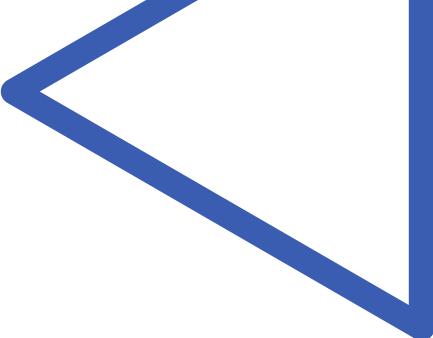




Swift

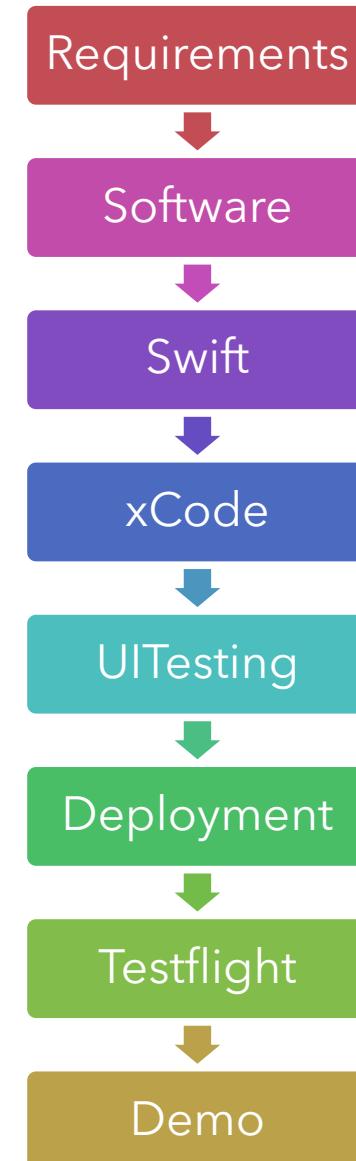


Developer



# iOS App Development Lifecycle

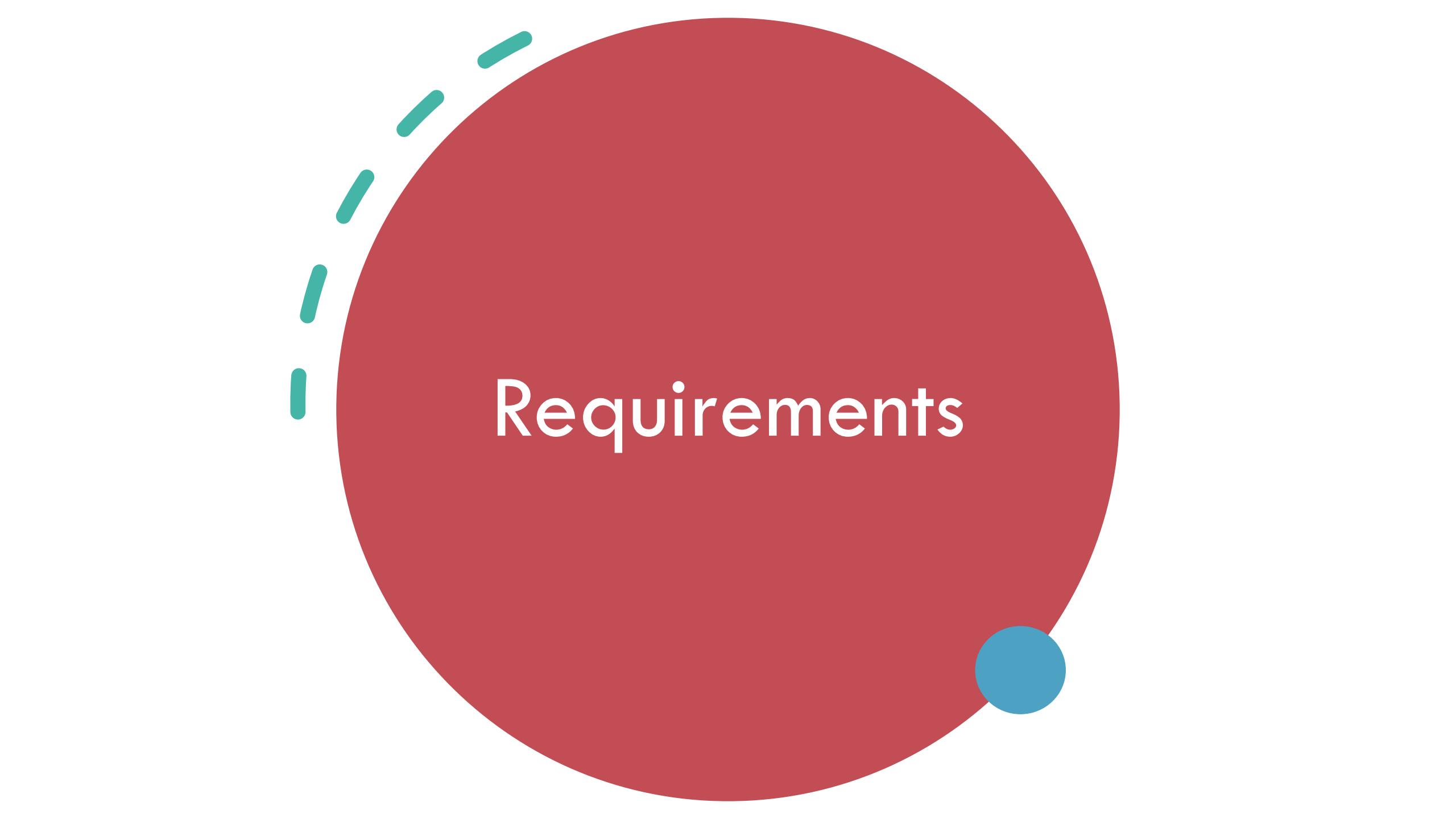
# Contents



## A short notice

This e-Portfolio is not about learning swift and it's best practices, it is simply to show how an iOS App joins the app store and which features apple provides to their developers to fulfill software engineering best practices.





Requirements

# Requirements



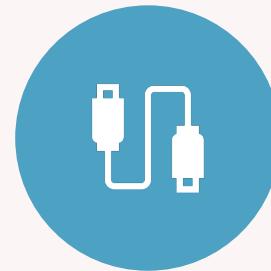
APPLE ID



OPERATING SYSTEM  
OS X



LICENSE COSTS  
\$99/YEAR  
(OPTIONAL)

