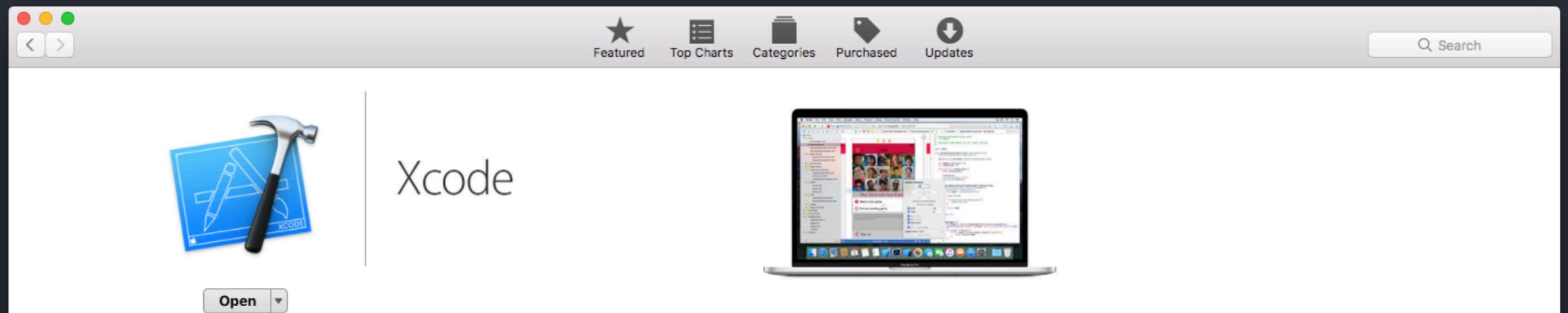


# Full Stack iOS Entwicklung mit Swift

WPF im MIM - WS 17/18  
Alexander Dobrynin, M.Sc.

# In der Zwischenzeit...



## Xcode

Essentials

Xcode includes everything developers need to create great applications for Mac, iPhone, iPad, Apple TV, and Apple Watch. Xcode provides developers a unified workflow for user interface design, coding, testing, and debugging. The Xcode IDE combined with the Swift programming language make developing apps easier and more fun than ever before.

...

...More

### What's New in Version 9.0.1

Xcode 9.0.1 includes Swift 4 and SDKs for iOS 11, watchOS 4, tvOS 11, and macOS High Sierra 10.13

...

...More

[Apple Web Site](#)

[Xcode Support](#)

[App License Agreement](#)

[Privacy Policy](#)

### Information

Category: Developer Tools

Updated: 16 October 2017

Version: 9.0.1

Price: Free

Size: 5.46 GB

Family Sharing: Yes

Language: English

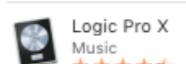
Developer: Apple Distribution International

© 1999–2017 Apple Inc.

Rated 4+

Compatibility:  
macOS 10.12.6 or later

### More Apps by Apple



Logic Pro X

Music

★★★★★



Final Cut Pro

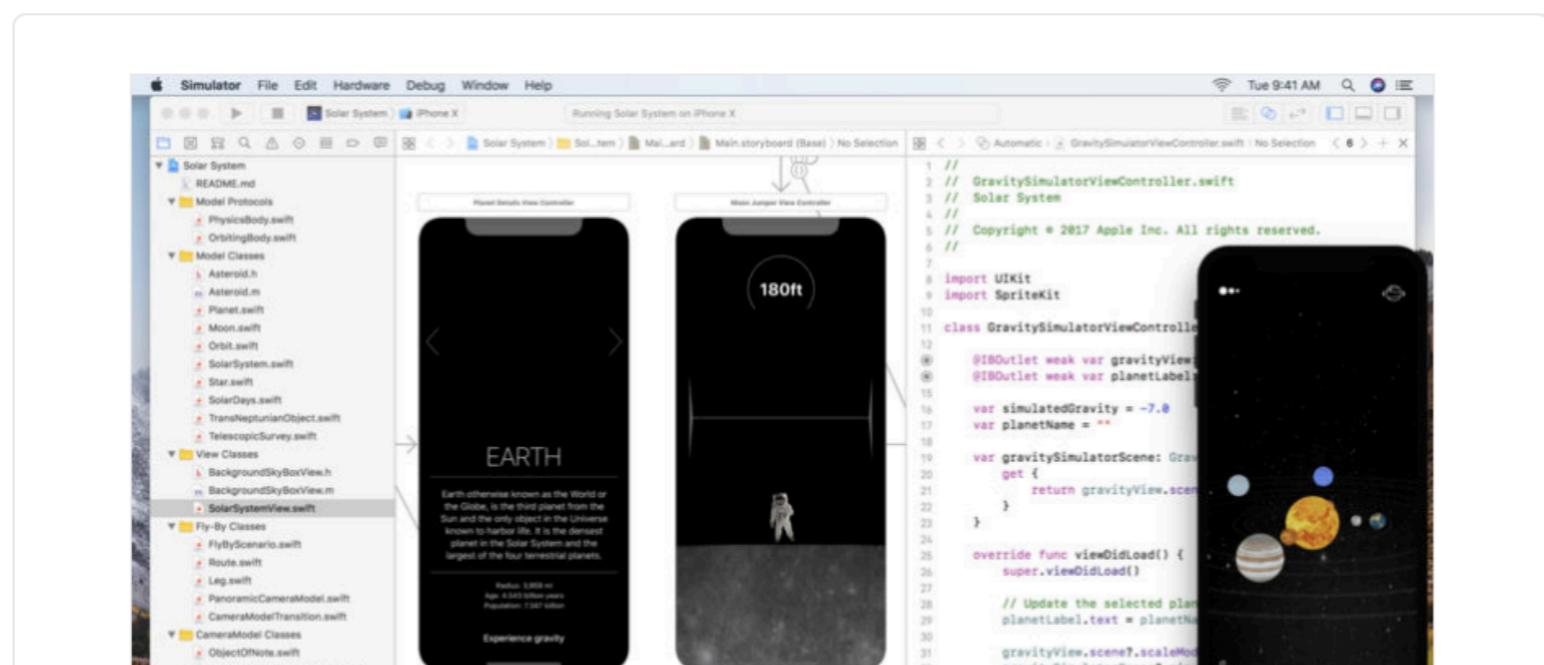
Video

★★★★★



MainStage 3

Music



# Heute

Über Mich / Euch  
Über die Veranstaltung  
Learning Outcomes

iOS Entwicklung  
Übersicht von iOS  
Swift  
Ressourcen

Demo

# Über Mich

- Alexander Dobrynin, M.Sc.
- Medieninformatik Bachelor & Informatik Master (SE)
- Wissenschaftlicher Mitarbeiter im ADV-Labor
- Freiberuflicher Softwareentwickler
- Java- und Scala-Enterprise-Apps (Backend)
- iOS- und Android-Apps (Front- und Backend)
- User-Centered-Agile-Software-Development (UCASD)
- Vollblut Nerd



@alexdobry  
[alexander.dobrynin@th-koeln.de](mailto:alexander.dobrynin@th-koeln.de)

# Über Euch

🙋‍♀️, wenn

- Medieninformatik BA?
- Entwickler?
- Designer?
- Mac Benutzer?
- iOS Benutzer?



<http://memeshappen.com/meme/picard-awesome/who-s-awesome-you-are-93491>

# Über die Veranstaltung

## Übersicht

- Wahlweise 6CP für MPO4 oder 5CP für MPO3 als WPF A oder B
- Max. 10 Personen
- Dienstag, 14:00 bis 16:00 Uhr in Raum 3.245 Neu
- Entwicklung von coolen iOS Apps mit Swift
  - vom Design bis zur Implementierung
  - kennenlernen einer neuen Programmiersprache und Systemumgebung
  - anwenden verschiedener Konzepte aus der Informatik
  - kollaboratives und agiles Arbeiten im Team
- Vorausgesetzt werden Kenntnisse in der objektorientierten Programmierung
- Besteht aus Vorlesung, Projekt und Abschlusspräsentation
- Benotet wird ein durchzuführendes Projekt und die Abschlusspräsentation
- Informationen und Organisatorisches im [Medieninformatik Wiki](#). Code, Folien und Gruppen auf [GitHub](#)

# Über die Veranstaltung

## Einordnung in den MIM

7

Welche fachlichen Ziele verfolgt das Studium?

