

# Full Stack iOS Entwicklung mit Swift

WPF im MIM - SS 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

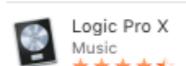
Developer: Apple Distribution International

© 1999–2017 Apple Inc.

Rated 4+

Compatibility:  
macOS 10.12.6 or later

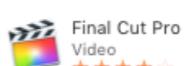
### More Apps by Apple



Logic Pro X

Music

★★★★★



Final Cut Pro

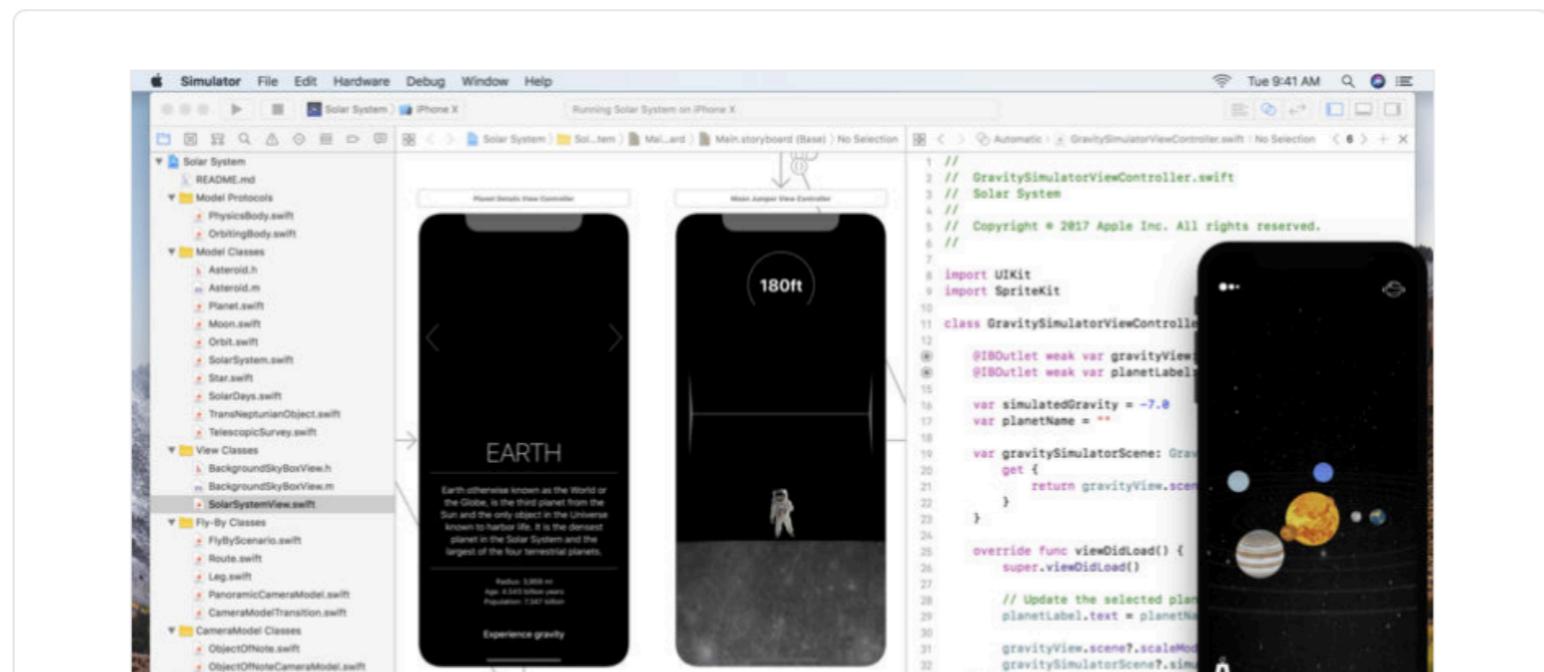
Video

★★★★★



MainStage 3

Music



# Heute

Über Mich / Euch  
Über die Veranstaltung

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

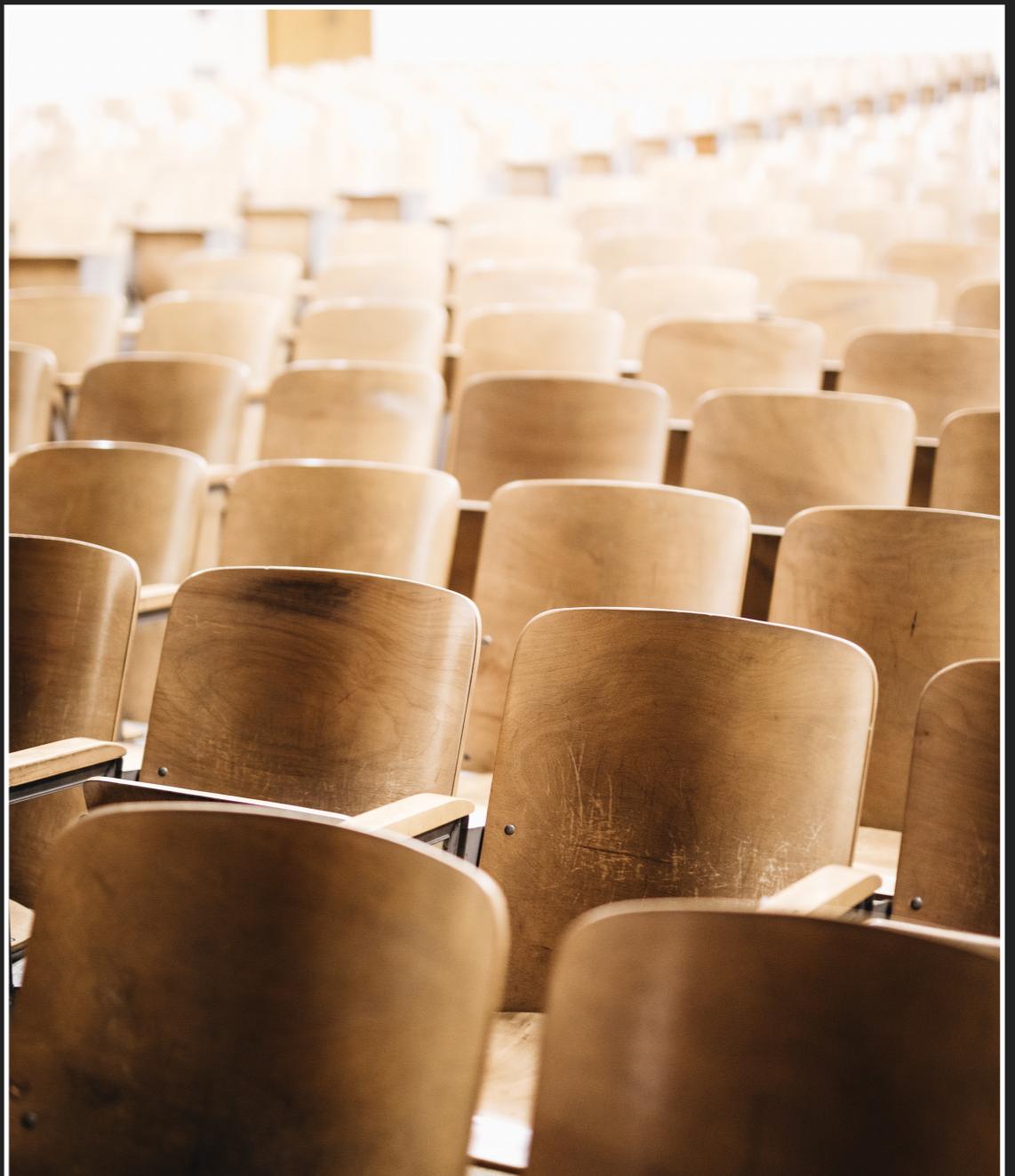


@alexdobry  
[alexander.dobrynin@th-koeln.de](mailto:alexander.dobrynin@th-koeln.de)

# Über Euch

🙋‍♀️🙋‍♂️, wenn

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



<https://unsplash.com/photos/xPHmmVKS8IM>

# Über die Veranstaltung

## Übersicht

- Wahlweise 6CP für MPO4 oder 5CP für MPO3 als WPF A oder B
- Max. 10 Personen
- Mittwoch, 13:00 bis 15:00 Uhr in Raum 3.245?
- 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 zur Veranstaltung 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