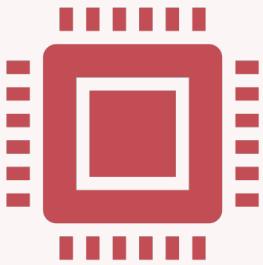


IPHONE / IPAD



Software

# Software



## On your Mac

xCode

iOS Simulator



## On your iPhone / iPad

Just a non jailbreak version of iOS

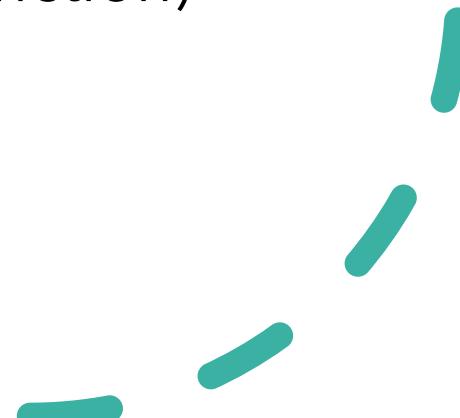


Swift

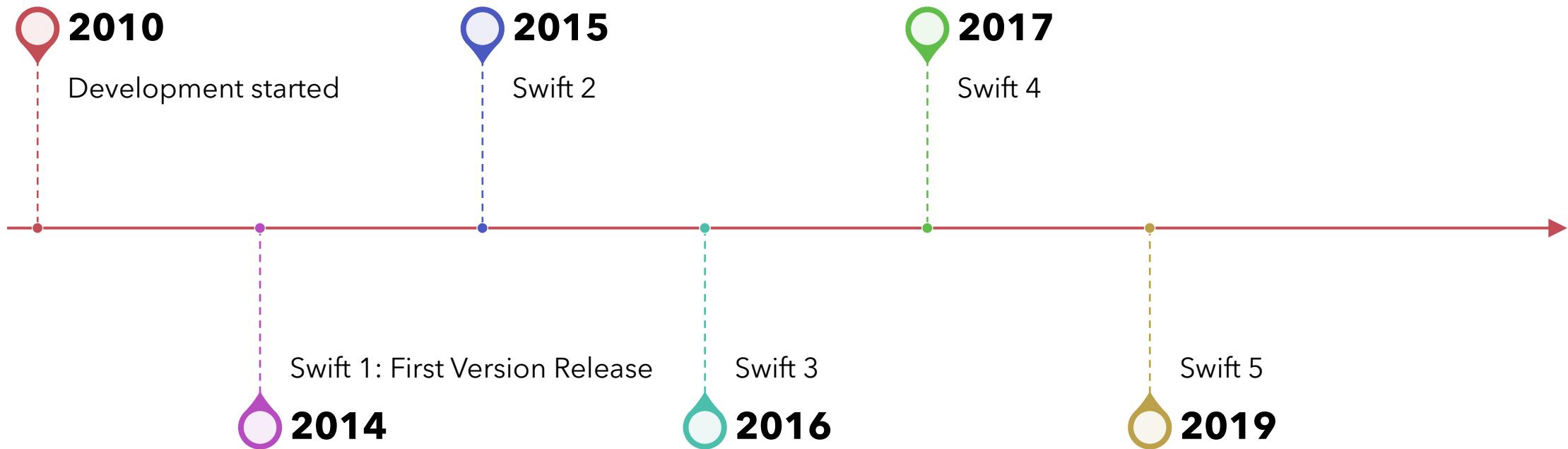


# Swift – Quick Facts

- Object oriented language
- Model View Controller Principle
- Objective-C Code allowed
  - `@objc func myAwesomeObjcFunction(){  
 //Code in Objective-C }`
  - Call by:  
`#selector(myAwesomeObjcFunction)`  
**everywhere** in Swift Code



# Swift - History



# Swift – Code samples I

- Initialization of Variable

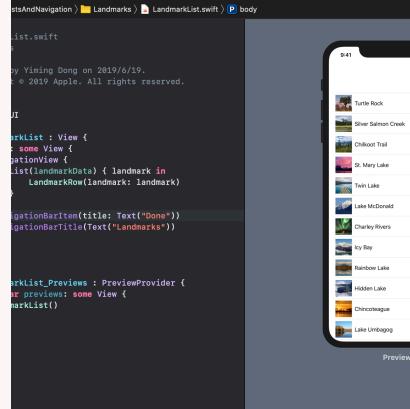
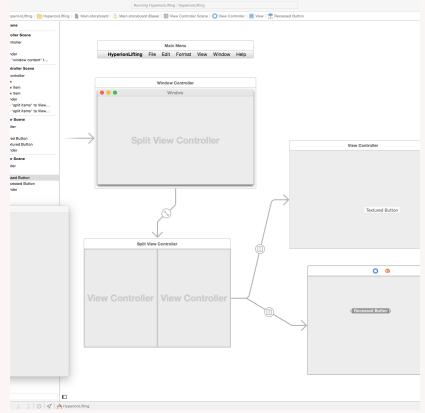
- var someFullyChangable : String = "" // has to be a String
- var anotherFullyChangable : String? // could be a String
- let changableProperties : People? //you can't change the object only properties (next Slide)

# Swift – Code samples II

A sample structure:

```
struct People : Codable { // If you want to load a json file into
    var name : String      your prog, you must create a
    var address : Address   codable object
    var postal : Int
    var town : String
}
```