Formale, algorithmische,  
mathematische Kompetenzen

Analyse-, Design-,  
Realisierungs- und  
Evaluierungskompetenzen

Theoretische und praktische  
Methodenkompetenzen

Gestalterische  
Medienkompetenzen

Fachliche, konstruktive  
Problemlösungs-  
kompetenzen

# Über die Veranstaltung

## Einordnung in den MIM

8

Welche “sozialen” Ziele verfolgt das Studium?

Projektmanagement- und  
Führungskompetenzen

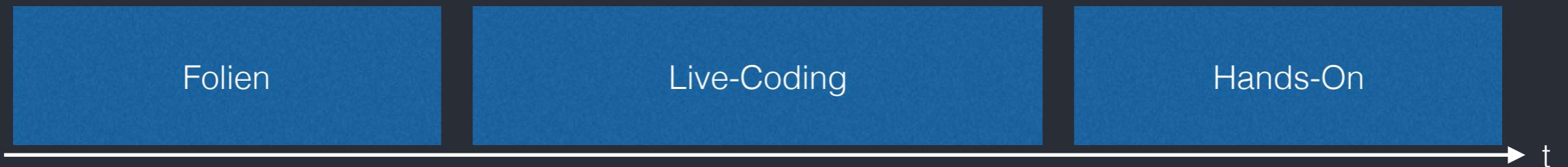
Selbstentwicklungs-  
kompetenzen

Kommunikationskompetenzen

Rezipierende und  
moderierende  
Medienkompetenzen

# Über die Veranstaltung

## Vorlesung



- Jede Vorlesung behandelt ein bestimmtes Thema
- Folien zum Thema mit anschließendem Live-Coding
  - Neu- und Weiterentwicklung unterschiedlicher Apps
  - Inhalte werden auf GitHub veröffentlicht
- Anschließende Bearbeitung von Assignments im Hands-On
  - Gelerntes unmittelbar konsolidieren
  - Snippets für das eigene Projekt fertigen
  - Ausprobieren, Refaktorieren, Syntax und Systemumgebung explorativ erkunden
  - Eigene Ideen entwickeln, Features Implementieren bzw. Optimieren
- Assignments können bis kurz vor der nächsten Vorlesung über Pull-Requests (GitHub) eingereicht werden, um eine positive Tendenz für die Note zu erhalten

# Über die Veranstaltung

## Vorlesung

- Mobile Entwicklung, iOS, Xcode und Tools
- Programmiersprache Swift, seine Konzepte und Patterns
- Objektorientierte Programmierung und angewandte funktionale Konzepte
- Verwenden von System-APIs und Third-Party-Bibliotheken
- Mobiler Network-Stack und mobile Storage Konzepte
- Multithreading und Event-Getriebene Denkweisen
- UI- und Interaction-Design, insbesondere im mobilen Kontext
- Verwenden eines Backend-Systems im mobilen Kontext
- ...

# Über die Veranstaltung

## Projekt

- Durchführung eines frei zu wählenden Projektes in max. 2er Teams
- Start der Bearbeitung wird im Laufe der Vorlesung “freigegeben”
- Erarbeiten einer neuen Programmiersprache und kennenlernen einer neuen Systemumgebung
- Design und Implementierung einer iOS App mit Swift, inklusive\*
  - der Implementierung und/oder Anbindung eines Backends,
  - der Verwendung einer Extension und/oder
  - der Verwendung eines ausgewählten Core-Modules (iOS System-API)

\*Die Optionen sind je nach Projekt zu wählen

- Projektmanagement, Organisation und Dokumentation auf GitHub
- Die Dokumentation beinhaltet u.a. eine Skizzierung und Erklärung der wesentlichen Komponenten der Architektur
- Zwischenaudits während des Projektes - aktueller Stand, Fragen und Feedback der Studierenden

# Über die Veranstaltung

## Abschlusspräsentation

- Anwesenheitspflichtig
- Termin wird gegen Ende der Vorlesungszeit abgestimmt
- Maximal 40 Minuten, bestehend aus
  - 20 Minuten Vortrag
  - 10 Minuten Demo
  - 10 Minuten Fragen und Diskussion
- Folien, bestehend aus
  - mindestens: Thema, Design, Architektur, Durchführung und Herausforderungen
  - des Weiteren: besondere Highlights, alles was fehlt und alles was begeistert
- Entscheidungen begründen und sinnvoll abwägen
- Live-Demo der App (eigenes Gerät oder Simulator)

# Über die Veranstaltung

## Bewertungsgrundlage

- Learning-Outcomes
- Beherrschen und Dekonstruieren von komplexen Problemen in lösbare Einheiten
- Qualität des Codes
- Einhalten von iOS und Swift Idiomen
- Verständnis und korrekte Verwendung von iOS
  - MVC, ViewController Lifecycle, Belastung des Main-UI-Threads, Berücksichtigung limitierter Ressourcen, etc.
- Cocoa-Touch und Frameworks sinnvoll verwendet
- Angeeignetes Wissen, welches nicht in der Veranstaltung thematisiert wurde und sonstige Transferleistungen
- Gebrauchstaugliches und adäquates mobiles Design
- Alles in der Abschlusspräsentation, vor allem der rote Faden und die Qualität der Begründungen
- Alles, was ansonsten begeistert
- Eure persönliche Entwicklung

# Learning Outcomes

- Probleme Analysieren und Dekonstruieren, Lösungen Konzipieren, Fertigen und Bewerten
- Bestehende Konzepte der Informatik konsolidieren
- Neue Programmiersprache, Konzepte und Sprachfeatures
- Mobile (iOS-lastige) Patterns und Programmierstile
- Funktionale Denkweise
- Mobile Entwicklung im Allgemeinen
- iOS Entwicklung im Speziellen
- Zusammenspiel von Design und Implementierung
- Zusammenspiel von Mobile- und Backend App
- Kollaborative Arbeiten mit Git und GitHub
- Agile Softwareentwicklung in einem Team

# Fragen soweit?

Über Mich / Euch  
Über die Veranstaltung  
Learning Outcomes

iOS Entwicklung  
Übersicht von iOS  
Swift  
Ressourcen

Demo

# iOS Entwicklung

*“Stellt euch an dieser Stelle eine beliebige Statistik vor, die zeigt, wie großartig mobile, insbesondere iOS Entwicklung ist.”*