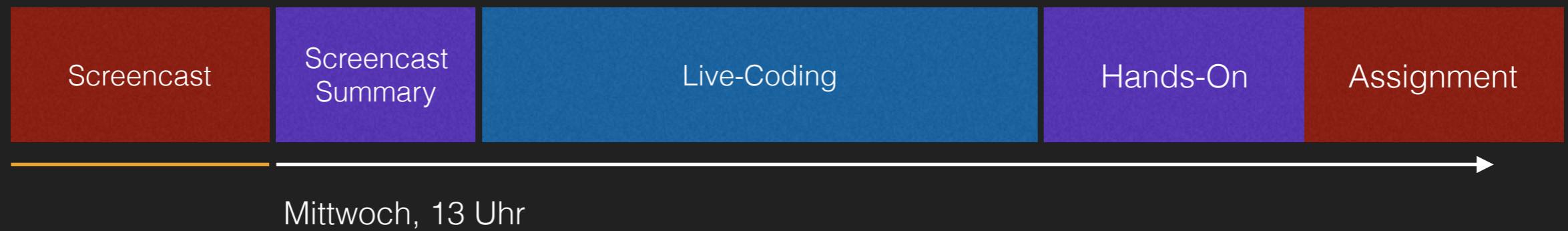
## Vorlesung

---

Mittwoch, 13 Uhr

# Über die Veranstaltung

## Vorlesung



# Über die Veranstaltung

## Vorlesung

- Jeder Termin behandelt ein bestimmtes Thema
- Zu jedem Termin wird mindestens ein **Screencast** angeboten
  - **Folien** mit Notizen als Einführung und Nachschlagewerk
  - Im **Selbststudium** vor der jeweiligen Vorlesung konsumieren
- Im Termin werden die Inhalte aus dem Screencast im **Live-Coding** konsolidiert
  - Praktische Anwendung anhand Neu- und Weiterentwicklungen unterschiedlicher Apps
  - Setzt das Wissen aus den Screencasts voraus
  - In den ersten 15 Minuten werden die Inhalte des Screencasts zusammengefasst, Fragen beantwortet und Unklarheiten beseitigt
- Anschließend werden **Assignments** im **Hands-On** bearbeitet
  - Kleine Aufgaben, Bug Fixes oder Optimierungen aus dem Live-Coding
  - Gelerntes unmittelbar konsolidieren, Snippets für das eigene Projekt fertigen, eigene Ideen entwickeln
  - Ausprobieren, Refaktorieren, Syntax und Systemumgebung explorativ erkunden
- Assignments werden in einer dedizierten Branch bearbeitet und 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

- Design und Implementierung einer iOS App mit Swift
- Mobile Entwicklung, iOS, Xcode, Playgrounds 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 Storage Konzepte
- Multithreading und Event-Getriebene Denkweisen
- UI- und Interaction-Design, insbesondere im mobilen Kontext
- Verwenden eines Backend-Systems im mobilen Kontext
- Anwenden verschiedener Konzepte aus der Informatik
- Kollaboratives und agiles Arbeiten im Team
- Arbeiten, Dokumentieren und Organisieren mithilfe von Versionsverwaltungs-Tools

# 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

- Erfüllungsgrad der Learning-Outcomes
- Beteiligung an den Assignments
- Beherrschen und Dekonstruieren von komplexen Problemen in lösbare Einheiten
- Qualität des Codes
- Einhalten von iOS und Swift Idiomen
- Verständnis und korrekte Verwendung von iOS, Cocoa-Touch und Frameworks
- 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, die Qualität der Begründungen und eure persönlichen Gedanken
- Alles, was ansonsten begeistert
- Eure persönliche Entwicklung

# Über die Veranstaltung

## 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
- (Kollaboratives) Arbeiten mit Git und GitHub
- Agile Softwareentwicklung (in einem Team)

# Fragen soweit?

Über Mich / Euch  
Über die Veranstaltung

iOS Entwicklung  
Übersicht von iOS  
Swift  
Ressourcen

Demo

# iOS Entwicklung

DOWNLOADS EXCEEDED

**175**  
BILLION  
**+60%**

GROWTH FROM 2015

CONSUMER SPEND EXCEEDED

**\$86**  
BILLION  
**+105%**

GROWTH FROM 2015

EACH USER SPENT NEARLY

**1.5**  
MONTHS  
IN APPS PER YEAR  
**+30%**

GROWTH FROM 2015

# iOS Entwicklung

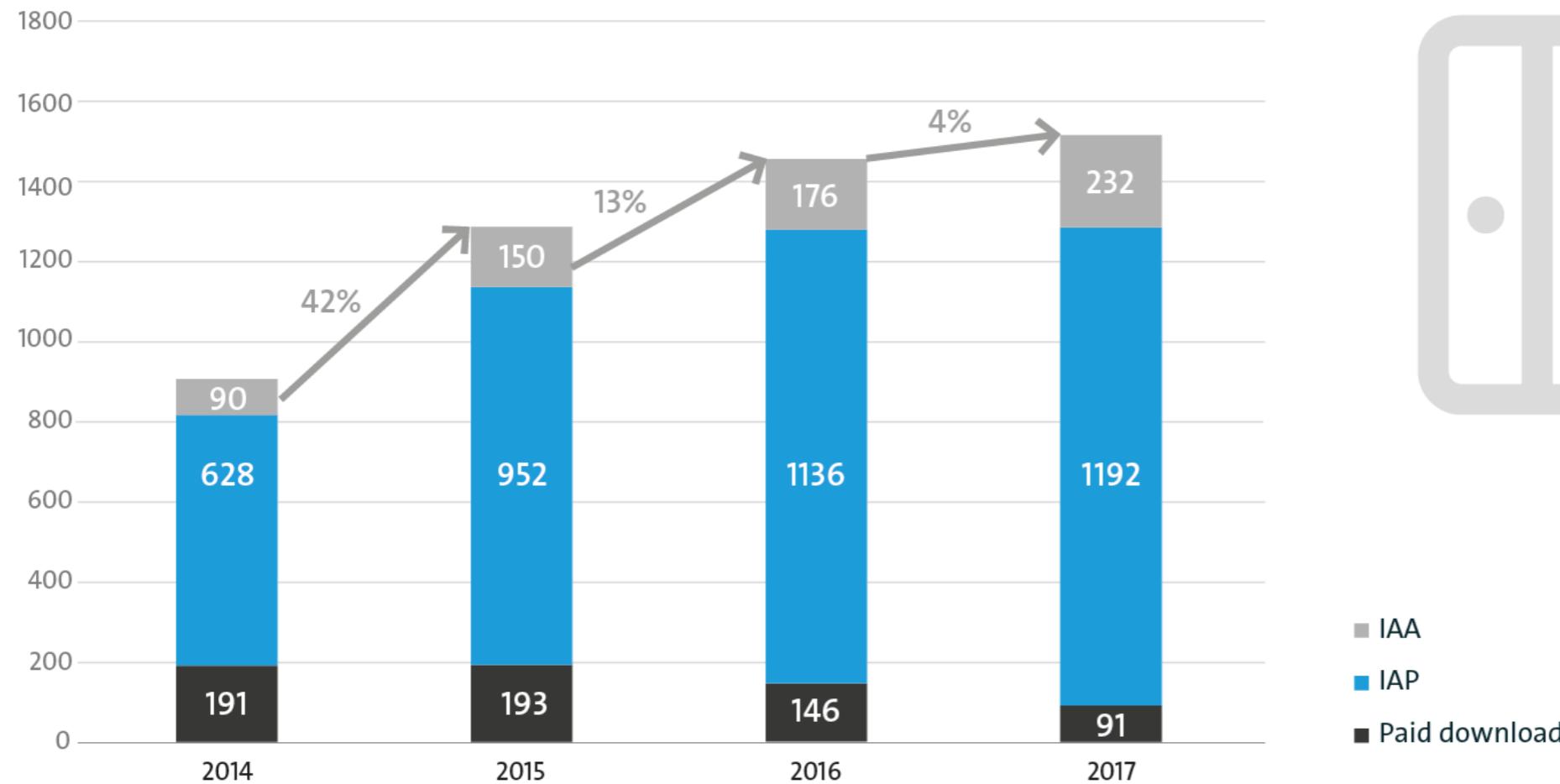
## Deutscher App-Markt mit 1,5 Mrd. Euro Umsatz

