

Individuelle IT-Konzepte und Softwarelösungen



Gesellschaft für
Informations- und
Kommunikationssysteme mbH

Vom Prototypen zur Anwendung

**iks-Thementag:
Mobile Applikationen – Ein Leitfaden für Unternehmen**

19.10.2011

Autor:
Ingo Röpling

Agenda

- **Was ist vorab zu beachten?**
- **Skills und Systemvoraussetzungen**
- **Implementierung**
- **Test**

Agenda

- **Was ist vorab zu beachten?**
- Skills und Systemvoraussetzungen
- Implementierung
- Test

Was ist vorab zu beachten? (1)

- **Keep it simple! Die App muss intuitiv bedienbar sein**
- **UI Design Guidelines sind wichtig, da sie intuitive Bedienung unterstützen**
- **Kein Feature Overkill**
- **Keine halbgaren Features**

Was ist vorab zu beachten? (2)

- **Schnelle Antwortzeiten sind notwendig**
- **Eingeschränkte Bildschirmgröße**
- **Eingeschränkte Systemressourcen**
- **Viele unterschiedliche Endgeräte müssen bedient werden**
- **Meist fehlt eine physikalische Tastatur**
 - Eingabe großer Datenmengen daher mühselig
- **Auf welchen Geräten soll es laufen?**

Agenda

- Was ist vorab zu beachten?
- **Skills und Systemvoraussetzungen**
- Implementierung
- Test

Skills und Systemvoraussetzungen – App Typen (1)

● Native App

- Zugeschnitten auf die Zielplattform
- Zugriff auf Systemkomponenten
- Installation notwendig
- Vertrieb über einen App Store möglich

● Web App

- Plattformunabhängig
- Keine Installation auf mobilem Gerät nötig
- Vollständige Kontrolle über die Applikation
- Internetverbindung notwendig
- Keine vollständige Integration in das Betriebssystem
- Läuft im Mobile Browser

Skills und Systemvoraussetzungen – App Typen (2)

● Hybrid App

- Native App, die Webkomponenten lädt
- „Grundgerüst“ wird auf mobilem Gerät installiert
- Die meisten Screens werden über das Internet nachgeladen
- Internetverbindung notwendig
- Teilweise Integration in das Betriebssystem
- Kontrolle über die Webkomponenten
- Grundgerüst kann über App Store vertrieben werden

Skills und Systemvoraussetzungen – Native Apps (1)

● iOS

- Objective-C in Verbindung mit
 - XCode (kostenfrei)
 - JetBrains AppCode (175 €)
- Mac OS X 10.7+
- Kostenpflichtiger Apple Developer Account (ab \$99)



● Android

- JAVA in Verbindung mit Eclipse
- Alle Systeme auf denen Eclipse läuft
- Kostenpflichtiger Android Market Account (\$25, optional)



Skills und Systemvoraussetzungen – Native Apps (2)

● Windows Phone 7.5

- .NET in Verbindung mit Windows Phone Developer Tools
- Windows Vista / Windows 7
- Kostenpflichtiger Entwicklerzugang (\$99)



● BlackBerry

- JAVA in Verbindung mit Eclipse
- BlackBerry WebWorks als Alternative
- Alle Systeme auf denen Eclipse läuft
- Kostenpflichtiger Entwicklerzugang



Skills und Systemvoraussetzungen – Native Apps (3)

● Einarbeitung

- Voraussetzung: OOP Kenntnisse
- Hält sich bei den unterschiedlichen Systemen die Waage
- Selbst bei Sprachkenntnis ist immer noch Einarbeitung in die unterschiedlichen Frameworks (z.B. Cocoa Touch) notwendig
- Sehr umfangreiche Dokumentation vorhanden, Durcharbeiten empfohlen



Skills und Systemvoraussetzungen – Native Apps (4)