# iOS Entwicklung

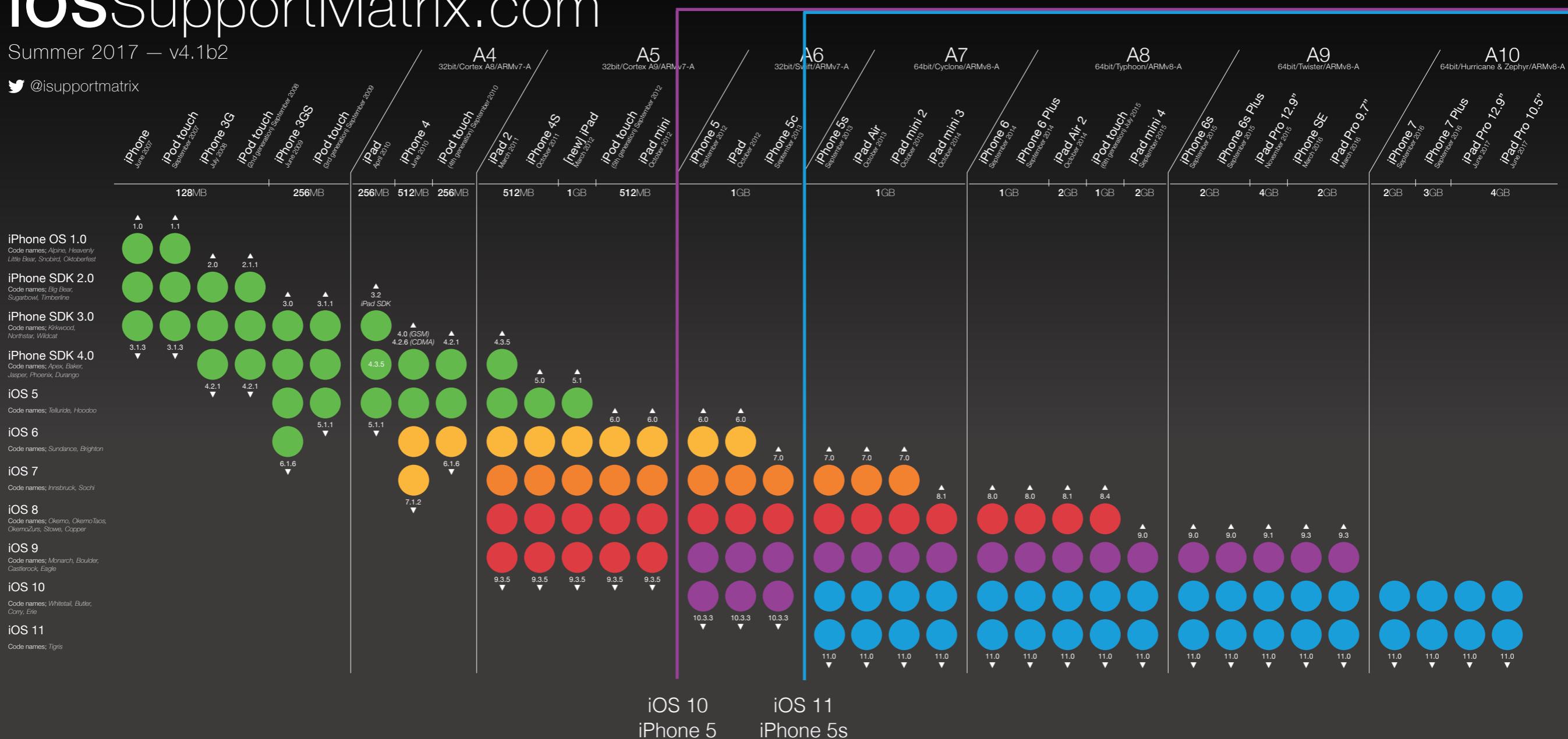
- Verschiedene Geräte, Gerätegrößen und OS-Versionen
- Verschiedene Plattformen (iOS, tvOS, watchOS, macOS)
- Weitreichendes Ökosystem
- Limitierte Ressourcen
- Verschiedene Sensoren, Kamera, mobiles Internet, usw.
- Mobiler Computer, mobile Lösungen
- Mit einfachen UIs komplexe Probleme effizient lösen
- Veröffentlichen von Apps über den App Store

# iOS Entwicklung

iOSSupportMatrix.com

Summer 2017 – v4.1b2

[@isupportmatrix](#)