Gesamt (Mio.): 909

1.287

1.456

1.509



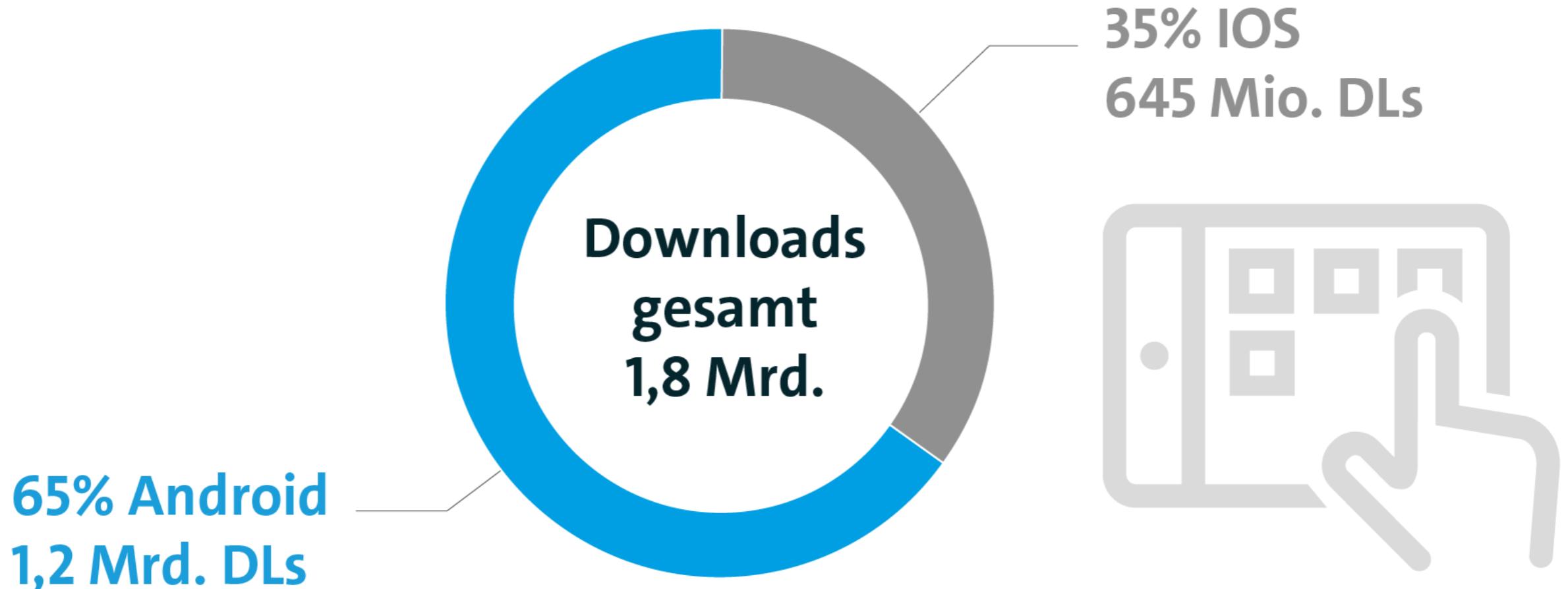
- IAA
- IAP
- Paid download

Quelle: research2guidance

**bitkom**

# iOS Entwicklung

**App-Downloads steigen auf 1,8 Milliarden**



Quelle: research2guidance

**bitkom**

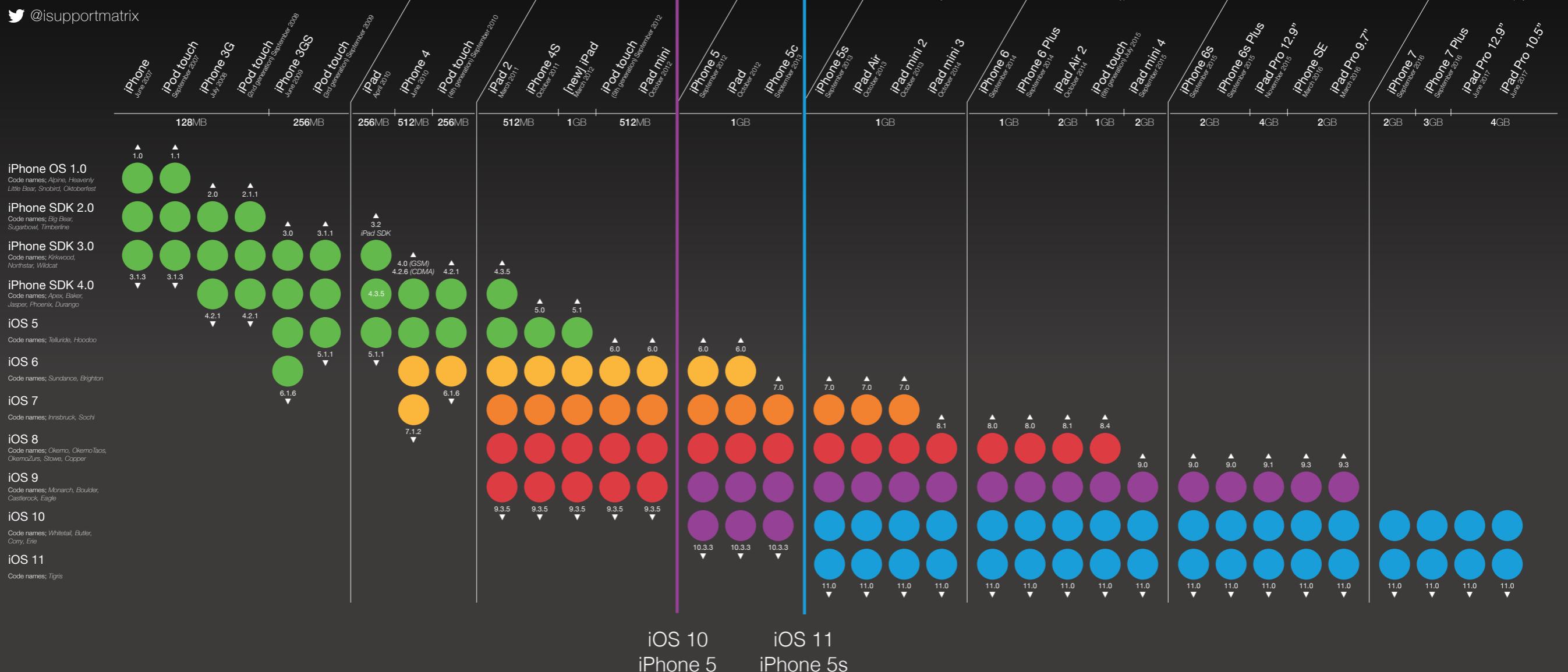
# iOS Entwicklung

- 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

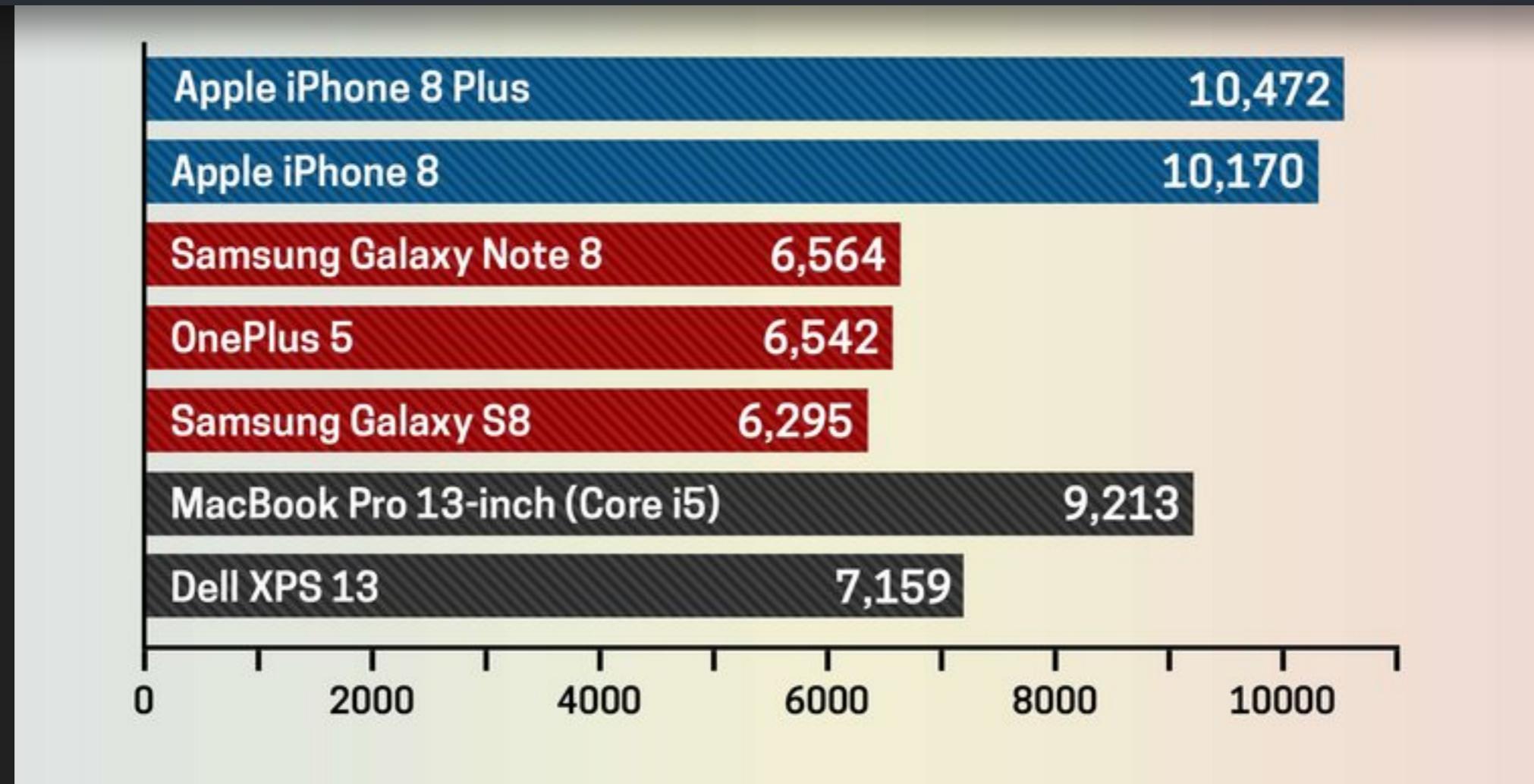


# iOS Entwicklung

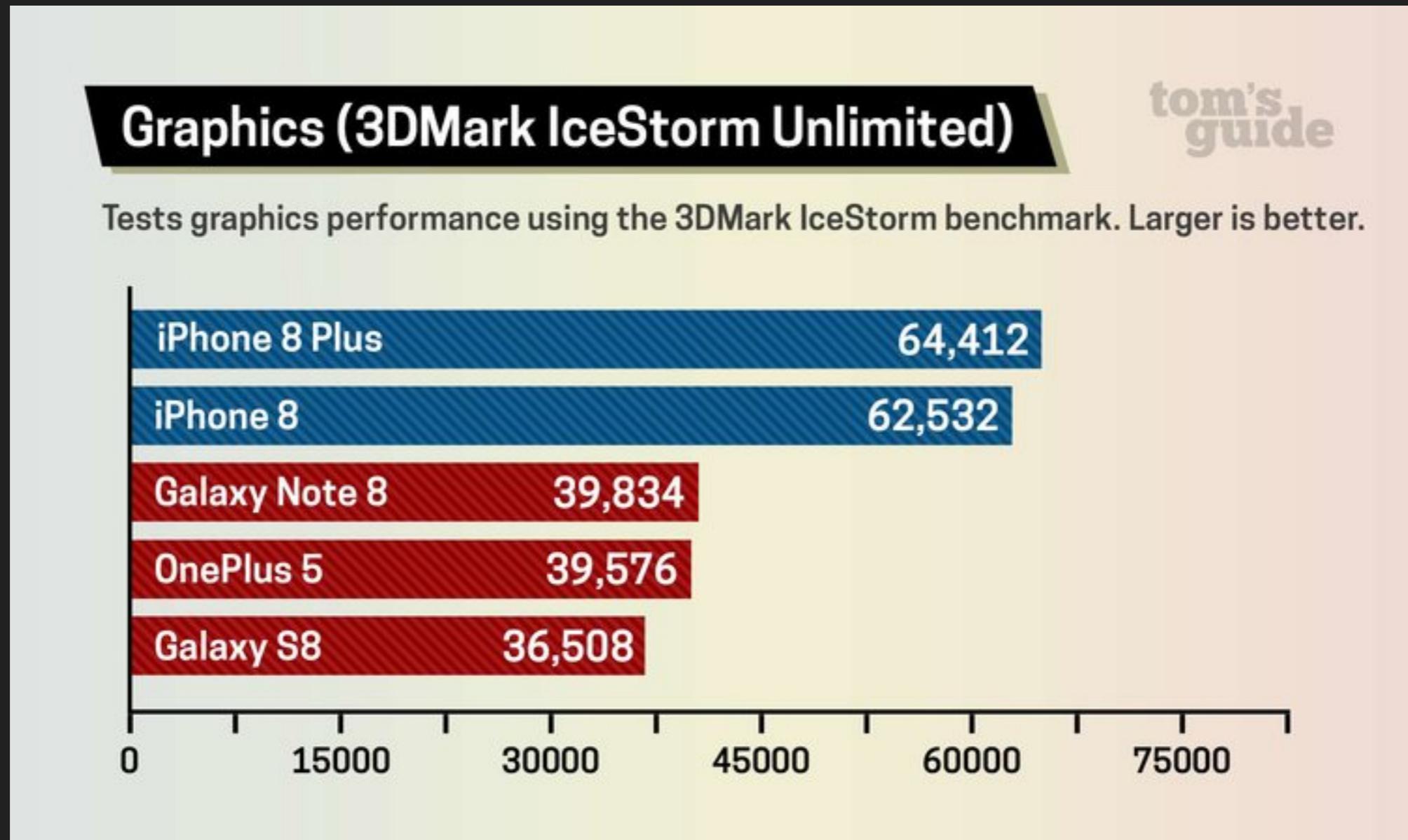
- Event-Getriebene Eingaben auf dem Main-Thread
- Zentraler Einstiegspunkt und definierter Lebenszyklus jeder App