● Crossplattformentwicklung

- Erstellung einer App mittels Webtechnologien (HTML 5, CSS 3, JavaScript)
- Toolgestützte Übersetzung in plattformspezifischen Code
 - z.B. PhoneGap
- Steckt jedoch noch in den Kinderschuhen
- Noch nicht die heilsbringende Lösung, die die Branche erwartet

Skills und Systemvoraussetzungen – Web Apps

- **Nicht gerätespezifisch, da die eigentliche Anwendung auf einem Webserver läuft und nur mittels Mobile Browser aufgerufen wird**
- **Programmiersprachen: JAVA, PHP, .NET ...**
- **Als Output: (X)HTML, CSS, JavaScript**
- **Einige Technologien werden jedoch nicht von allen Endgeräten unterstützt**



Skills und Systemvoraussetzungen – Hybrid Apps

- Vereinen die Anforderungen an eine native App und eine Web App
- Vorteile und Nachteile von nativer App und Web App



Agenda

- Was ist vorab zu beachten?
- Skills und Systemvoraussetzungen
- **Implementierung**
- Test

Generelles zum Entwicklungsprozess

- **Der Prozess an sich ist bekannt**
 - Requirements Engineering
 - Prototyping
 - Implementierung
 - Test

- **Fokus bei der App an sich liegt auf Frontend und Speicherverwaltung**

Mock-Up-Phase: Die Idee konkretisieren (1)

● **Mock-Up? Was ist das?**

- Modell der Benutzeroberfläche
- Zeigt nur das Layout und das Design
- Zur Abbildung des Workflows geeignet
- Ausgezeichnetes Werkzeug zur frühzeitigen Präsentation und Diskussion der GUI
- Form kann frei gewählt werden
 - Skizze auf Papier
 - Spezielles Werkzeug
 - etc.

● **Iterativer und vor allem kooperativer Prozess**

- Alle Stakeholder ins Boot holen
- Erhöht die Akzeptanz

Mock-Up-Phase: Die Idee konkretisieren (2)

- Beispiel für Mock-Up und Produkt



Mock-Up-Phase: Die Idee konkretisieren (3)

- **Warum sollte genug Zeit in das Mock-Up investiert werden?**
 - Der Fokus einer mobilen App liegt klar auf der GUI
 - Auf kleiner Fläche müssen ggf. komplexe Workflows mit möglichst einfachen Mitteln abgebildet werden
 - Unklarheiten, Fehler, Schwächen der GUI lassen sich durch schnelles Feedback frühzeitig finden
 - Je später im Entwicklungsprozess eine solche „Unschärfe“ gefunden wird, desto teurer ist deren Behebung
 - Bedürfnisse des Benutzers werden frühzeitig erkannt
 - Dadurch Zeitersparnis bei der Implementierung der GUI