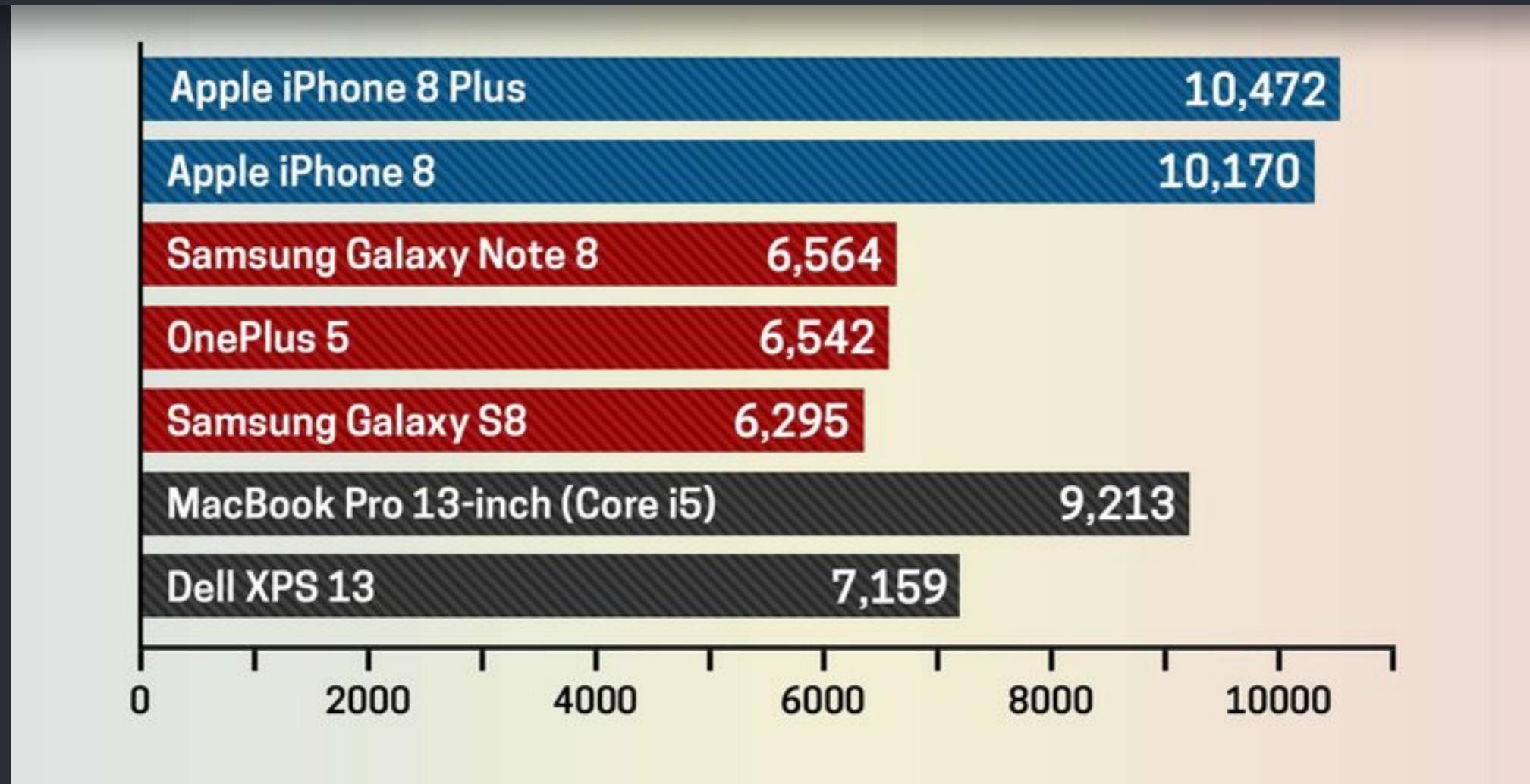


# iOS Entwicklung

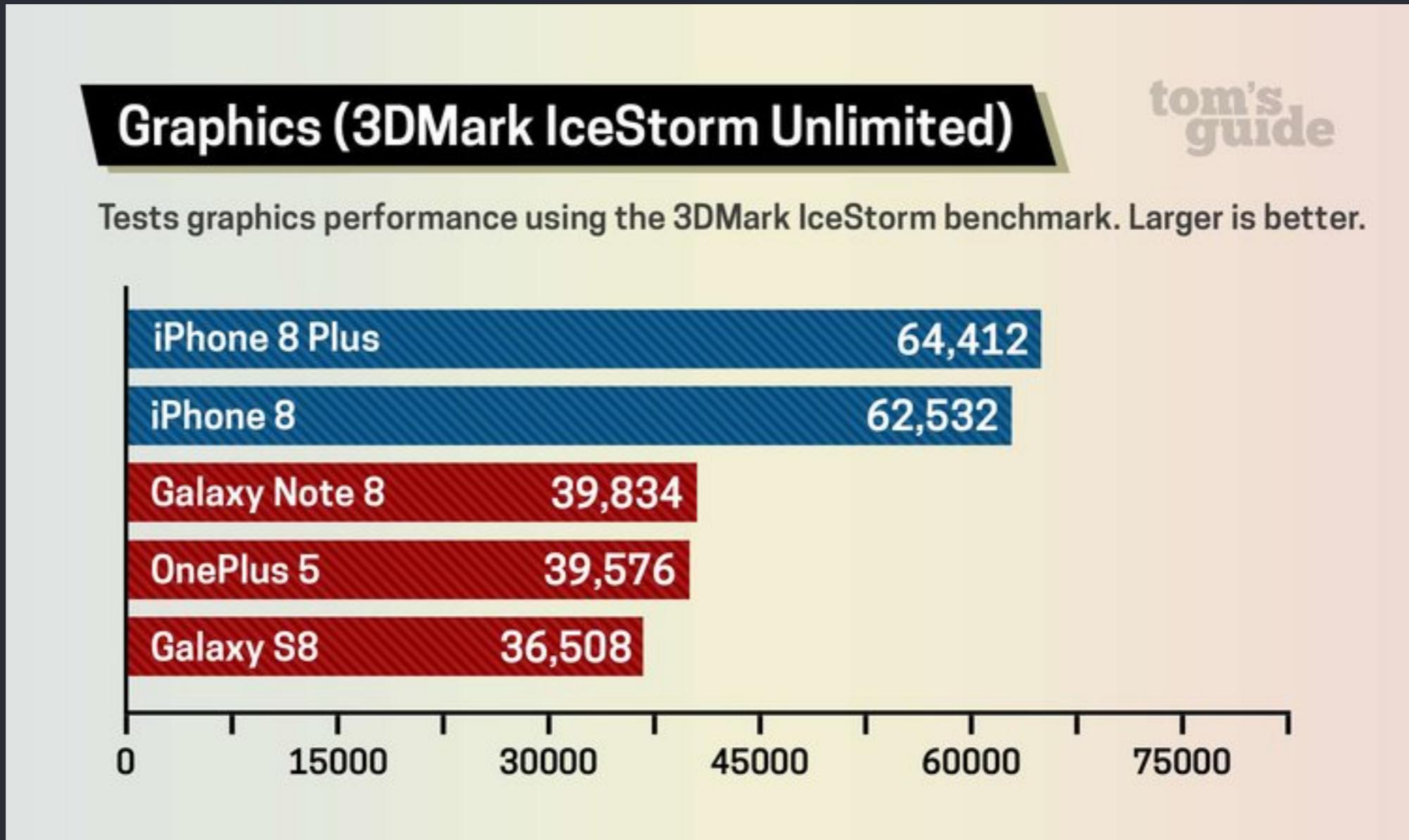
- Event-Getriebene Eingaben auf dem Main-Thread
- Zentraler Einstiegspunkt und definierter Lebenszyklus jeder App
- Zustände, Transitionen und Callbacks
- Synchrone- und Asynchrone Kommunikation (Poll und Push)
- Integration und Interprozesskommunikation mit anderen Apps und Systemen
- Reaktive, funktionale und objektorientierte Programmiermodelle
- Unterstützung für Augmented Reality (ARKit) und Machine Learning (Core ML)
- Interessante Kamera Möglichkeiten wie Infrarot, Tiefen-Informationen und Porträt-Modus mit iPhone 8 Plus und iPhone X

# iOS Entwicklung

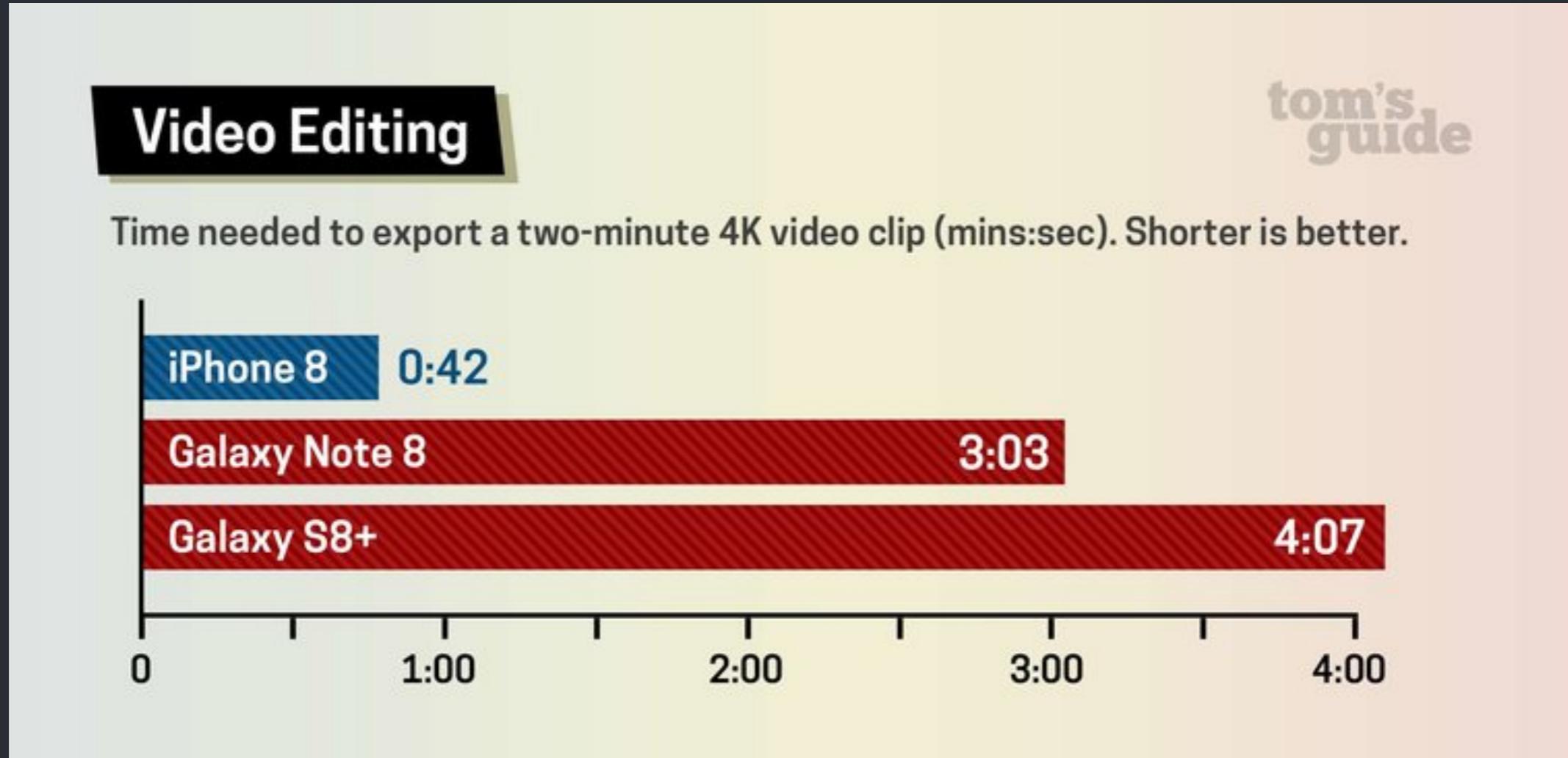
*“the short is answer is yes that the scores are comparable across platforms, so if an iPhone 8 scores higher than an i5, then the iPhone 8 is faster than the i5” - John Poole, Founder of Geekbench*