- Zustände, Transitionen und Callbacks
- Synchrone- und Asynchrone Kommunikation (Poll und Push)
- Integration und Interprozesskommunikation mit anderen Apps und Systemen
- Reaktive, funktionale 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
- Absurd leistungsstarker Computer in der Hosentasche

# iOS Entwicklung

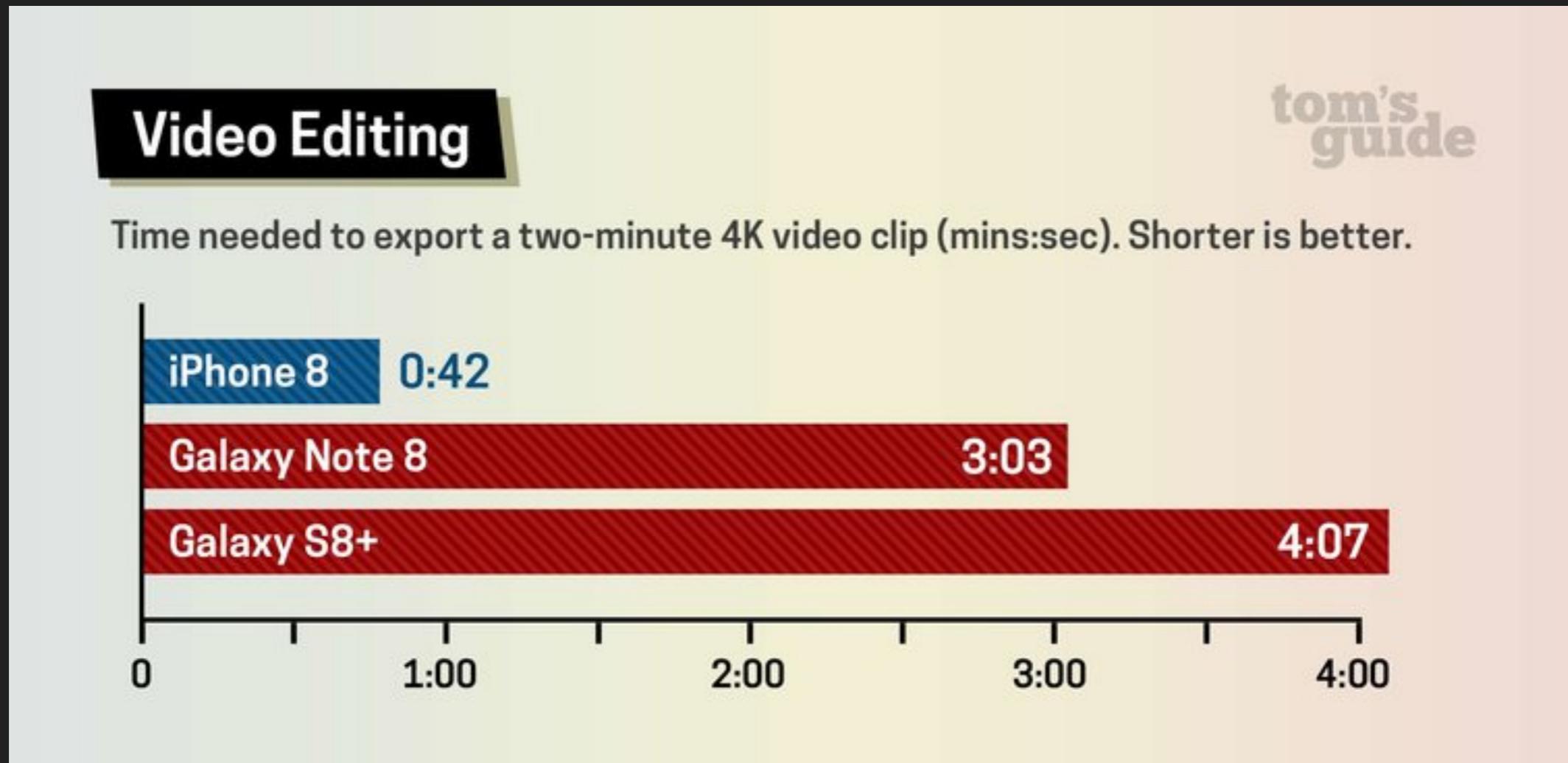
*“the short is answer is yes that the scores are comparable across platforms, so if an iPhone 8 scores higher than an i5, then the iPhone 8 is faster than the i5” - John Poole, Founder of Geekbench*



