierend auf Microsofts .NET-Technologien, und einen Web-Client mit reduziertem Funktionsumfang. Der Geschäftsleiter stellt fest, dass er zunehmend Aufträge an Mitbewerber verliert, weil er keine mobile Lösung bieten kann, und beschließt zu handeln.

Produktmanagement

Im Gespräch mit einem erfahrenen Dienstleister erfährt der Produktmanager, worauf es bei der Konzeption einer Mobillösung ankommt: „Eine gute App ist nicht einfach eine Kopie der bestehenden Lösung, sondern eine clevere Adaption des Produkts, abgestimmt auf die Bedürfnisse mobiler Nutzer und auf die Eigenheiten von Tablets oder Smartphones.“

So eignet sich nicht jedes Feature, das auf dem Desktop Sinn macht, auch für eine mobile Lösung (siehe **Abbildung 1**). Bedingt durch die begrenzte Bildschirmgröße, die fehlende Tastatur und die geringe Rechenleistung eignen sich vor allem häufig genutzte Funktionen, die unterwegs einen Mehrwert bieten, wie das Browsen von Daten oder kurze, einfache Interaktionen. Besser auf dem Desktop bleiben jedoch komplexe Dateneingaben, Spezial-

funktionen oder lang dauernde Berechnungen. Bei der Zeiterfassungslösung wären dies das Eintragen der persönlichen Arbeitszeit, das Abfragen des Stunden- und Feriensaldos und für Manager eine Kostenübersicht pro Projekt.

Marketing

Auch das Marketing beschäftigt sich mit der geplanten Mobillösung. Durch den App Store öffnet sich ein neuer Vertriebskanal, über den Kunden die App direkt testen, kaufen, updaten und bewerten. Zudem können über In-App-Käufe zusätzliche Funktionen freigeschaltet werden. Dafür verrechnet der Betreiber des Stores allerdings 30 % des Umsatzes.

Das Marketing muss sich also überlegen, wie das Geschäftsmodell in Zukunft aussehen soll. Denn neben dem Verkauf des

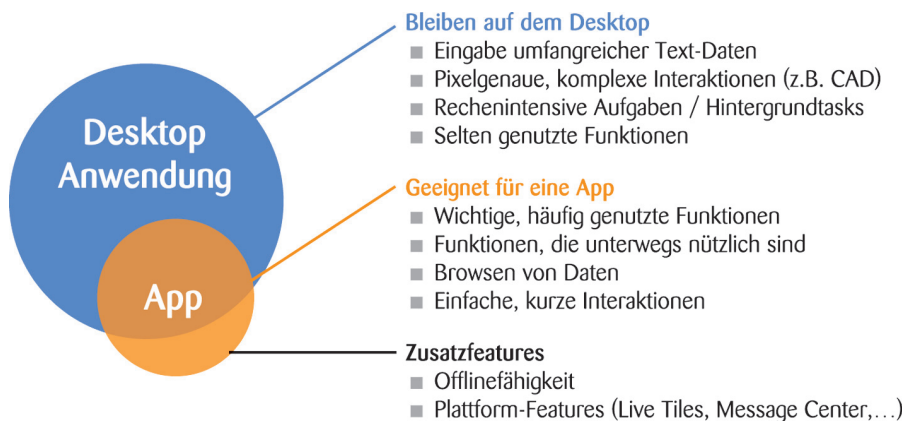


Abb. 1: Nicht jedes Feature ist geeignet für eine App.

Produkts als eigenständige Lösung wäre ein SaaS-Modell (*Software as a Service*) eine interessante Option. Besonders kleinere Unternehmen könnten so auf die Cloud-Lösung zugreifen, anstatt eine eigene Infrastruktur zu betreiben. Dazu gäbe es im Store eine kostenlose App, die sich die Mitarbeiter herunterladen und damit über ein Login auf ihren Firmen-Account zugreifen könnten.

Für das Marketing stellt sich zudem die Frage, welche Plattformen unterstützt werden sollen. Android ist mit über 74 % Marktanteil der größte Player im Smartphone-Markt. iOS schafft es in Deutschland auf 17 %; Windows Phone erreicht knapp 7 %; BlackBerry und Symbian teilen sich den Rest. Ähnlich sieht die Marktverteilung für Tablets aus. Interessant ist hier, dass Windows 8 sowohl auf Tablets als auch auf PCs läuft. Damit können mit einer App mehr Benutzer angesprochen werden. Die Plattform ist zudem noch relativ jung und die Chance ist hoch, eine der ersten Apps dieser Art im Store zu haben.

Das Marketing entscheidet sich für die Entwicklung einer Android- und einer iOS-App. So können mit zwei Plattformen rund 90 % der Kunden erreicht werden. Falls die App ein Erfolg wird, plant das Marketing zudem eine Windows Phone- und eine Windows 8-Version.

R&D

Für den Leiter der R&D-Abteilung stellen die Anfragen des Marketings zunächst eine ziemlich große Herausforderung dar. Denn das Wissen seiner Mitarbeiter, Entwicklungs- und Testwerkzeuge, bestehende Komponenten,

aber auch Lizenzen sind ausschließlich auf Microsoft .NET-Technologien ausgerichtet.