# iOS Entwicklung

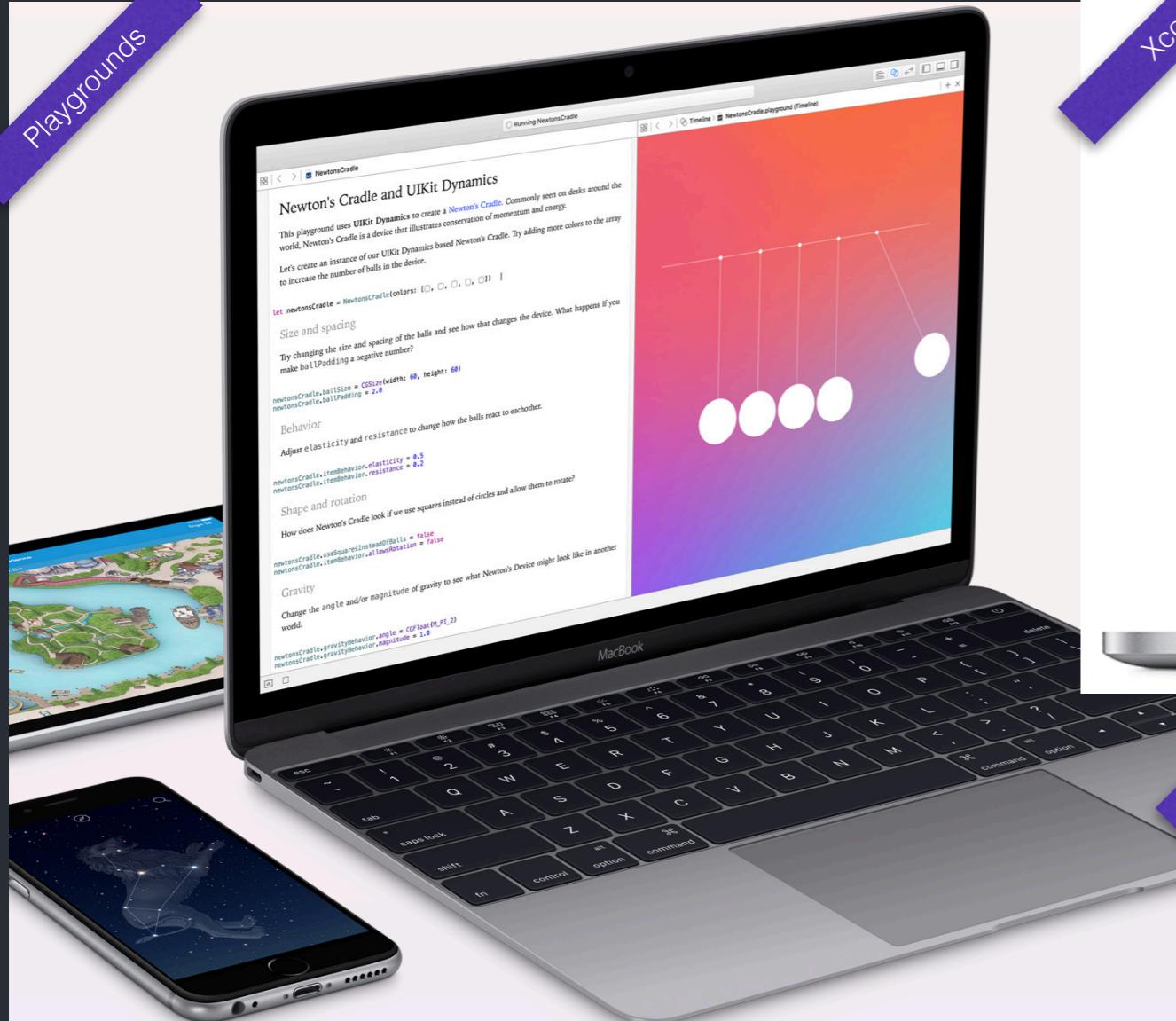


# iOS Entwicklung

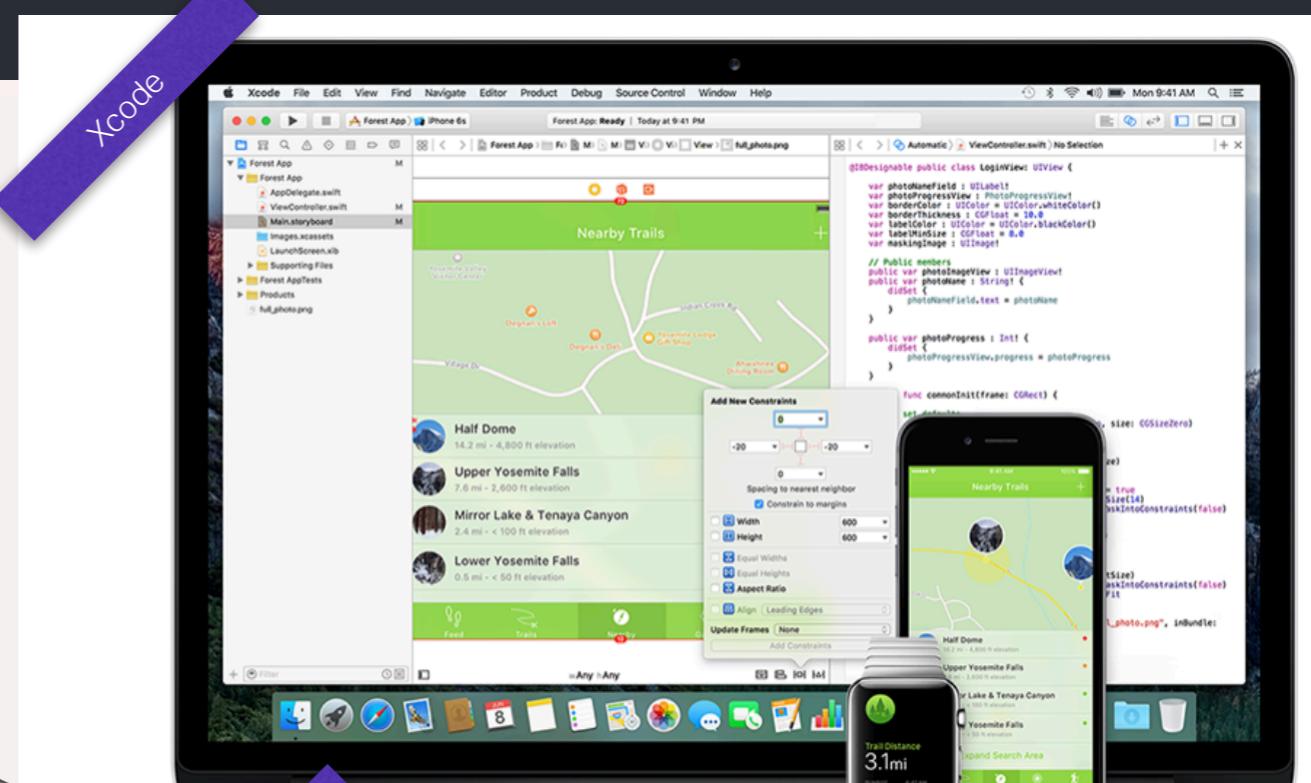


# iOS Entwicklung

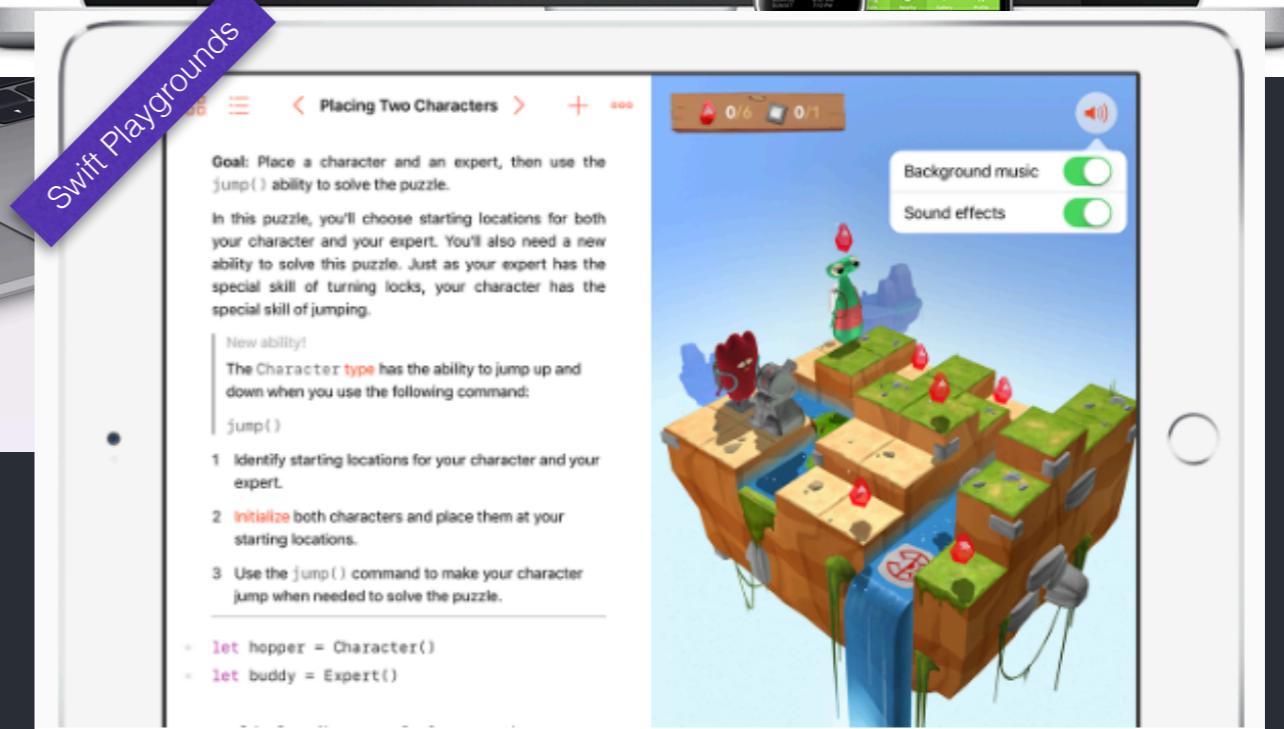
Playgrounds



Xcode

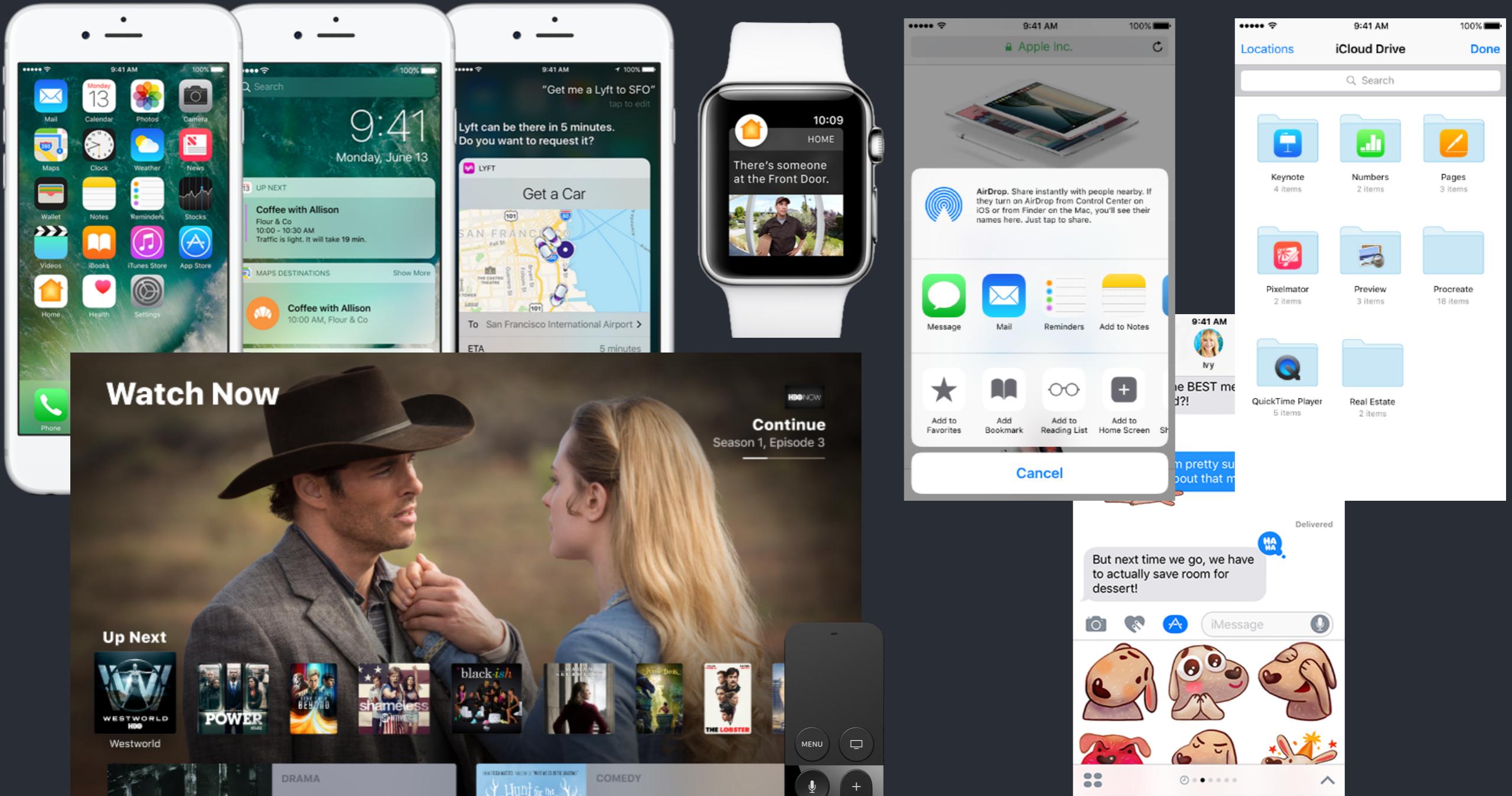


Swift Playgrounds



<https://developer.apple.com/programs/>  
<https://www.apple.com/everyone-can-code/>  
<https://developer.apple.com/swift/>

# iOS Entwicklung



<https://developer.apple.com/ios/human-interface-guidelines/extensions/messaging/>  
<https://developer.apple.com/ios/human-interface-guidelines/overview/design-principles/>  
<https://developer.apple.com/watchos/human-interface-guidelines/overview/>  
<https://developer.apple.com/tvos/human-interface-guidelines/overview/>

# Full Stack iOS Entwicklung

*“A full-stack developer is someone who can handle a vertical slice of all the technologies used to deliver a feature from the ~~backend to the frontend.~~”*