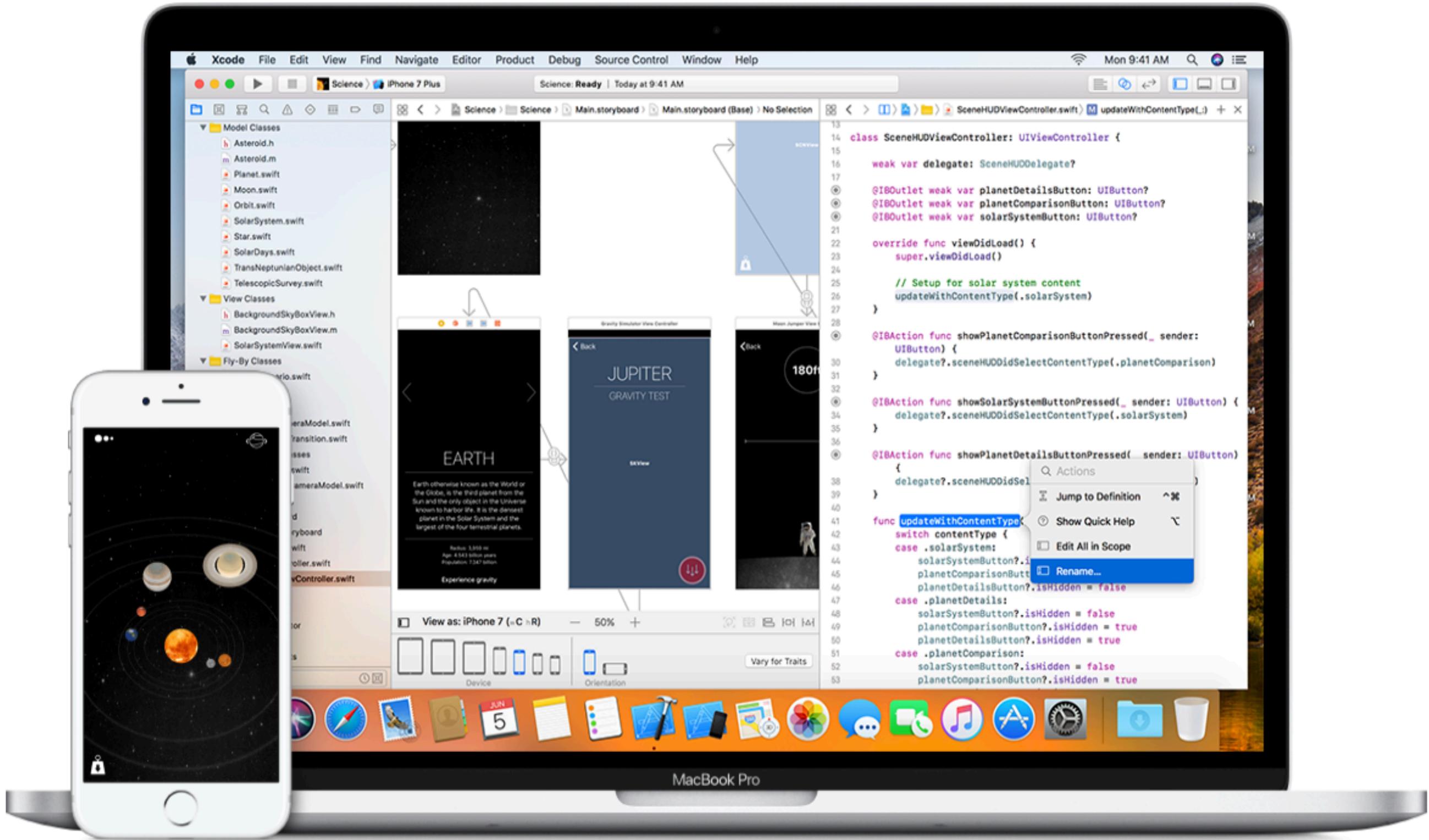
# iOS Entwicklung



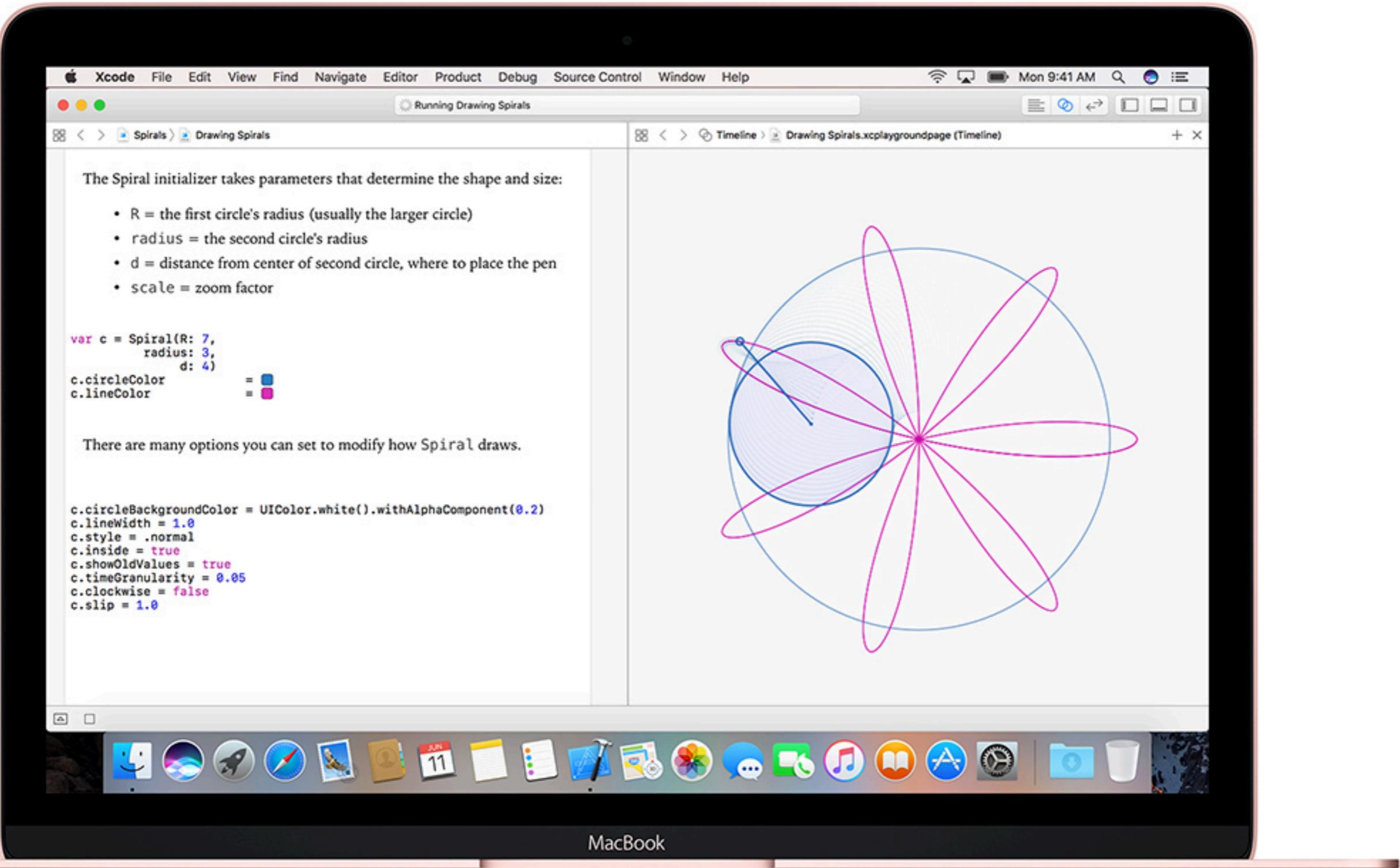
# iOS Entwicklung



# iOS Entwicklung



# iOS Entwicklung

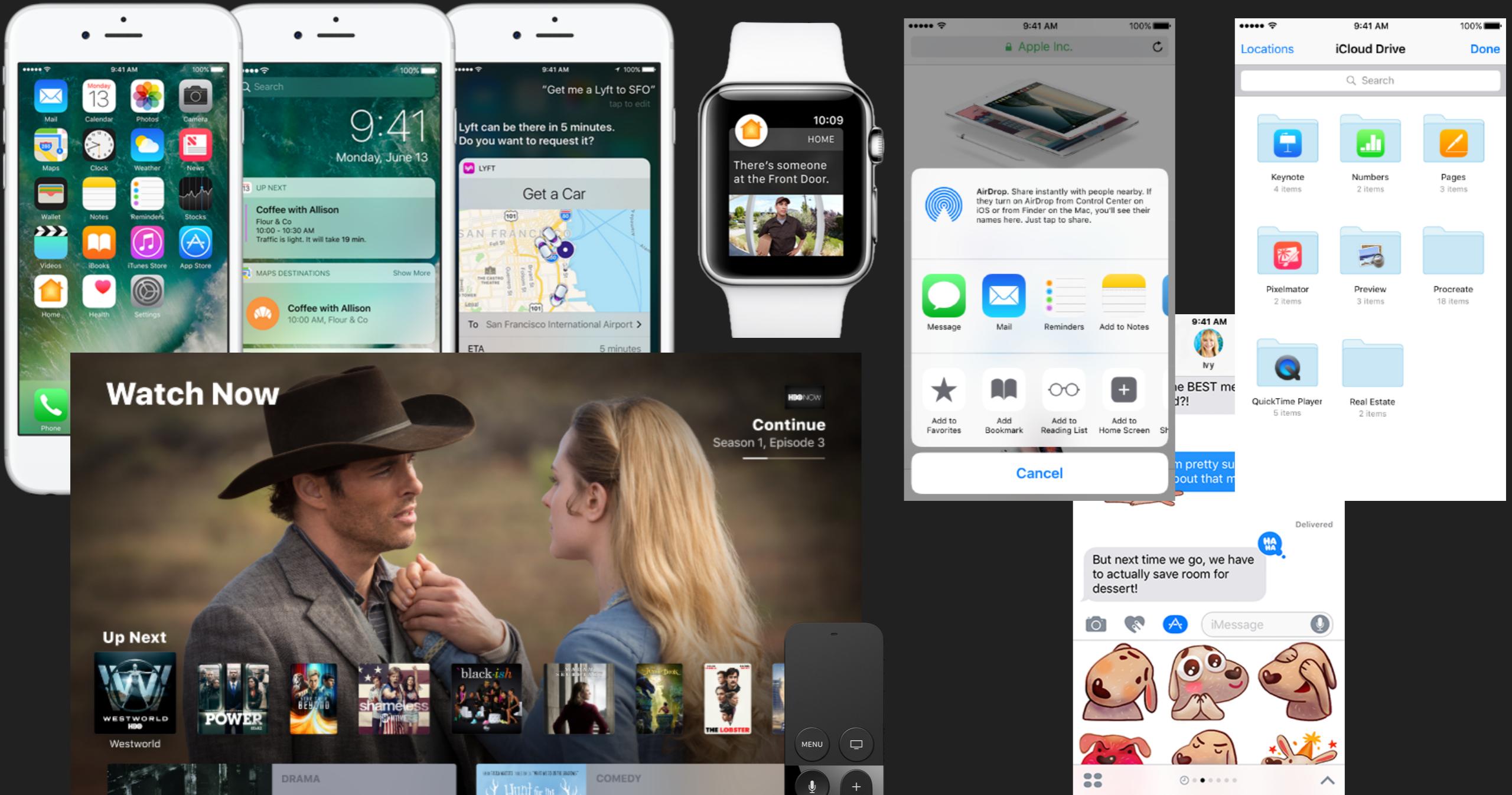


# iOS Entwicklung



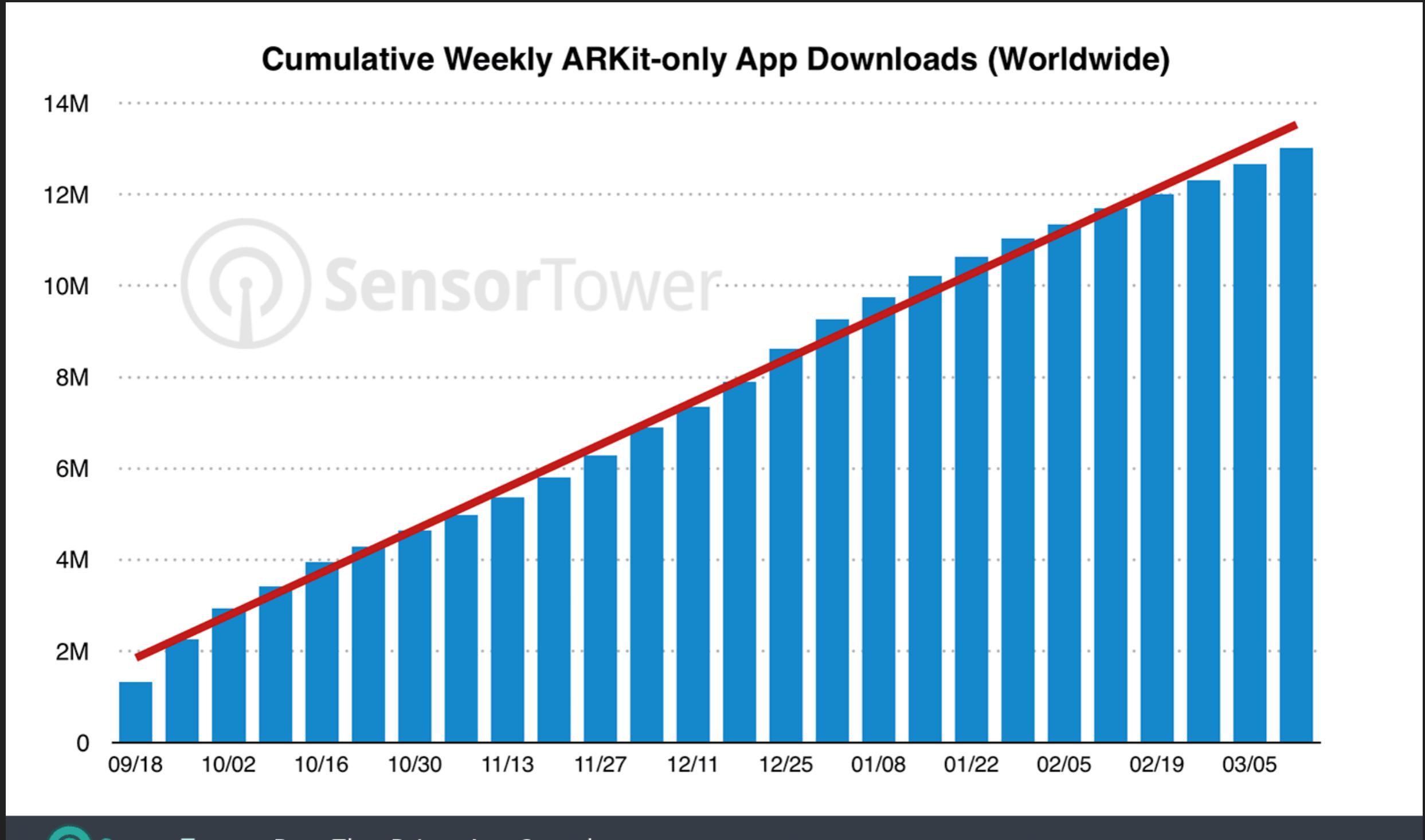
<https://developer.apple.com/swift-playgrounds/>

# 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/>

# iOS Entwicklung



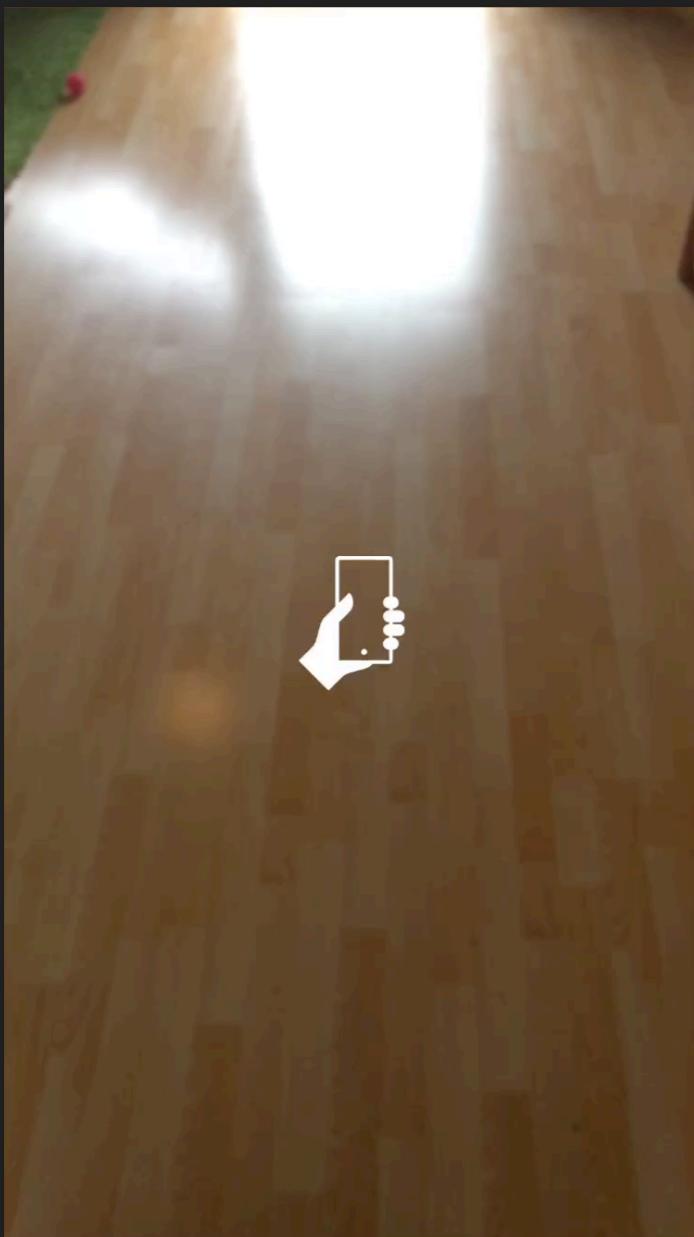