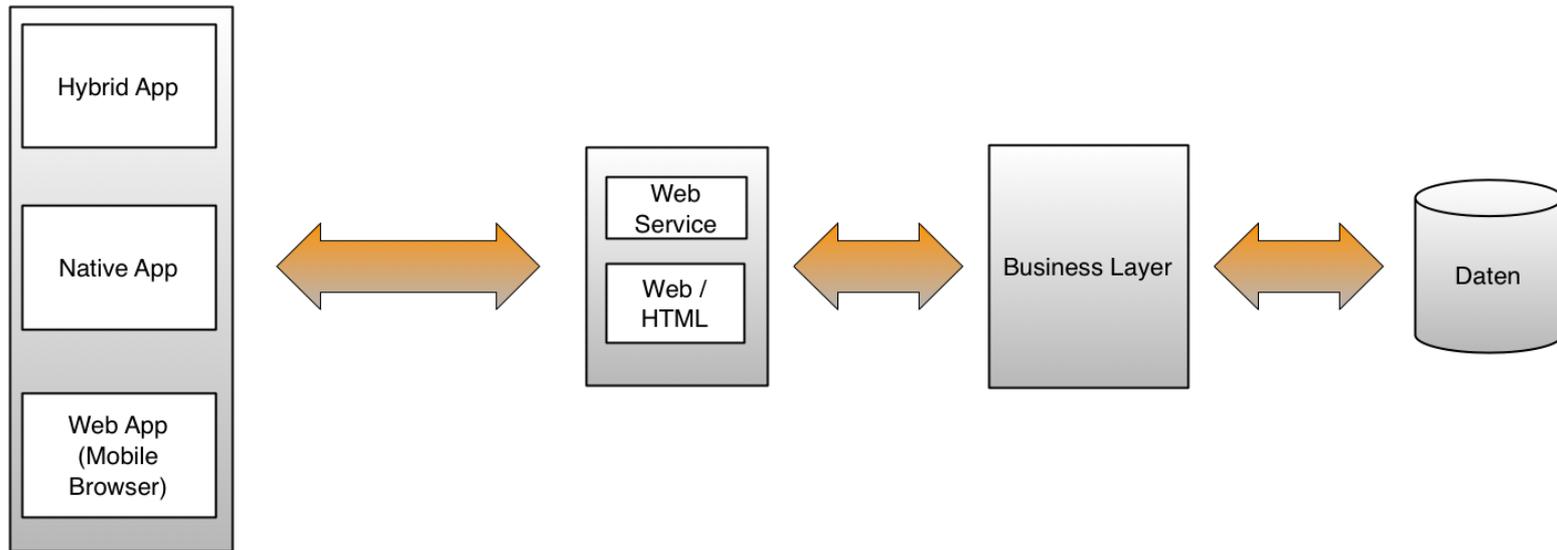
Frontend

- **Komponentenbasierter Ansatz**
- **Frameworks der einzelnen Betriebssysteme bieten vorgefertigte Anzeige- und Bedienelemente**
- **Unter iOS und Windows Phone 7 gilt es die Guidelines einzuhalten**
 - Ablehnungsgrund bei Distribution
- **Für Android gibt es keine festen Guidelines**
- **Orientierung an vorhandenen Guidelines empfohlen, um intuitive Bedienung zu gewährleisten.**
- **Viel Dokumentation vorhanden, die man unbedingt lesen sollte!**

Backend (1)

- **Eigenes Backend nicht immer notwendig (standalone native App oder Third Party Backend)**
- **Falls nicht vorhanden aber benötigt, kann die Implementierung des Backends den größten Teil des Gesamtaufwands ausmachen, da diese ggf. die Geschäftslogik beinhaltet**
 - ggf. Auswirkungen auf Releasetermin
- **Verwendete Technologien sind dabei geräteunabhängig**
 - z.B. REST Schnittstelle
 - Kommunikation via XML über WebServices
 - Kann unterschiedliche Endgeräte mit unterschiedlichen Betriebssystemen bedienen
 - Es werden nur „Anzeigedaten“ übertragen

Backend (2)



Backend (3)

● Vorteile

- Änderungen an Geschäftslogik können ohne Änderungen der App durchgeführt werden, wenn die Schnittstelle gleich bleibt
- Dadurch schneller möglich auf Fehler in der Geschäftslogik zu reagieren
- Unterschiedliche Apps / Apptypen können das Backend benutzen

● Wichtig

- App kann ggf. nur online benutzt werden
- Meist 24 / 7 Betrieb notwendig
- Hosting Skills notwendig
- Skalierungsspezialisten

Agenda

- Was ist vorab zu beachten?
- Skills und Systemvoraussetzungen
- Implementierung
- **Test**

Test (1)

- **Prozess an sich ebenfalls bekannt**
- **Die einzelnen Entwicklungsumgebungen bieten Toolunterstützung**
 - Simulator
 - Debugger
 - Performance Tools
 - Unit Testing
- **Darüber hinaus sind externe Testframeworks verfügbar**
 - z.B. Fonemonkey (iOS), Robotium (Android)
- **Weitere Alternative: Remote Testing**
 - Externer Dienstleister hostet App für Test
 - Jedoch meist nicht sehr performant