# Swift – User Interfaces

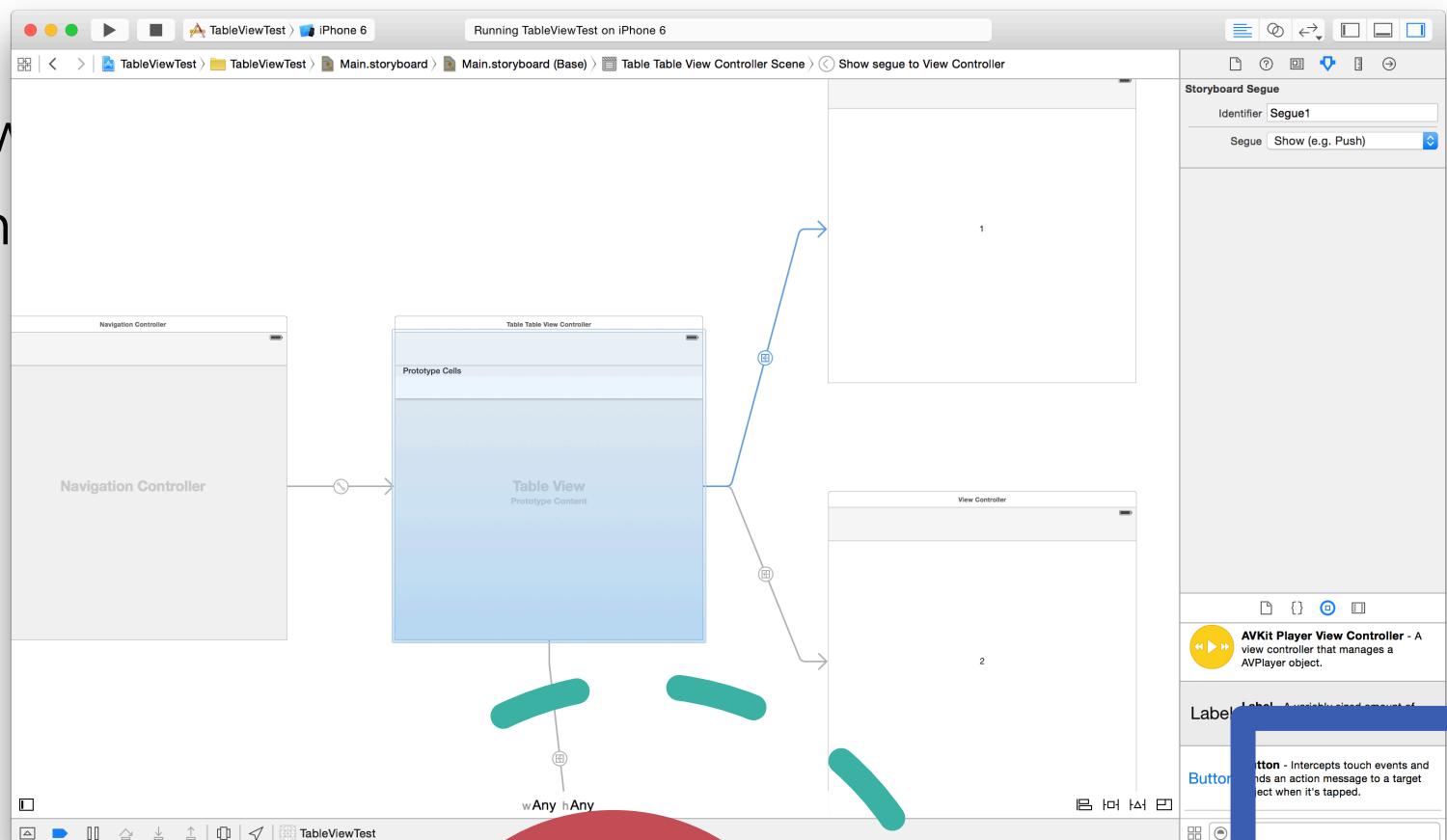


# UIKit

# SwiftUI

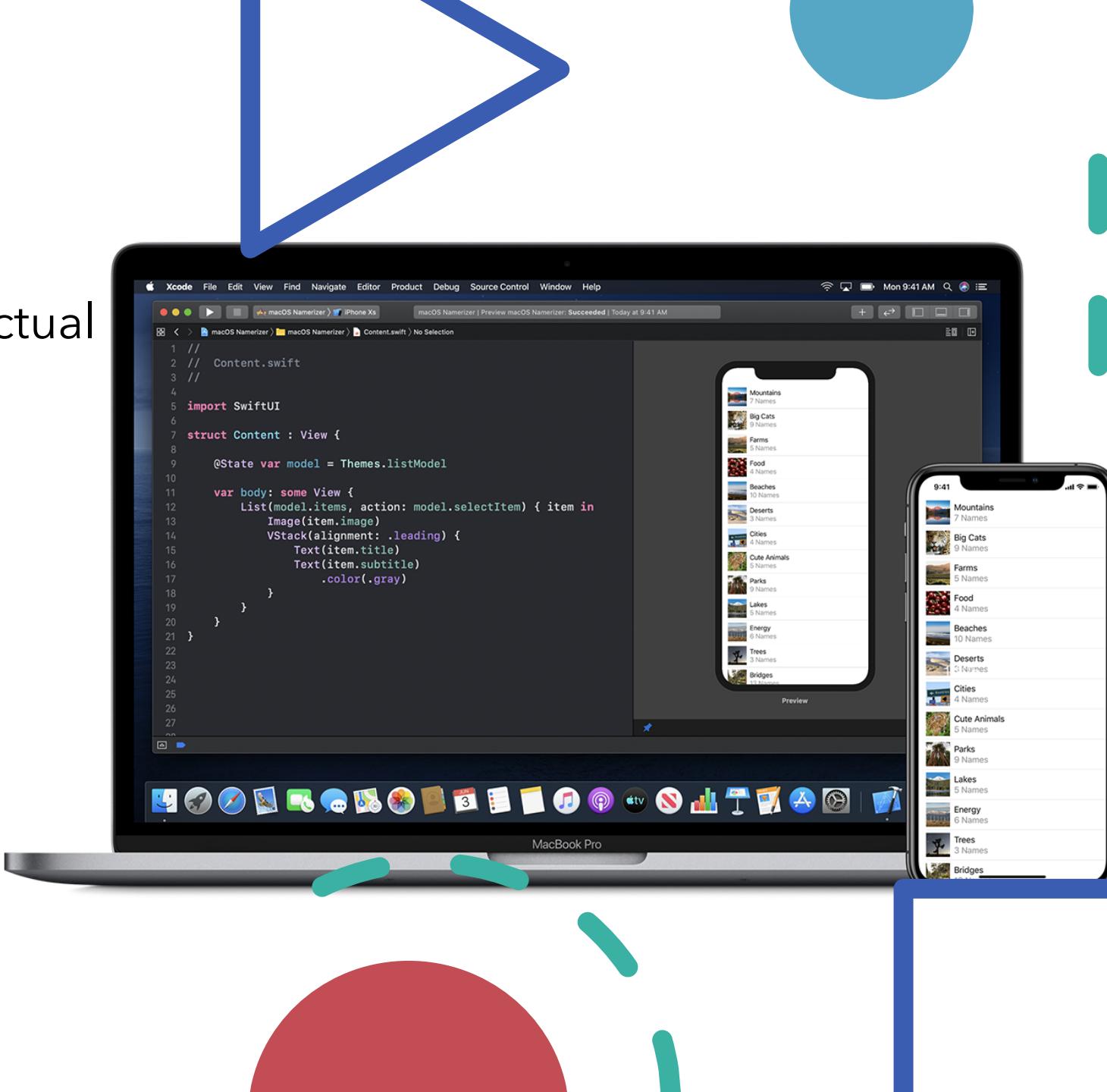
# Swift - UIKit

- A View is a “storyboard”
- Storyboards got introduced in 5 One Storyboard for One View
- Navigation Controller between those Views
- In Replacement -> SwiftUI



# SwiftUI

- Implemented since iOS 13 (actual version)
- Uses Model View Controller principle



# SwiftUI

Horizontal, Vertical and 3-rd dimension orientation by:

- HStack
- VStack
- ZStack

Uses UI design guidelines of Apple automatically

- Different font sizes, correct font
- Full support for darkmodes
- Perfect sizing for all Apple devices (including Mac Apps)

# xCode



# xCode 1/2



IDE to develop Swift and Objective-C Applications



Developed by Apple



Included Simulator for iPhone and iPad



Direct integration of git

# xCode 2/2



INTEGRATED  
UITESTING



INTEGRATED CODE  
TESTING



LIVE PREVIEW OF UI  
DESIGN



UI Testing

# UI Testing

- Apple implements a UI Testing library called XCTest
- You can create Test classes with the project

# Example UI Testing Code

- **func** testCheckIfSomeThingWorks() {
- // UI tests must launch the application that they test.
- **let** app = XCUIApplication()
- app.launch()
- //You can simply access a button by its label / text
- app.buttons["ButtonText"].tap()
- //Or test with expectations to be true or false as well as wait for a code
- //For reference: chevron.up is a system icon (arrow up)
- XCTAssertTrue(app.buttons["chevron.up"].waitForExistence(timeout: 10))
- }