Sensor Tower Data That Drives App Growth

[sensortower.com](https://sensortower.com)

<https://sensortower.com/blog/arkit-six-months>

# 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)

# 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 ...*

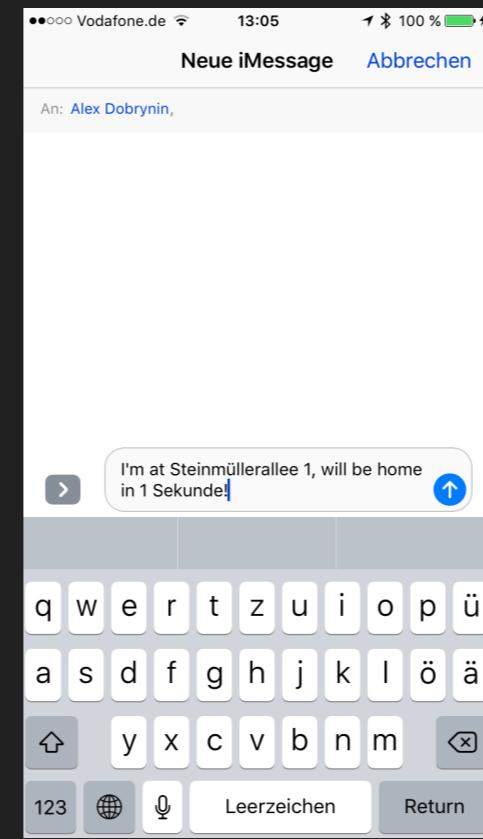
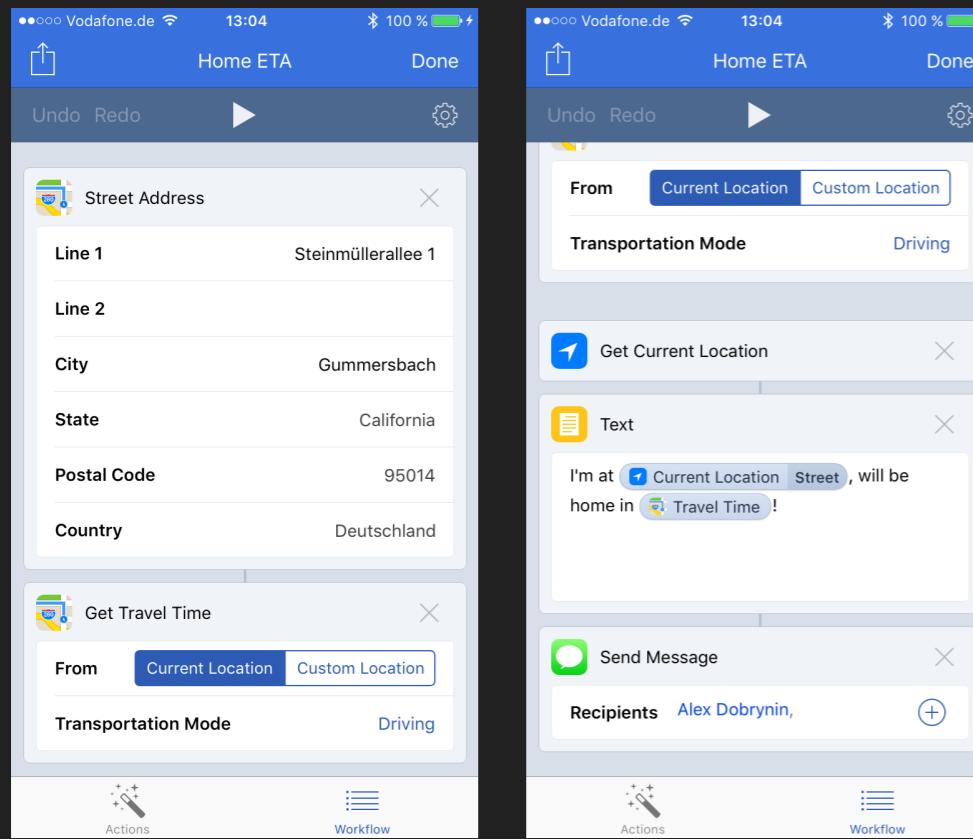
*a concrete and complex solution.*