Test (2)

- **Wann soll mit dem Testen begonnen werden?**
 - Idealerweise bereits parallel zur Entwicklung
 - Frühzeitiges Finden von Speicherlecks und Performanceproblemen
- **Fokus dabei klar auf Usability (Frontend) und Speicherverwaltung / Performance**
- **Test im Simulator keinesfalls ausreichend**
 - So schnell wie möglich auf „richtigem“ Gerät testen
 - Testgeräte „breit streuen“
- **Review gut geeignete Test- und QM Technik**
- **Herausforderung hierbei: Fragmentierung**

Fragmentierung – Herausforderung vs. Lösung (1)

- **Viele unterschiedliche Geräte und OS Versionen**
 - Unterschiedliche Bildschirmauflösung, Leistungsmerkmale etc.
 - Unklarheit ob der Benutzer das entsprechende OS Update erhält
 - Hersteller trifft die Entscheidung
 - Unterschiedliche Hardwareausprägungen
 - mit / ohne physikalischer Tastatur
 - mit / ohne Kamera
 - Prozessor ...
- **Schwierigkeit der Reproduktion gemeldeter Fehler**
- **Betrifft überwiegend Android**
- **iOS und Windows Phone 7 aufgrund von Regeln weniger betroffen**

Fragmentierung – Herausforderung vs. Lösung (2)

● Herangehensweisen zur Überwindung der Hürde

- Unterstützte Geräte und OS Versionen im Vorfeld eingrenzen
 - Welche Geräte benötige ich für meine Features?
 - Idealfall: Ich entscheide selbst über Geräteanschaffung (Businessanwendungen)
 - Welche OS Version benötige ich für meine Features?
- Testgeräte entsprechend der Vorbedingungen bereitstellen
 - Unterschiedliche Gerätegenerationen abdecken
 - Unterschiedliche OS Versionen abdecken
 - etc.

Referenzen

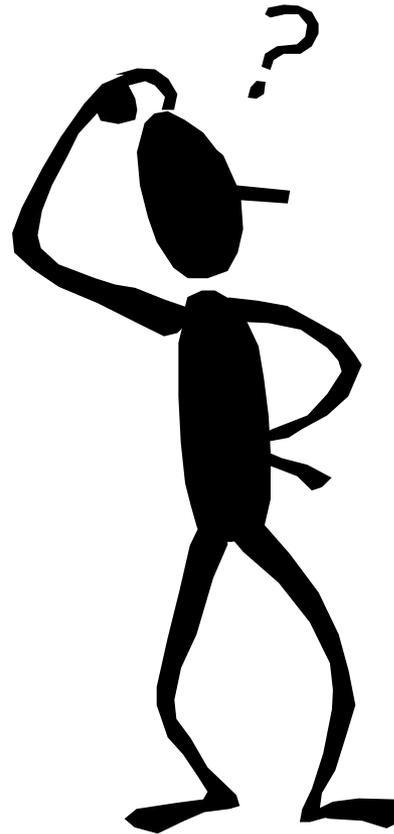
[wiki01]

<http://de.wikipedia.org/wiki/Mock-up>

Weiterführende Literatur

- **Kevin Mullet & Darrell Sano, „Designing Visual Interfaces“**
- **Dave Mark, „Beginning iPhone 4 Development“**
- **Microsoft , „UI Design and Interaction Guide for Windows Phone 7“**
<http://download.microsoft.com/download/7/7/3/77371BBD-6613-4C1A-ACBF-08365C09D5FA/UI%20Design%20and%20Interaction%20Guide%20for%20Windows%20Phone%207%20v2.0.pdf>

Fragen?



www.iks-gmbh.com