

Mobile Apps Workshop

Workshop 6



Überblick

- Referenten
- Inhalt des Kurses
- Termine
- Sourcen
- Abschlussarbeit
- App-Entwicklung allgemein
- Selbsteinschätzung (Javascript, Swift)
- Teams
- Hausaufgabe

Referenten



Roman Rast - Usability Engineer

roman.rast@fhnw.ch

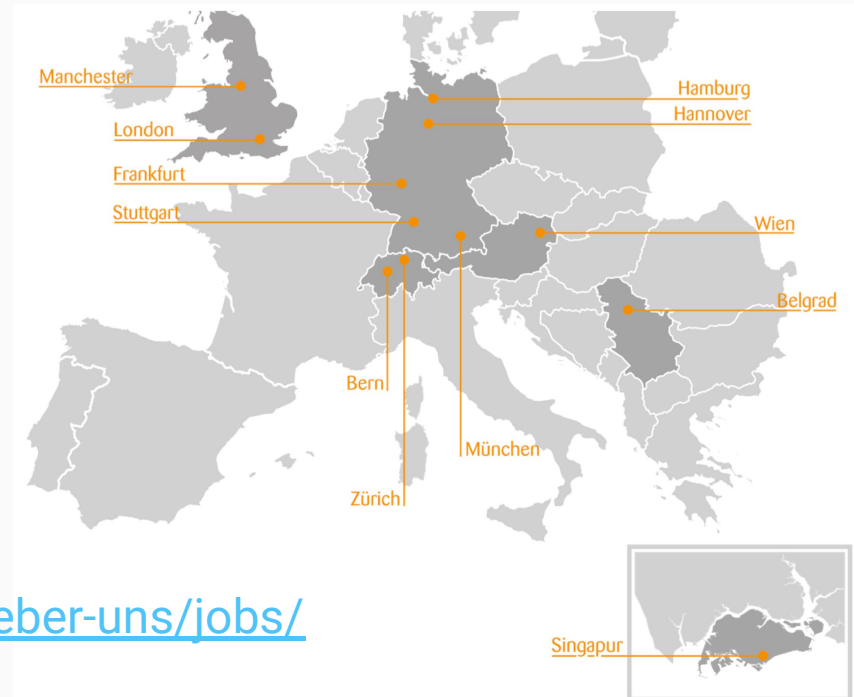


Oliver Gepp - Software Architect

oliver.gepp@fhnw.ch



- Innovationsdienstleister
 - Gegründet 1968
 - Hauptsitz in Schlieren (ZH)
- 4 Standbeine
 - Softwareentwicklung
 - Produktentwicklung
 - Management-Consulting
 - Ventures
- rund 800 Mitarbeiter
- Umsatz 2016: CHF 133 Millionen

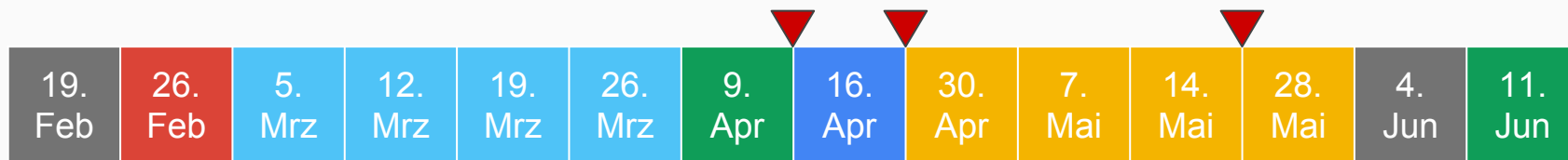


→ <https://www.zuehlke.com/ch/de/ueber-uns/jobs/>

Inhalt des Workshops

1. App-Design
2. Entwicklung mit Ionic
3. App-Stores, App-Icon, Splashscreen
4. Swift & iOS native
5. Wrap-Up, FAQ, Empfehlungen
6. Abschluss-Präsentationen

Termine



Design

Ionic

Zwischenpräsentation -
Prototyp
App-Stores, Icon,
Splashscreen

Swift & iOS

WrapUp, FAQ
Abschlusspräsentation

Sourcen

Sämtliche Unterlagen und Code bei github:

<https://github.com/Zuehlke/fhnw-mobile-workshop>



Abschlussarbeit - Thema Sportturnier

Mögliche Problemstellungen

- Spielverwaltung - z.B. für ein Tennismatch
- Fanclub
- Auslosen von Kombinationen (Töggelturnier,...)
- Weitere Ideen?