*abstract and elusive vision of the user ...*

*a concrete and complex solution.*

1. Server, Network, and Hosting Environment

2. Data Modeling

+ Agile Software Development

3. Business Logic

4. API layer / Action Layer / MVC

5. User Interface

+ User-Centered Design

6. User Experience

+ Interaction Design

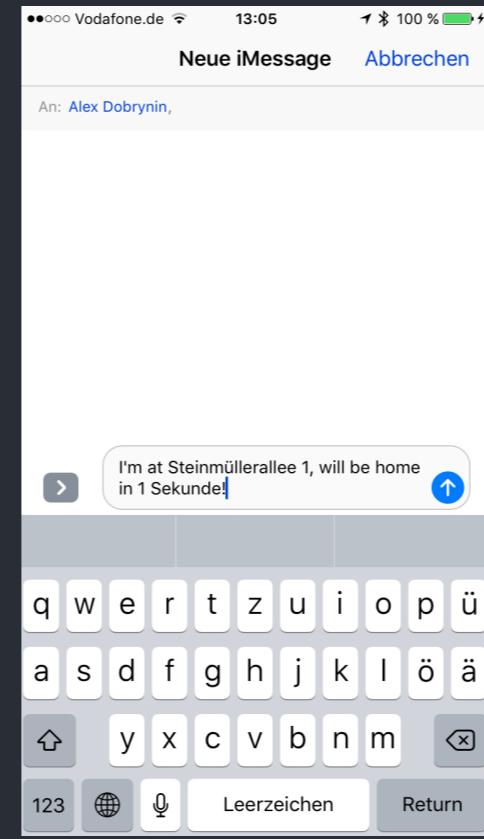
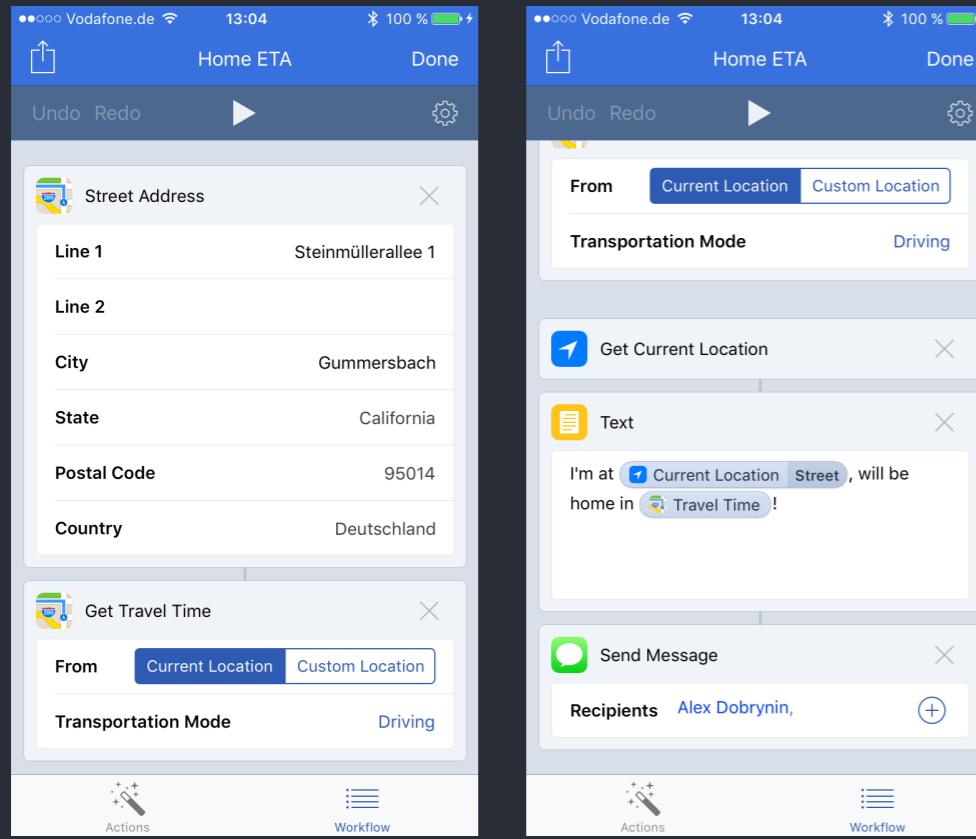
7. Understanding what the customer and the business need

[https://www.reddit.com/r/iOSProgramming/comments/3e7m9q/what\\_does\\_it\\_mean\\_to\\_be\\_a\\_full\\_stack\\_ios\\_developer/](https://www.reddit.com/r/iOSProgramming/comments/3e7m9q/what_does_it_mean_to_be_a_full_stack_ios_developer/)  
<https://www.laurencegellert.com/2012/08/what-is-a-full-stack-developer/>

=> User-Centered-Agile-Software-Development

# Full Stack iOS Entwicklung

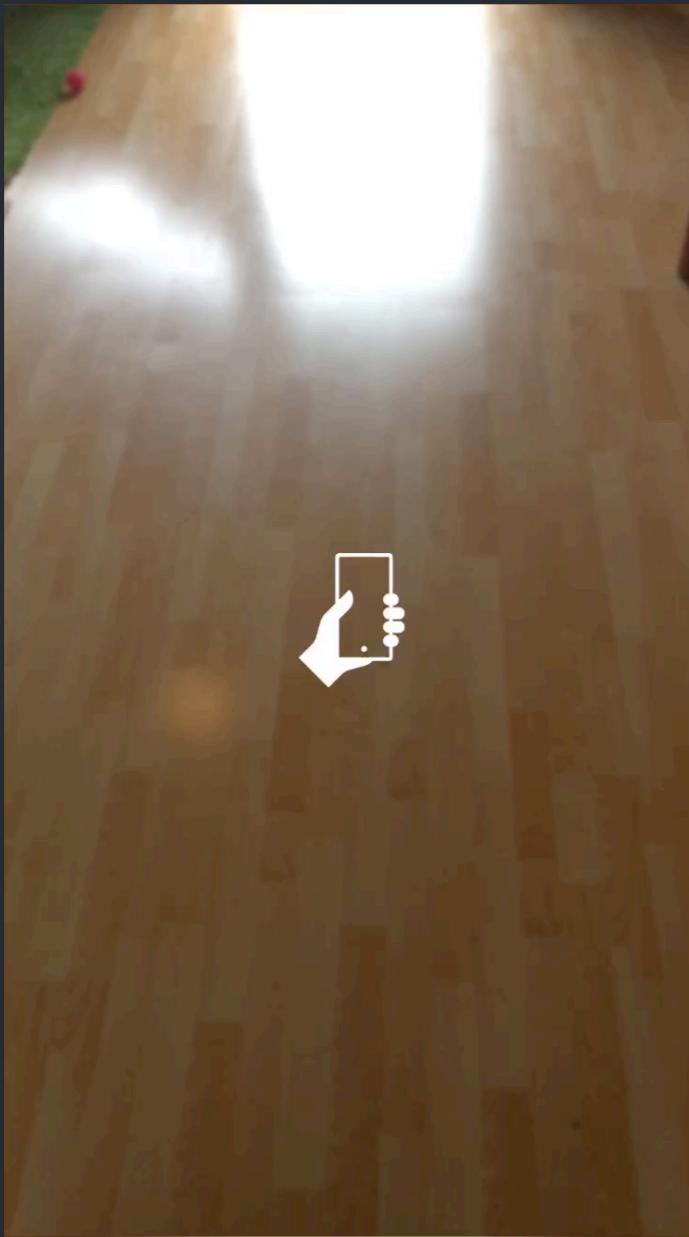
- Konzept + App + Backend = mobile Development
- App trägt zum Erfolg des Geschäfts bei
- Steuerung des Geschäftsmodells
- "Build Features, Not Apps"



