Improving Existing Apps
Using modern best practices
Session 213

Woody L., 🐳 to the Knowledge
Agenda

Reduce Technical Debt
Asset Catalogs
Dependency Injection
Live Playgrounds
Cycle of Development

You down with ‘Dub-DC? Yeah, you know me.
Lots of Requests

Your boss
More Requests
Your customers
Technical Debt
//TODO: Write and clean up

Customer’s Perspective

Actuality
New API
New and Updated Platforms

macOS  iOS  tvOS  watchOS
A Dev’s Run Loop

- Bug Fixes
- Technical Debt
- Platforms
- New and Updated APIs
- Customer Roadmap
A Dev’s Run Loop

- Bug Fixes
- Technical Debt
- New and Updated APIs
- Platforms
- Customer Roadmap
A Dev’s Run Loop

- Bug Fixes
- Technical Debt
- New and Updated APIs
- Platforms
- Customer Roadmap
The Essentials

A very good place to start
Minimum Deployment of iOS 8

- 95% of Devices

As measured by the App Store on May 9, 2016
Pick a Deployment Target
Latest update of previous release
Deprecated API

- 'drawAtPoint:withFont:' is deprecated: first deprecated in iOS 7.0 - Use -drawAtPoint:withAttributes:
- 'sizeWithFont:' is deprecated: first deprecated in iOS 7.0 - Use -sizeWithAttributes:
- 'drawAtPoint:withFont:' is deprecated: first deprecated in iOS 7.0 - Use -drawAtPoint:withAttributes:

- Implicit conversion loses integer precision: 'NSUInteger (aka 'unsigned long') to 'int'
Deprecated API

'drawAtPoint:withFont:' is deprecated: first deprecated in iOS 7.0 - Use -drawAtPoint:withAttributes:

Implicit conversion loses integer precision: 'NSUInteger' (aka 'unsigned long') to 'int'
Treat Warnings as Errors

LLVM—Warning Policy, Xcode 7.3
Treat Warnings as Errors
Swift Compiler—Warning Policy, Xcode 8
Accessibility in Xcode
As much a part of your user interface as the artwork
Accessibility in Xcode
As much a part of your user interface as the artwork
// Programmatic Accessibility of UI Components

let button = UIButton()

button.accessibilityLabel = "Add Bookmark"
## Locale-Aware APIs

<table>
<thead>
<tr>
<th>API</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Locale</strong></td>
<td>Obtain current region and format</td>
</tr>
<tr>
<td><strong>DateFormatter</strong></td>
<td>Format and parse dates and times</td>
</tr>
<tr>
<td><strong>NumberFormatter</strong></td>
<td>Format and parse numbers</td>
</tr>
<tr>
<td><strong>Calendar and TimeZone</strong></td>
<td>Current calendar and associated operations</td>
</tr>
<tr>
<td><strong>Dimension</strong></td>
<td>Represent specific units of measure</td>
</tr>
<tr>
<td><strong>MeasurementFormatter</strong></td>
<td>Format and parse units of quantity</td>
</tr>
<tr>
<td><strong>PersonNameComponents</strong></td>
<td>Names of people</td>
</tr>
</tbody>
</table>
## Locale-Aware APIs

<table>
<thead>
<tr>
<th>API</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Locale</strong></td>
<td>Obtain current region and format</td>
</tr>
<tr>
<td><strong>DateFormatter</strong></td>
<td>Format and parse dates and times</td>
</tr>
<tr>
<td><strong>NumberFormatter</strong></td>
<td>Format and parse numbers</td>
</tr>
<tr>
<td><strong>Calendar and TimeZone</strong></td>
<td>Current calendar and associated operations</td>
</tr>
<tr>
<td><strong>Dimension</strong></td>
<td>Represent specific units of measure</td>
</tr>
<tr>
<td><strong>MeasurementFormatter</strong></td>
<td>Format and parse units of quantity</td>
</tr>
<tr>
<td><strong>PersonNameComponents</strong></td>
<td>Names of people</td>
</tr>
</tbody>
</table>
Add Peek, Pop, and Quick Actions
Getting Ready for Autumn 2016

Checklist for developer preview seeds

Run the Swift Migrator

Does it work?
• No? Is it Apple’s problem or is it your code?
• Are you using APIs in a way we didn’t expect?
• Unit tests pass?
File Bug Reports

bugreport.apple.com
File Bug Reports
bugreport.apple.com

Banter
MobileMe?
Broken!

