Mobile Apps Workshop

Workshop 6

Überblick

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

Referenten

zühlke
empowering ideas

Roman Rast - Usability Engineer

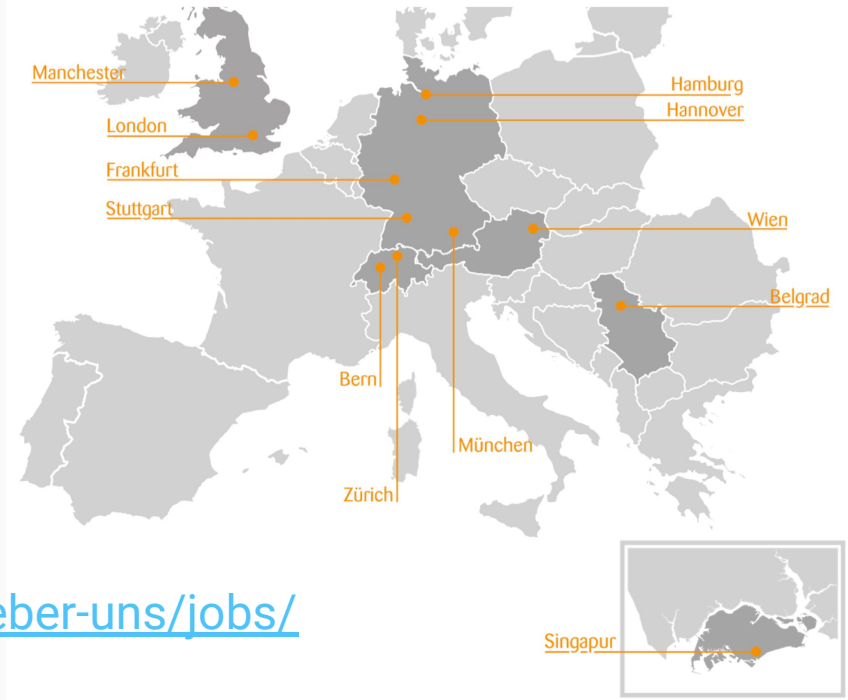
roman.rast@fhnw.ch

Oliver Gepp - Software Architect

oliver.gepp@fhnw.ch

Zühlke - unser Arbeitgeber

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

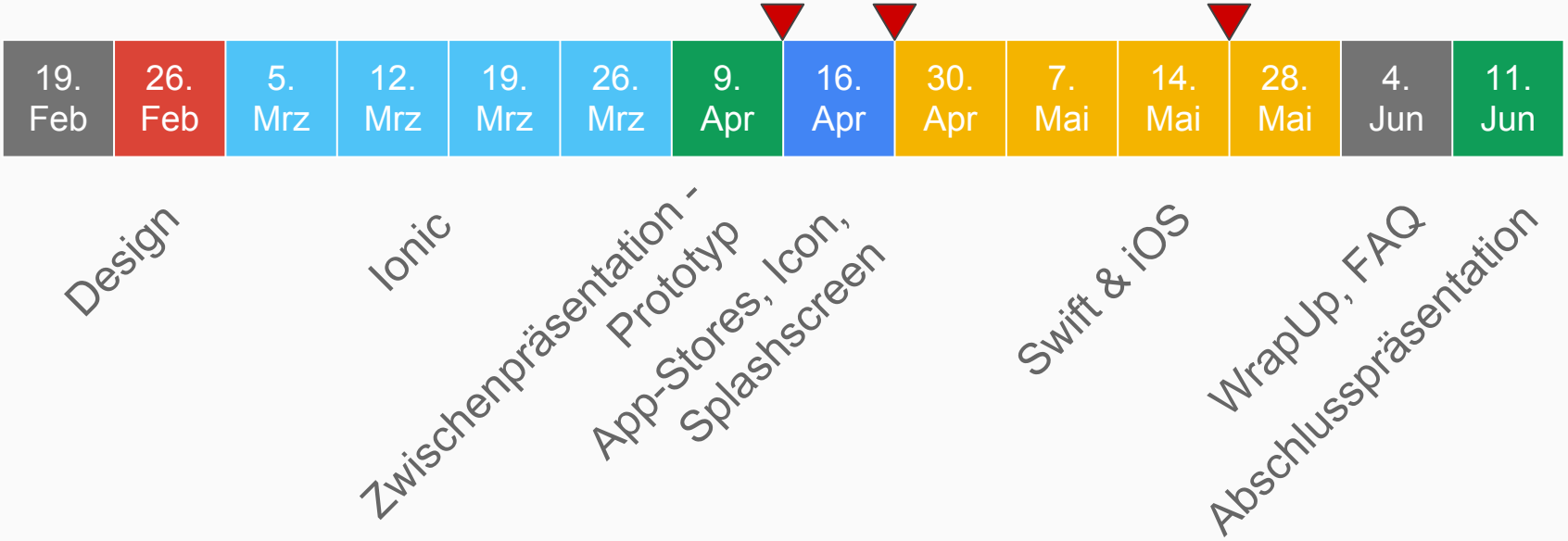


→ https://www.zuehlke.com/ch/de/ueber-uns/jobs/

Inhalt des Workshops

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

Termine


19. Feb
26. Feb
5. Mrz
12. Mrz
19. Mrz
26. Mrz
9. Apr
16. Apr
30. Apr
7. Mai
14. Mai
28. Mai
4. Jun
11. Jun
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 (Töggeliturnier,...)
- Weitere Ideen?