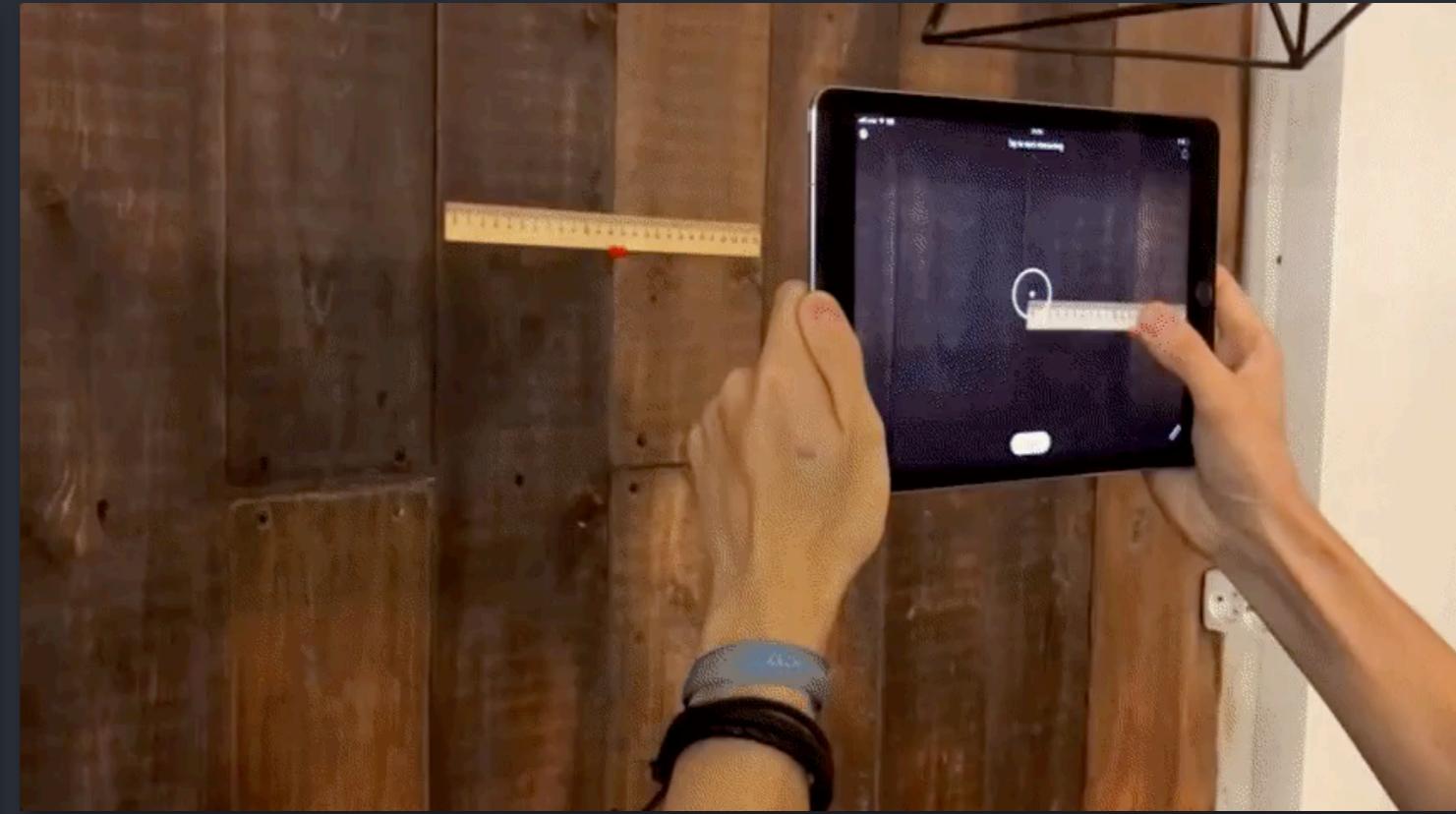
[https://commons.wikimedia.org/wiki/File:Car2go\\_logo.svg](https://commons.wikimedia.org/wiki/File:Car2go_logo.svg)  
[https://en.wikipedia.org/wiki/File:Tinder\\_logo.png](https://en.wikipedia.org/wiki/File:Tinder_logo.png)  
<https://outbankapp.com>  
<https://box.fileee.com/en/>  
[https://de.wikipedia.org/wiki/Datei:Instagram\\_logo\\_2016.svg](https://de.wikipedia.org/wiki/Datei:Instagram_logo_2016.svg)  
<https://workflow.is/static/landing/img/help-icon.png>

# Full Stack iOS Entwicklung

Ikea Place



AR MeasureKit



[https://cdn.vox-cdn.com/thumbor/QWH02Iys-jSFh7kTo1\\_AF8kH2yY=/0x0:862x484/1520x0/filters:focal\(0x0:862x484\):gifv\(\):no\\_upscale\(\)/cdn.vox-cdn.com/uploads/chorus\\_asset/file/9274153/measure.gif](https://cdn.vox-cdn.com/thumbor/QWH02Iys-jSFh7kTo1_AF8kH2yY=/0x0:862x484/1520x0/filters:focal(0x0:862x484):gifv():no_upscale()/cdn.vox-cdn.com/uploads/chorus_asset/file/9274153/measure.gif)

# Übersicht von iOS

Cocoa Touch

UIKit, MapKit, App Extensions, ...

Media

Core Graphics, Core Audio, AVFoundation, ...

Core Services

SQLite, Foundation, Core Location, ...

Core OS

Kernel, Security, Filesystem, ...

# Übersicht von iOS

- 100% objektorientiert (zumindest der Kern von Cocoa-Touch)
- Objective-C und Swift als Programmiersprachen
- Xcode als IDE
- Swift Playgrounds als “interaktive REPL”
- Profiling mit Instruments
- Speicherverwaltung mit Automatic-Reference-Counting (ARC)
- Code-Design mit Model-View-Controller (MVC)
- UI-Design mit Storyboards (in Xcode)
- Apple Developer Program

# Swift

- Vorgestellt auf der WWDC am 02.06.2014
- Statisch typisierte, multiparadigmen Programmiersprache
- Mittlerweile in der Version 4 und Open-Source
- “Objective-C without the C”
- Auf lange Sicht als primäre Programmiersprache für iOS und MacOS Anwendungen

*“Swift combines the best in **modern language thinking** with wisdom from the wider Apple engineering culture. The compiler is **optimized for performance**, and the language is **optimized for development**, without compromising on either. It’s **designed to scale** from “hello, world” to an entire operating system. All this makes Swift a sound future **investment for developers and for Apple.**”*

# Swift

- Structs, Klassen und Enums
- Methoden, Funktionen und Properties
- `func` `dismissKeyboard() { ... }`, `func` `remove(element: Element) -> Element? {...}`
- Basistypen wie String, Int, Double, Float, ... und Collections wie Array, Dictionary, ...
- Deklaration von Properties über `var` (mutable) oder `let` (immutable)
- Inference Engine leitet den Typen aus der rechten Seite (rhs) ab, sondern vorhanden
- `let course = "FSIOS"`
- Ansonsten werden Typen mit einem Doppelpunkt nach dem Variablenamen angegeben
- `var applicants: [String] = []`
- Optionals als expliziter Typ für “nullable Values”
- `var passed: Bool?; passed = true`

# Ressourcen

- [swift.org](http://swift.org)
- [github.com/apple/swift](https://github.com/apple/swift)
- [github.com/apple/swift-evolution/blob/master/proposals](https://github.com/apple/swift-evolution/blob/master/proposals)
- [developer.apple.com/swift/programming\\_language](https://developer.apple.com/swift/programming_language)
- [itunes.apple.com/de/book/the-swift-programming-language-swift-3-1](https://itunes.apple.com/de/book/the-swift-programming-language-swift-3-1)
- [developer.apple.com/swift/getting\\_started](https://developer.apple.com/swift/getting_started)
- [developer.apple.com/sample\\_code](https://developer.apple.com/sample_code)
- [developer.apple.com/ios/human-interface-guidelines](https://developer.apple.com/ios/human-interface-guidelines)
- [Developing iOS 10 Apps with Swift](https://Developing iOS 10 Apps with Swift)
- [CodePath iOS Guides](https://CodePath iOS Guides)



# Demo - Yatodoa

Xcode, Projekt erstellen, App im Simulator ausführen

UI-Design im Storyboard

Klassen, Variablen, Methoden und fundamentale Datentypen in Swift

Verbindungen zwischen Storyboard (UI) und ViewController (Code)

UITextField, UIButton, UILabel und Target-Action

# Yatodoa - Assignment

- Zweiten ToDo-Eintrag hinzufügen
- Assoziation zwischen UIButton und UITextField modellieren
- Beide Einträge sollen autonom funktionierten
- Erledigte Aufgaben als lesbaren Text in einem UILabel modellieren
- Optional: Aufgaben durchstreichen wenn erledigt
- Sonstige Änderungen und Verbesserungen sind willkommen
- Per Pull-Request einreichen