- App muss funktionieren und einen Zweck erfüllen
 - Integration eines REST-Interfaces oder einer Datenbank
 - Inklusive App-Icon und Splash-Screen
 - Umsetzung in Ionic oder Swift
- Übergabe ausschliesslich per github
 - git history vollständig (nicht nur ein commit)
 - E-Mail mit repo-url an Workshopleiter senden
- Abgabetermin: 08.06.18 0 Uhr
- Abschlusspräsentation im letzten Workshop
 - Prototyp bei Präsentation zeigen und mit Ergebnis vergleichen
 - Jede Gruppe hat genau 3 Minuten Präsentationszeit
- Kein Copy & Paste
- 2er/3er-Gruppe
- Arbeitsumfang etwa 25 h pro Person

Umfang der Abschlusspräsentation:

- Zeit: 3 Minuten pro Gruppe
- Vorstellung der App
- Prototyp vs. finale App (kurzer Vergleich)
- Was waren die technischen Herausforderungen?
- Was würdet ihr nächstes mal besser machen?

Testat erreicht wenn:

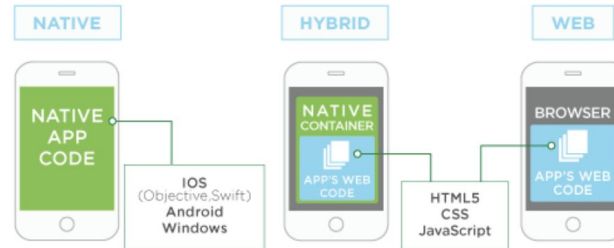
- Zwischenpräsentation Prototyp erfolgreich
- App wurde rechtzeitig eingereicht und ist funktionsfähig
- Abschlusspräsentation App erfolgreich
- 80% Anwesenheit

MOBILE APPS AT A GLANCE NATIVE VS. HYBRID VS. WEB APPLICATIONS



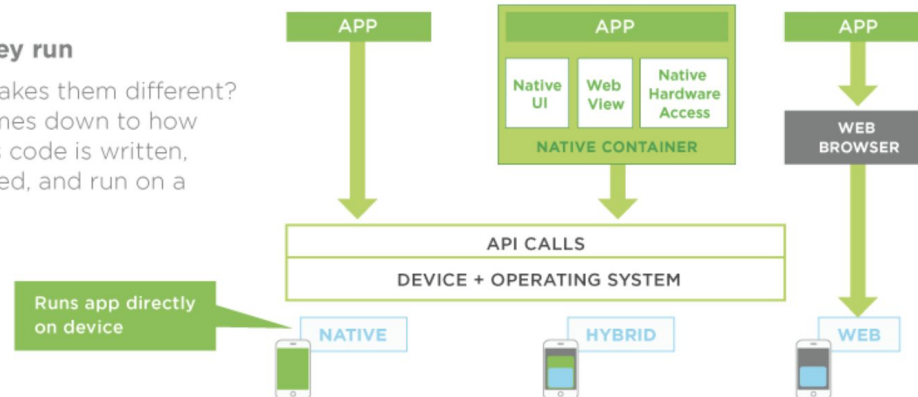
How they're structured

When it comes to building your mobile app, you have three options: native, web, or hybrid.



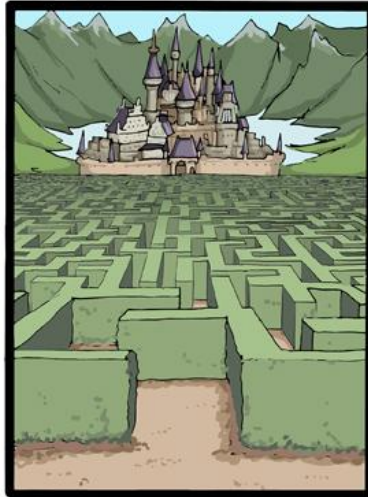
How they run

What makes them different? It all comes down to how an app's code is written, structured, and run on a device.