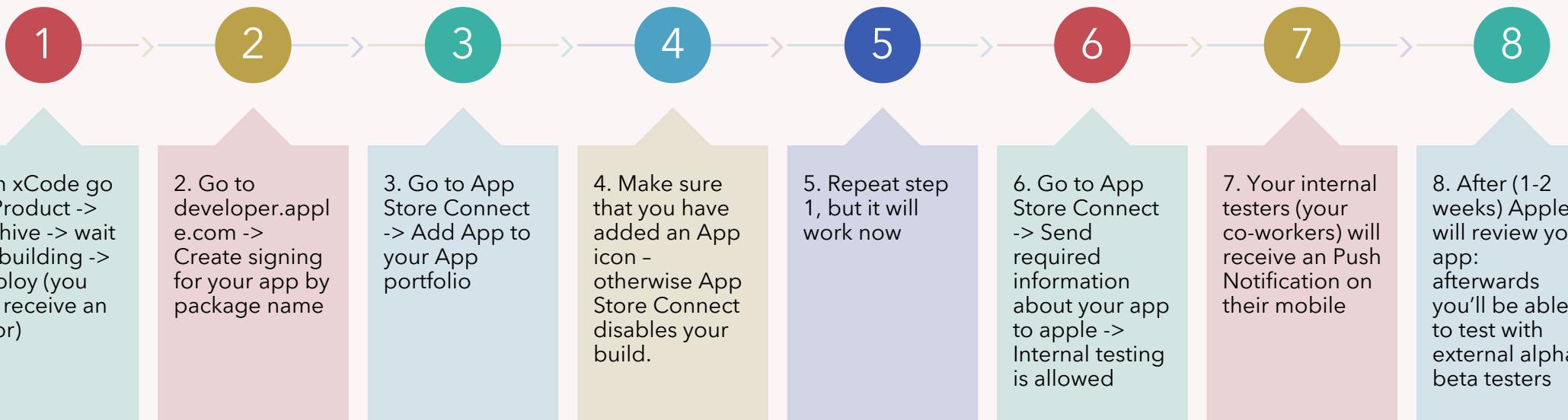
# Testflight



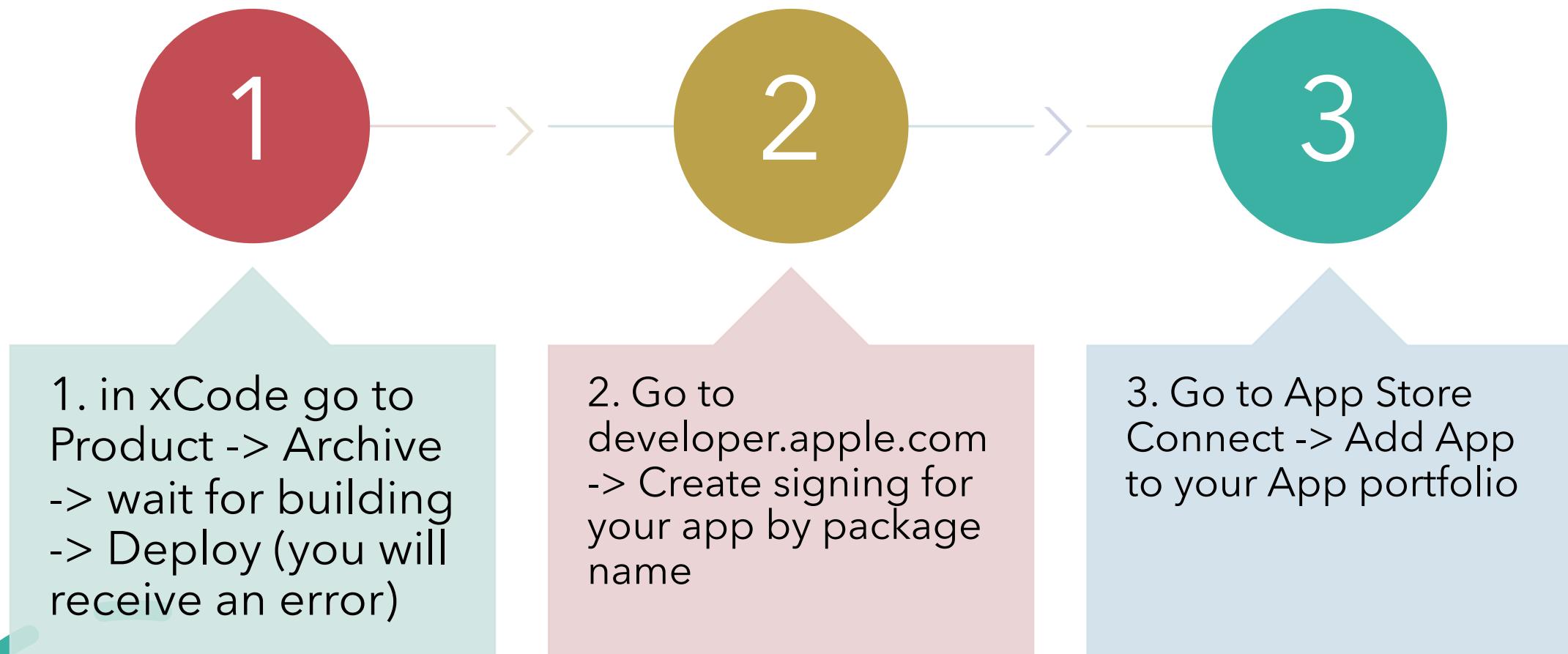
# Testflight

- App in App Store to download experimental apps
- You can invite up to 10.000 alpha / beta testers
- App will be checked by apple -> takes up to 2 weeks