Abschlussarbeit - Rahmenbedingungen

- 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 / bitbucket
  - 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
- Kein Copy & Paste
- 2er/3er-Gruppe
- Arbeitsumfang etwa 25 h pro Person

Abschlusspräsentation

Umfang der Abschlusspräsentation:

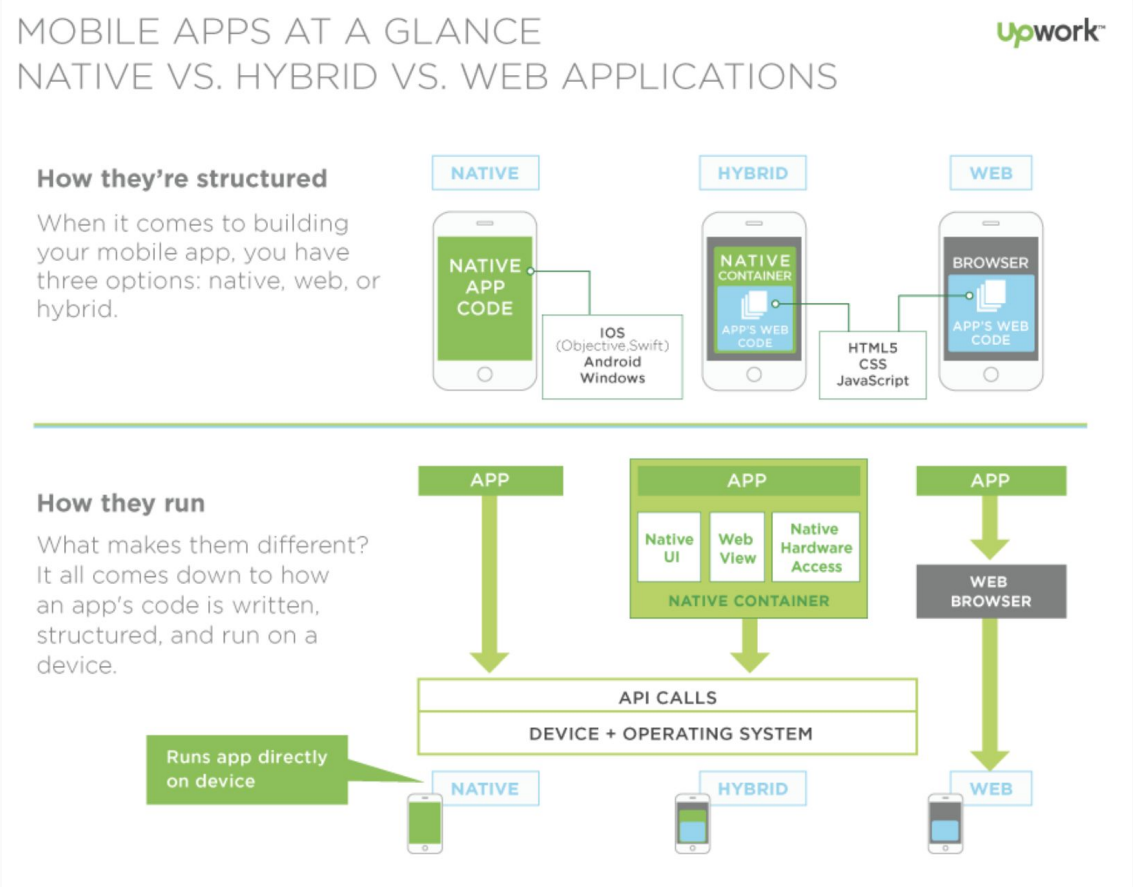
- Zeit: 3 Minuten pro Gruppe
- Vorstellung der App
- Prototyp vs. finale App (kurzer Vergleich)
- Was waren die technischen Herausforderungen?
- Was würdet ihr nächstes mal besser machen?

Testat

Testat erreicht wenn:

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

Überblick App-Entwicklung



Überblick App-Entwicklung

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

Überblick App-Entwicklung

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

Überblick App-Entwicklung

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

Überblick App-Entwicklung

Cross Plattform Apps

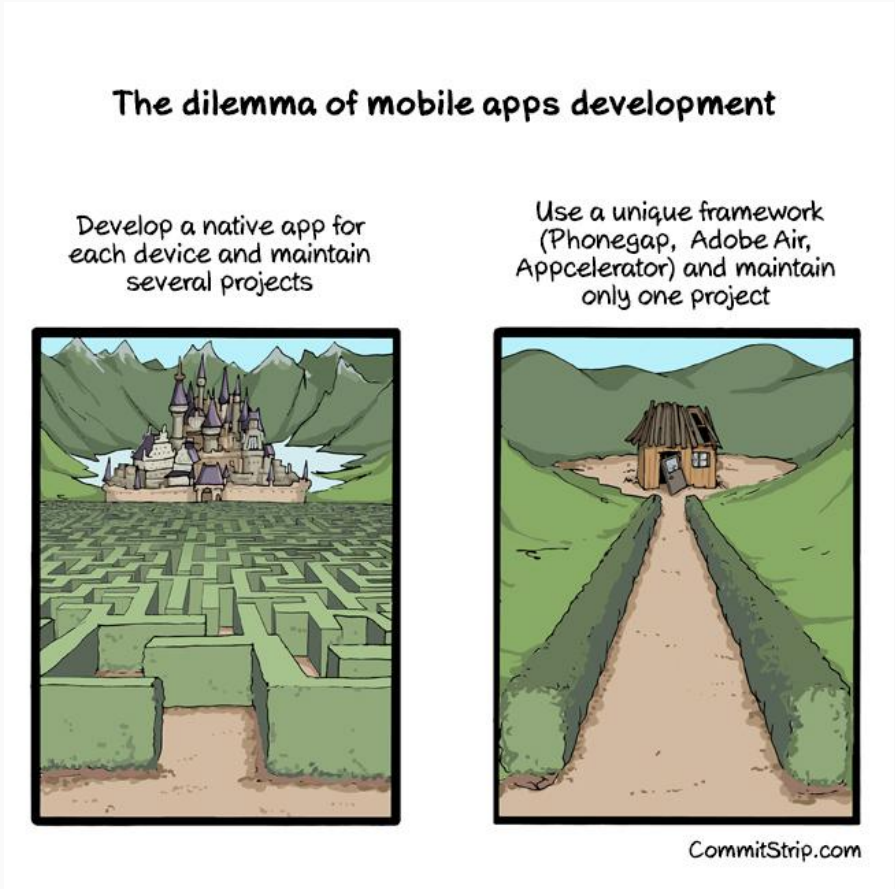
- Entwicklung ähnlich der Entwicklung von Hybrid Apps
  - Eine gemeinsame Codebasis mit Generierung des Codes für die jeweilige Plattform
- Prominenter Vertreter:
  - Xamarin (Microsoft)
    - Entwicklung in C#, Verwendung von .Net
    - Ermöglicht die separate Entwicklung des platform-spezifischen UIs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

Überblick App-Entwicklung

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

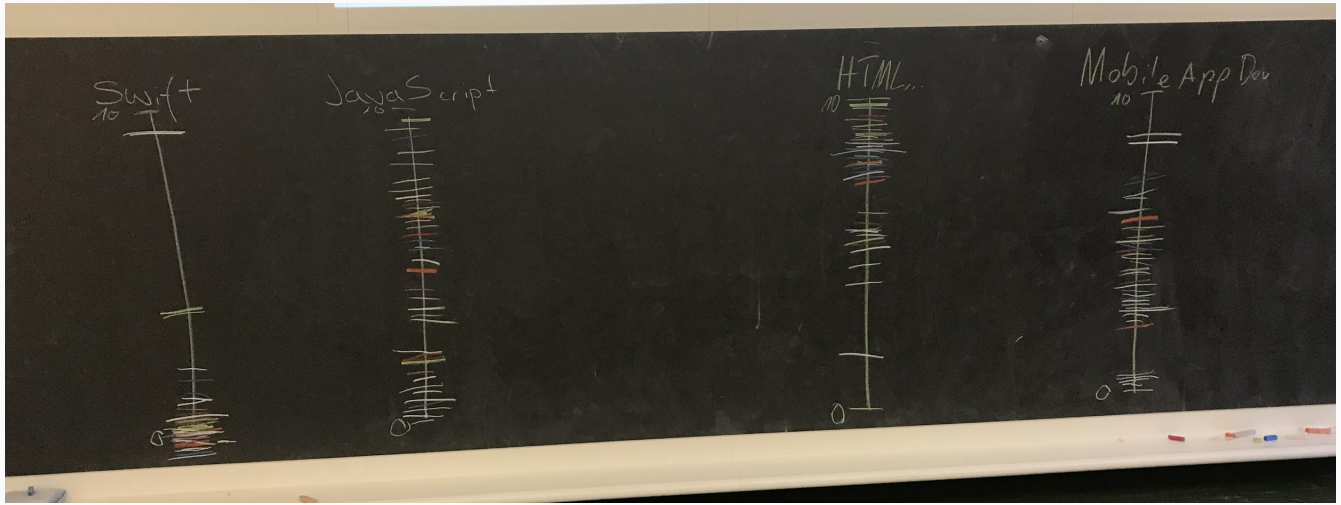
Überblick App-Entwicklung



Selbsteinschätzung

Wie gut sind meine Kenntnisse in

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



Erwartungen

Was erwartest du von diesem Workshop?

https://goo.gl/Q3AAnS

Sportturnier?

- Sammlung alternativer Themen
- Alternative Vorschläge bitte an roman.rast@fhnw.ch
- Einsendeschluss 21. Februar
- Nächste Mal Voting per google Forms

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