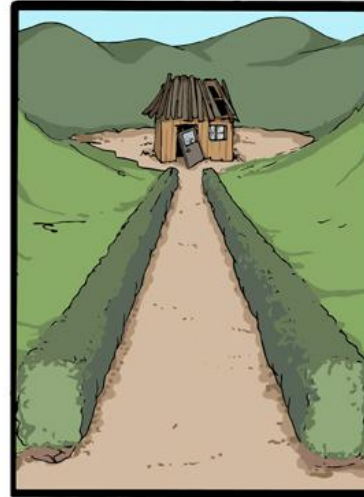


The dilemma of mobile apps development

Develop a native app for each device and maintain several projects



Use a unique framework (Phonegap, Adobe Air, Appcelerator) and maintain only one project



Web Apps

- Mobile Version einer Webseite
- Keine Installation - kein Speicherplatzverbrauch
- HTML5, CSS, JavaScript
- Vertrieb ohne AppStore, ohne Beschränkungen
- Zugriff auf Hardware ist limitiert
- Keine Offlinefunktionalität
- Keine Push-Notifications



Progressive Web Apps

- Relativ Neues Browser Feature
- Service Workers & Cache -> limitierte offline-Nutzung
- Push APIs erlauben Push Notifications
- Vertrieb ohne AppStore, ohne Beschränkungen
- Nicht alle Web App Probleme werden gelöst
 - z.B. plattformspezifische Navigation



Hybrid Apps

- Kombination aus nativer App & Web
- Nativer App Container mit Webbrowser
- Webtechnologien (HTML5, CSS, JS/TS)
- Geringer Aufwand viele Plattformen abzudecken
 - Testaufwand nicht unterschätzen
 - Alte Android-Geräte stossen schnell an Grenzen
- Zugriff auf Hardware erfolgt über Plugins
 - Kamera, Mikrofon
 - Kalender, Kontakte, Fotos
 - Push Notifications
 - Aber: Anpassung von Plugins ist nicht trivial und setzt native Kenntnisse voraus



Cross Plattform Apps

- Entwicklung ähnlich der Entwicklung von Hybrid Apps
 - Eine gemeinsame Codebasis mit Generierung des Codes für die jeweilige Plattform
- Prominenter Vertreter:
 - Xamarin (Microsoft)
 - Entwicklung in C#, Verwendung von .Net
 - Ermöglicht die separate Entwicklung des platform-spezifischen UIs
 - Ermöglicht hohe Wiederverwendung
 - Aber auch keine Silverbullet
 - leichte Abstriche bei Performance
 - begrenzter Zugriff auf OpenSource-Libraries



Native Apps

- Voller Hardwarezugriff und beste Performance
- Entwicklung für jede Plattform separat
 - iOS: Swift (oder Objective C)
 - Android: Kotlin (oder Java)
 - Ebenso spezifisch für legacy Plattformen
 - z.B. Blackberry, Windows, Symbian, ...
- Vertrieb nur über den jeweiligen App-Store
- Erste Ansätze für Code-Sharing sind im Entstehen
 - Zugriff auf Kotlin-Code von Swift
 - Z.B. zur Teilen der Business-Logik
 - Entwickeln von Android-Apps in Swift oder iOS-Apps in Kotlin

Wie gut sind meine Kenntnisse in

- Swift
- JavaScript
- HTML, CSS
- Mobile App Entwicklung

Was erwartest du von diesem Workshop?

Teams & MacBooks

Wer benötigt ein MacBook?

Bitte bei Urs Adam melden: urs.adam@fhnw.ch

Jetzt: Teams bilden

je 2 oder 3 Personen

→ Pair Programming

→ Abschlussarbeit

Team Up!



Order MacBook



Coolest App ever!



Hausaufgabe

Ionic & IDE installieren

<https://ionicframework.com/>

Node.js installieren (LTS Version)

<https://nodejs.org/en/>

Xcode und Android Studio installieren

- Xcode aus App Store auf Mac
- Android Studio:
<https://developer.android.com/studio/index.html>

IDE (kostenfrei)

<https://code.visualstudio.com/>

→ Hello World sollte laufen