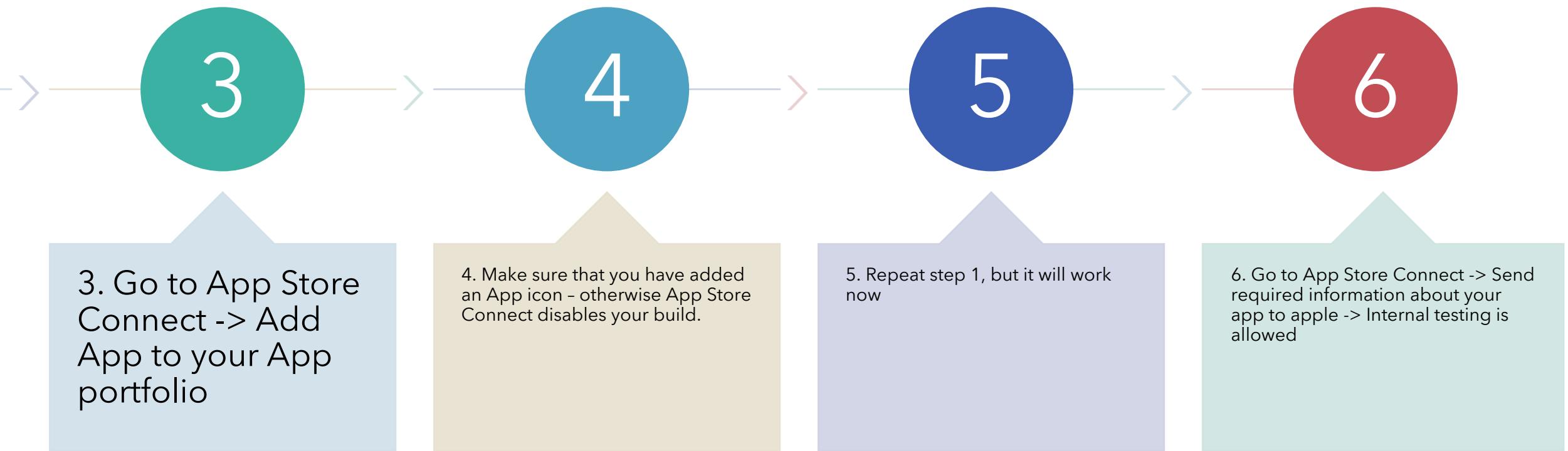
# Testflight – Process



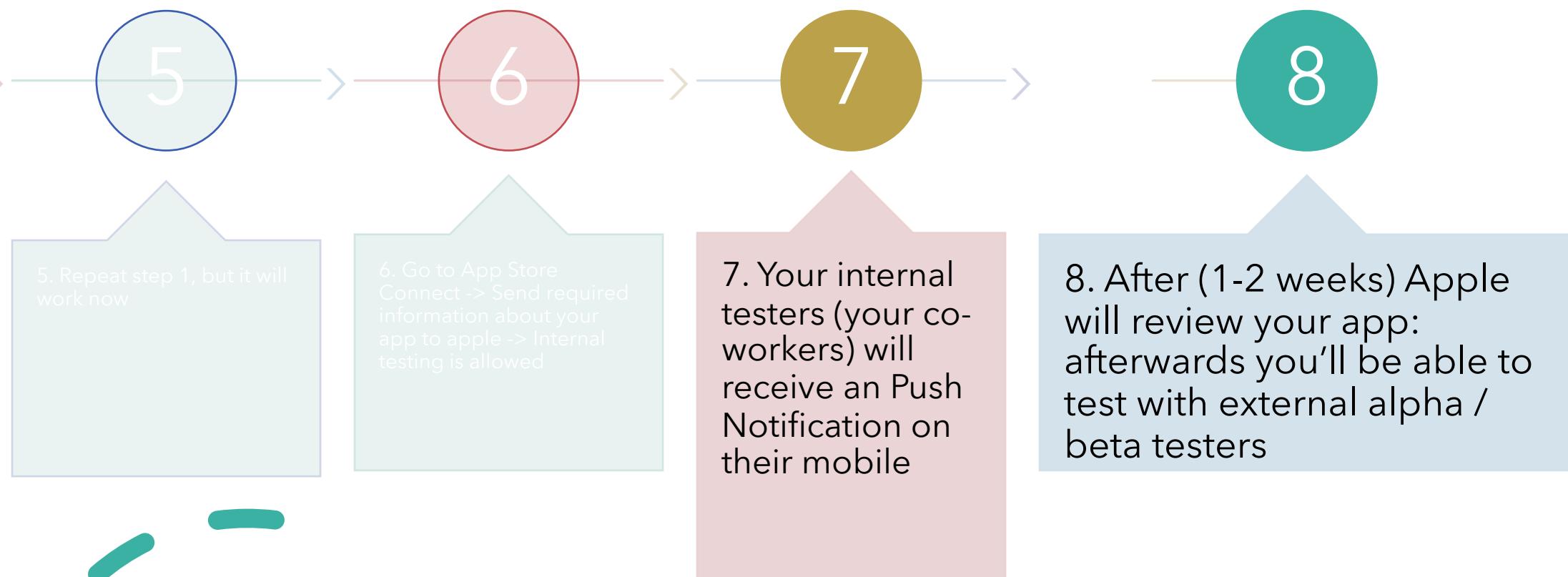
# Testflight / Build – Process



# Testflight / Build – Process



# Testflight / Build – Process

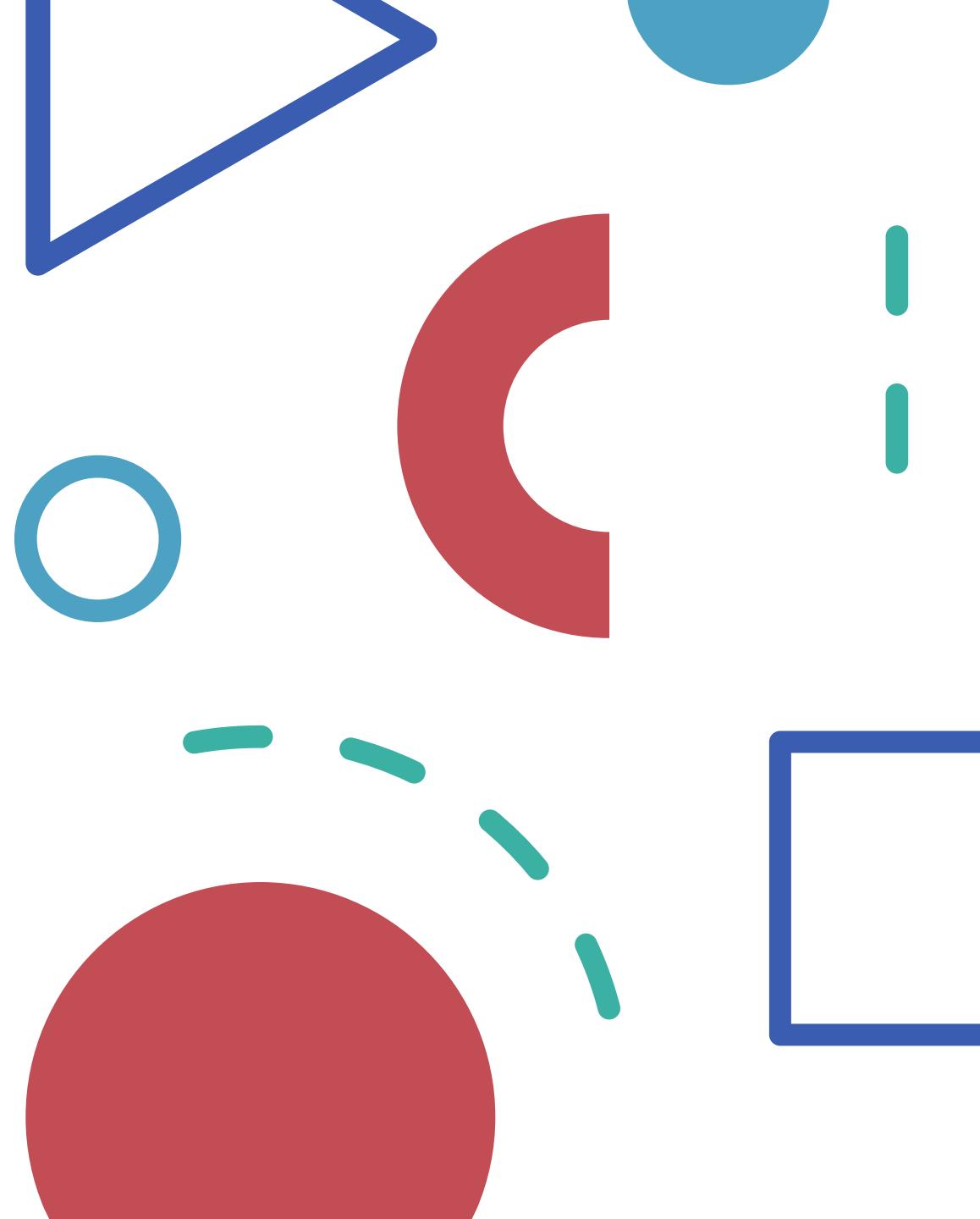


# Testflight - Features

- Your participants can give direct feedback or report bugs by taking a screenshot
- You see on how much devices your application is installed
- You'll get direct feedback
- You'll receive direct crash reports

App Store Connect-Benutzer	Externe Tester	Einladungen	Installationen	Letzte 7 Tage	Crashes	Feedback
<span>● Im Test</span> Läuft in 78 Tagen ab	<span>● Genehmigt</span> Läuft in 78 Tagen ab	1	2			

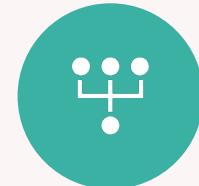
# Demo



# What we will do



1. Create a project



2. Implement a codable structure



3. Build the application and deploy



4. Implement a ListView with SwiftUI



5. Load mock data



6. Test the application

# Sources

- Swift (May 4, 2020)  
[https://de.wikipedia.org/wiki/Swift\\_\(Programmiersprache\)](https://de.wikipedia.org/wiki/Swift_(Programmiersprache))
- Apple Developer (May 6, 2020)  
<https://developer.apple.com>
- Objective-C (May 7, 2020)  
<https://en.wikipedia.org/wiki/Objective-C>
- What is a storyboard? (May 11, 2020)  
<https://www.hackingwithswift.com/example-code/language/what-is-a-storyboard>

# Might be helpful

- SwiftRocks <https://swiftrocks.com>
- HackingWithSwift <https://www.hackingwithswift.com>
- Apple Developer <https://developer.apple.com>
- RayWenderlich (iOS & Android)  
<https://www.raywenderlich.com/ios>

# The end