Er sieht drei Wege:

- Sein Team baut Know-how in iOS- und Android-Entwicklung auf und entwickelt alles inhouse.
- Er setzt auf eine Cross-Plattform-Lösung (z. B. mit Xamarin), um möglichst viel Know-how und Code wiederverwenden.
- Er vergibt den Auftrag an einen externen Partner – damit geht er allerdings eine gewisse Abhängigkeit ein.

Die Entscheidung fällt für eine Cross-Plattform-Lösung mit Xamarin, die inhouse entwickelt wird. So kann vorhandenes .NET-Know-how genutzt und ein Teil des bestehenden Source-Codes wiederverwendet werden. Für einen raschen Know-how-Aufbau zieht der R&D-Leiter externe iOS- und Android-Experten hinzu.

Architektur

Auch das Architektur-Team macht sich Gedanken zur geplanten Mobillösung. Bislang baute die Architektur auf eine schnelle, konstante Verbindung zwischen Client und Server. Bei einer Mobillösung ist dies jedoch nicht gegeben. Um den Benutzern trotzdem ein gutes Nutzererlebnis zu bieten, wird ein Vorladen von Daten und Chaches und eventuell ein Offline-Modus unumgänglich.

Auch die Schnittstelle zwischen Client und Server wird sich verschieben. Um rasch weitere Apps auf den Markt zu bringen, sollten möglichst viele Funktionen server-

seitig implementiert oder Teil des gemeinsamen Codes sein.

Bei einer Cross-Plattform-Architektur besteht die Kunst darin, den plattformunabhängigen vom plattformabhängigen Code zu entkoppeln, damit möglichst viel davon geteilt werden kann (siehe **Abbildung 2**). Das reduziert die Entwicklungszeit und den Testaufwand.

Plattformabhängiger Code findet sich in der Regel zuoberst in der Benutzerschnittstelle und zuunterst beim Plattformzugriff (Netzwerk, Sensoren, Dateien, Kontakte). Beide Schichten sollten möglichst dünn gehalten und über Interfaces entkoppelt werden, damit sie leicht ersetzt werden können.

Design

Auch das Design-Team bereitet sich auf das neue Projekt vor und betrachtet dazu die UI-Konzepte von Android, iOS und Windows.

Die Herausforderung beim Cross-Plattform-Design ist, das Look&Feel des Produkts mit demjenigen der Plattform zu verschmelzen. Der Benutzer sollte auf jeder Plattform ein möglichst konsistentes Produkterlebnis haben – die App sollte sich aber trotzdem „native“ anfühlen. Dazu muss die Informationsarchitektur angeglichen werden.

Bezeichnungen und Struktur sollten auf jeder Plattform gleich sein und die Apps sollten ähnlich aussehen. Dazu werden auf allen Plattformen gleiche Farben, Logos und Symbole eingesetzt. Navigation, Menüs, Datenauswahl, Dialoge oder Bildschirmübergänge lassen sich hingegen nur bedingt vereinheitlichen. Zudem gibt es auf jeder Plattform spezifische Features, wie das Message Center bei iOS oder LiveTiles bei Windows, die ebenfalls genutzt werden sollten, um den Benutzern ein möglichst integriertes Erlebnis zu bieten.

Fazit

Tablets und Smartphones boomen. Sie lösen immer häufiger den PC ab – diesen Trend sollten die Hersteller von klassischen Softwareprodukten beachten und sorgfältig prüfen, welche Chancen und Risiken dies für ihr Geschäftsfeld bringt. Nicht für jeden lohnt sich die Entwicklung einer App. Aber wenn Sie sich für diesen Weg entscheiden, beraten wir Sie gerne.

Mehr erfahren Sie auf: www.zuehlke.com

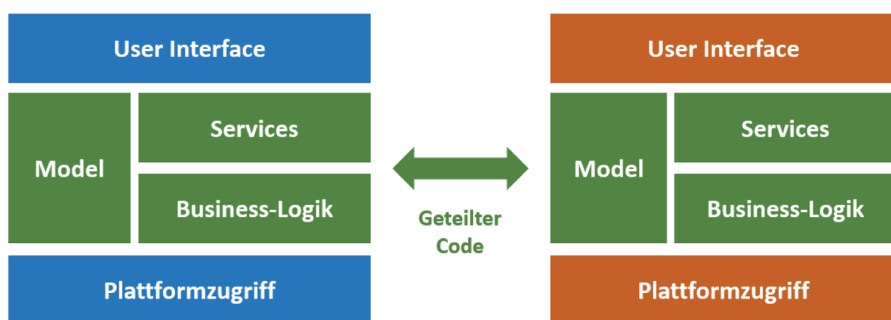


Abb. 2: Eine gute Cross-Plattform-Architektur sollte plattformabhängigen von plattformunabhängigem Code trennen, um möglichst viel wiederverwenden zu können.