1. Server, Network, and Hosting Environment + Agile Software Development
2. Data Modeling
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/>

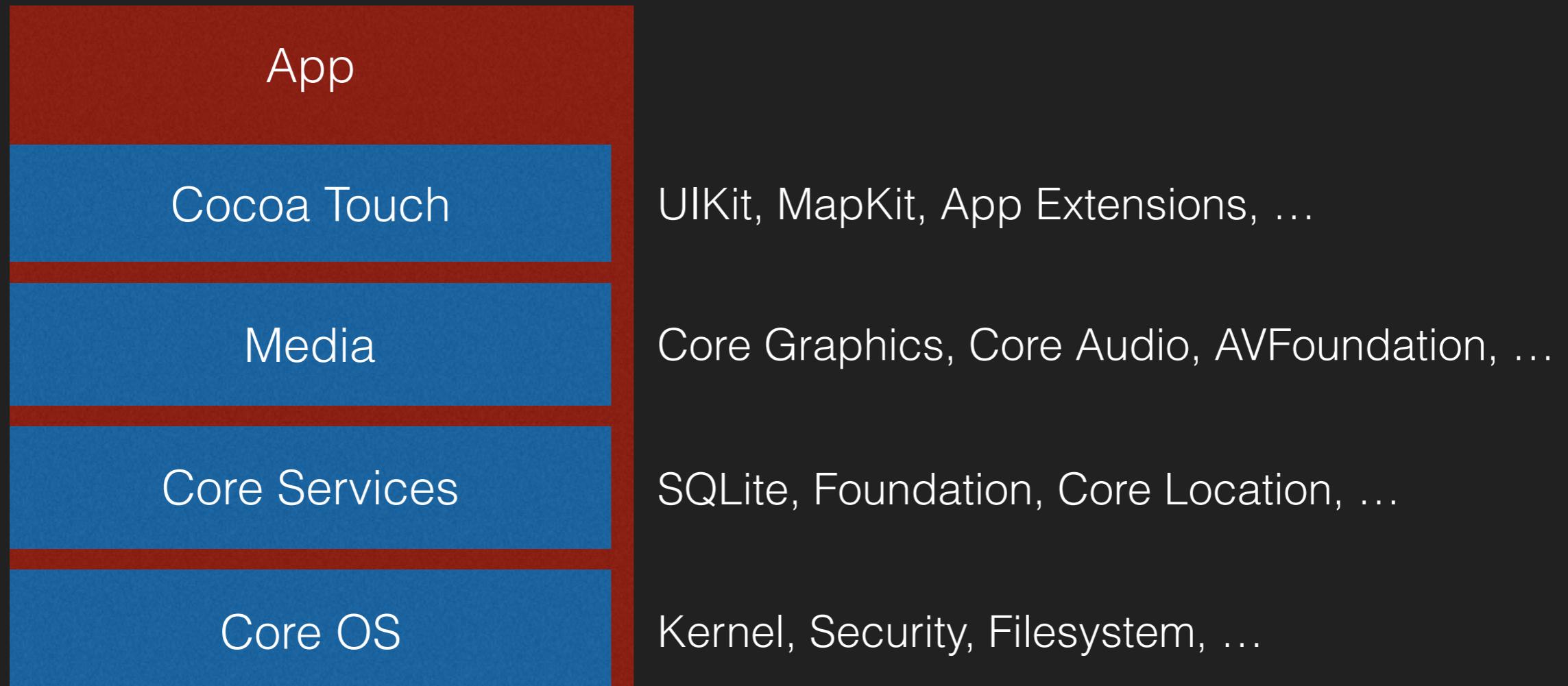
# Full Stack iOS Entwicklung

- 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>

# Übersicht von iOS



<https://developer.apple.com/library/content/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html>

# Übersicht von iOS

## Cocoa Touch

- Der Teil, der iOS ausmacht
- Entwickelt und optimiert für Objective-C
- Muss für beide Sprache kompatibel bleiben
- Nicht alle APIs sind “swifty”, weshalb man gelegentlich seltsamen Swift Code schreiben muss

## Swift

- Einer der zwei high-level Programmiersprachen für iOS Entwicklung
- Mittlerweile relativ stabil
- Unabhängig von Cocoa-Touch
- In Swift können bspw. Kommandozeilen-Skripte oder Server Anwendungen geschrieben werden

# 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

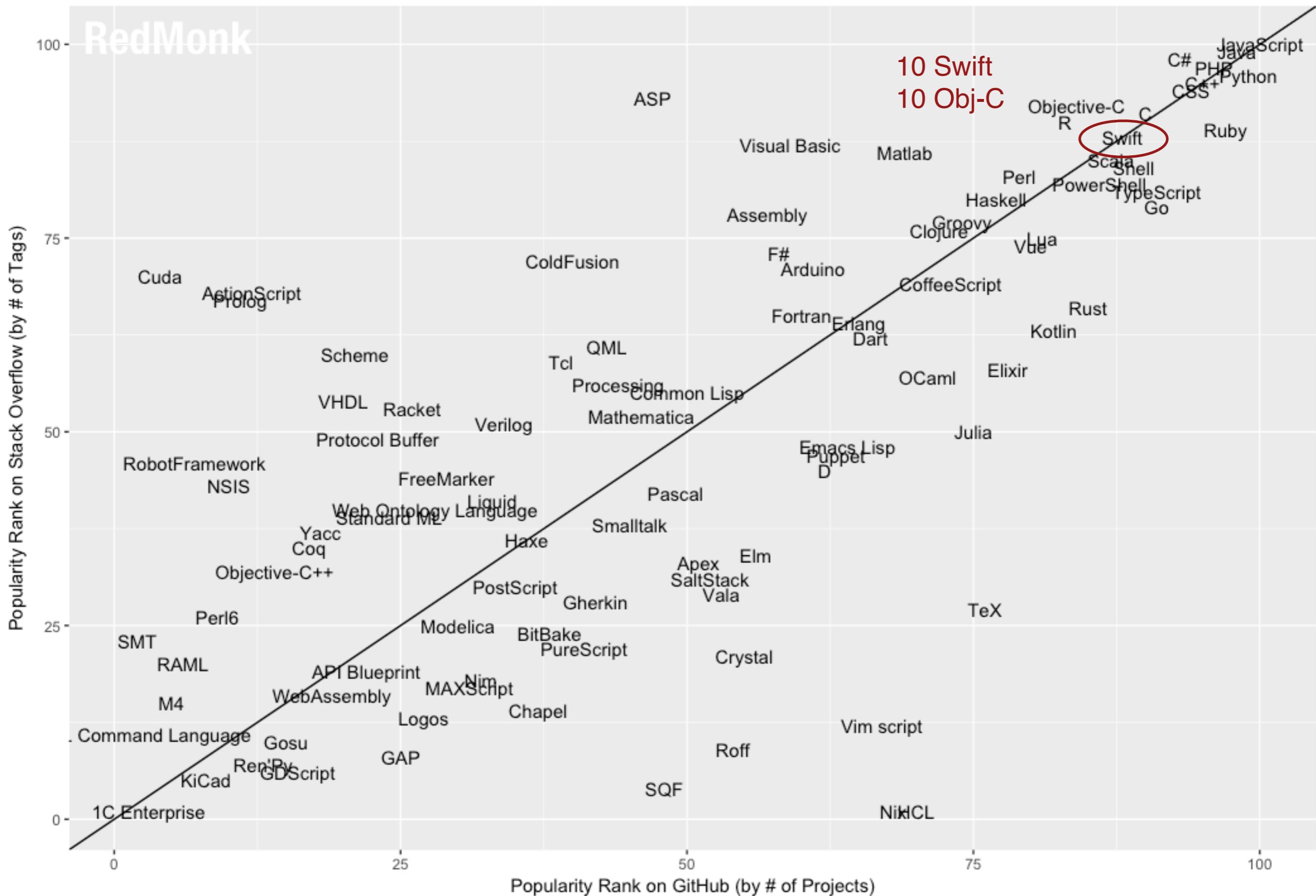
- Vorgestellt auf der WWDC am 02.06.2014
- Statisch typisierte, multiparadigmen Programmiersprache
- “Objective-C without the C”
- Mittlerweile in der Version 4.1 und Open-Source
- Community partizipiert an der Entwicklung von Swift (Swift-Evolution)
- Auf lange Sicht als primäre Programmiersprache für 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

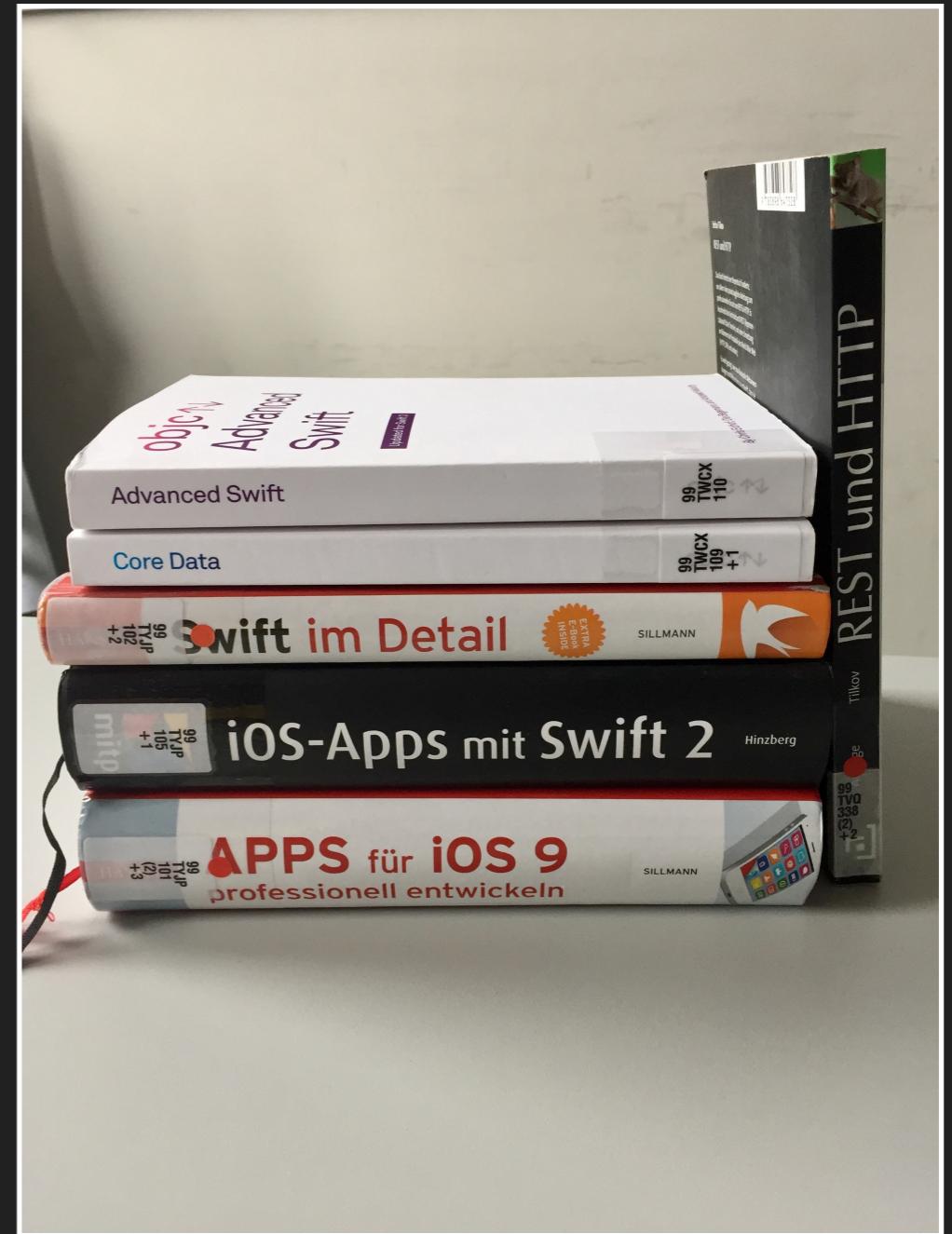
- Structs, Klassen und Enums
- Methoden, Funktionen und Properties
- `func` `dismissKeyboard() { ... }`, `func` `remove(element: Element) -> Element? { return nil }`
- Basistypen wie String, Int, Double, Float, ... und Collections wie Array, Dictionary, ...
- Deklaration von Properties über `var` (mutable) oder `let` (immutable)
- Inference Engine leitet den Typen auf Basis der rechten Seite (rhs) ab (sofern vorhanden)
- `let course = "FSIOS"`
- Ansonsten werden Typen mit einem Doppelpunkt nach dem Variablenamen angegeben
- `var applications: [String] = []`
- Nicht vorhandene Werte (nullable Values) werden explizit als Optional mit dem Literal ? deklariert
- `var room: Int?; room = 3245`

# RedMonk Q118 Programming Language Rankings



# Ressourcen

- [Swift auf GitHub](#)
- [Swift Evolution](#)
- [Swift Programming Language Guide \(auch ePub\)](#)
- [Start Developing iOS Apps \(Swift\)](#)
- [Apple Developer Sample Code](#)
- [Apple Human Interface Guidelines](#)
- [Apple WWDC 2017 Videos](#)
- [Developing iOS 11 Apps with Swift \(Stanford University Course\)](#)
- [Ray Wenderlich - iOS and Swift Tutorials](#)
- [AppCoda - Learn iOS Programming](#)
- [CodePath iOS Guides](#)
- [Awesome iOS - Courses, Frameworks, Tools, Xcode, Good Websites, Podcasts, Books](#)



# Heute

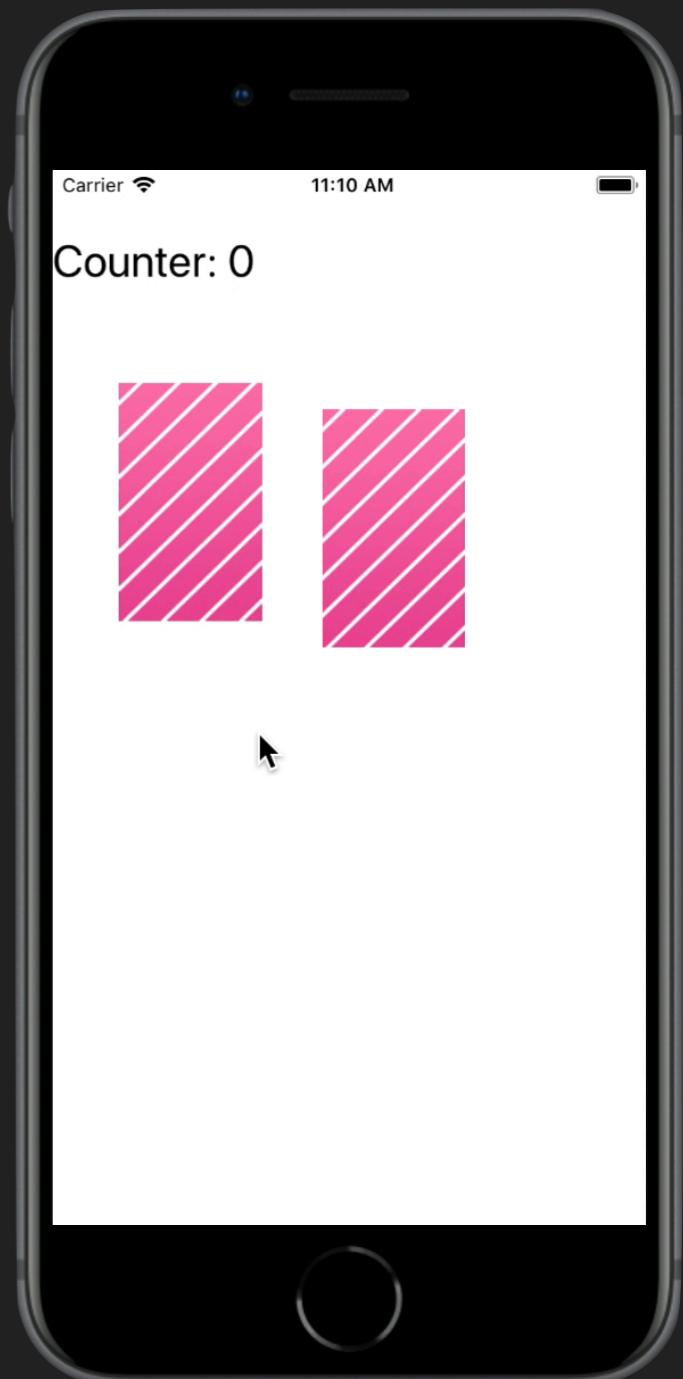
Über Mich / Euch  
Über die Veranstaltung

iOS Entwicklung  
Übersicht von iOS  
Swift  
Ressourcen

Demo

# FlipCards - Demo

- Xcode, Simulator, Storyboard (UI-Design)
- UIButton, UILabel, UIGestureRecognizer
- IBOutlet, IBCollection, IBAction
- Klassen, Funktionen, Properties, Typen und Optionals in Swift
- Verbindung zwischen Storyboard (UI) und ViewController (Code)
- Erste Grundlegende Datenstrukturen, Konstrukte und Sprachfeatures von Swift



# FlipCards - Assignment

- Zwei weitere Karten, welche in einer Outlet-Collection verwaltet werden
- Tap-Gesture flipped und unflipped die Karte immer
- Swipe funktioniert jeweils nur in eine Richtung
  - Swipe nach Rechts macht ein flip
  - Swipe nach Links macht ein unflip
- Reset-Button welcher das gesamte Spiel (Karten und Counter) zurücksetzt