Dev Forums

Email
BugReport.apple.com

New iOS Problem

iOS

Please include only one issue per report. All fields are required except where noted. While we cannot respond directly to every report, all reports are reviewed by the appropriate engineering teams.

Problem will be AutoSaved in 2 minutes.

Classification*  Serious Bug
Reproducibility*  Always

Please be sure your issue does not better fit under another more specific classification.

Show instructions for gathering logs

Attach a file

Title: Provide a short but descriptive sentence that summarizes the issue

Example: Compass app shows the wrong direction
Where Do Lonely Bug Reports Go?
“La-La, Can’t Hear You”
“Each bug report is as unique as a snowflake.”

Paul M., Apple Software Engineer
**Get Started Today**

Prepare your codebase

<table>
<thead>
<tr>
<th>iOS 9.3 + Xcode 7</th>
<th>iOS 10 + Xcode 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update project settings</td>
<td>Use the Swift Migrator</td>
</tr>
<tr>
<td>Fix warnings</td>
<td>Provide bug reports</td>
</tr>
<tr>
<td>Replace deprecated API</td>
<td>Incorporate new API</td>
</tr>
<tr>
<td>Localize</td>
<td></td>
</tr>
<tr>
<td>Accessibility</td>
<td></td>
</tr>
</tbody>
</table>
Asset Catalogs

Or, as some might write, Catalogues
The Old Way
Files with naming conventions
The Modern Way
Asset catalog
Add a Catalog

Asset Catalog
Asset catalogs store and categorize resources for different platforms, devices, and capabilities (such as scale factors). When built, items in asset catalogs are compiled into a unified, efficient runtime format or exported to their expected format (based on use).
Multiple Asset Catalogs
Multiple Asset Catalogs
Migrating Project Images
Migrating Project Images
Migrating Project Images
Migrating Project Images
Migration Completed
// Loading Image Assets

// Old Way
if let logo = UIImage(contentsOfFile: Bundle.main().pathForResource("Logo", ofType: "png")) {
    imageView.image = logo
}

// Modern Way
if let logo = UIImage(named: "List") {
    imageView.image = logo
}
// Loading Image Assets

// Old Way
if let logo = UIImage(contentsOfFile: Bundle.main().pathForResource("Logo", ofType: "png")) {
    imageView.image = logo
}

// Modern Way
if let logo = UIImage(named: "List") {
    imageView.image = logo
}
// Loading Image Assets

// Old Way
if let logo = UIImage(contentsOfFile: Bundle.main().pathForResource("Logo", ofType: "png")) {
  imageView.image = logo
}

// Modern Way
if let logo = UIImage(named: "List") {
  imageView.image = logo
}
// Loading Image Assets

// Old Way
if let logo = UIImage(contentsOfFile: Bundle.main().pathForResource("Logo", ofType: "png")) {
    imageView.image = logo
}

// Modern Way
if let logo = UIImage(named: "List") {
    imageView.image = logo
}
Multiple Representations of Assets
Multiple Representations of Assets
Multiple Representations of Assets
Multiple Representations of Assets
Multiple Representations of Assets
Multiple Representations of Assets
Multiple Representations of Assets
Multiple Representations of Assets
Multiple Representations of Assets
Asset Types

Individual Scales
- PNGs

Single Vector
- PDFs
Selecting the Scale Factor
Missing Scaled Asset Representations
No 2x and 3x artwork
Missing Scaled Asset Representations
No 2x and 3x artwork
Missing PNG Asset Representations
Only 3x media

Missing 1x and 2x artwork

Scaled down for 1x and 2x displays
Asset Scaling and Memory

1x
9Kb

3x
943Kb
Memory Pressure

Temporary Memory Spike

Start 3x Image Load

Image Opened and Scaling to 2x

Scaled, Original Deallocated
Memory Pressure

Temporary Memory Spike

- Start 3x Image Load
- Image Opened and Scaling to 2x
- Scaled, Original Deallocated
Memory Pressure

Temporary Memory Spike

Memory Utilization

Start 3x Image Load  Image Opened and Scaling to 2x  Scaled, Original DEALLOCATED
Automator
Vector Assets
Scalable to any size
Vector Assets
Scalable to any size
Vector Assets
Scaled and rasterized at build time
Vector Assets

Scaled and rasterized at build time
Override Universal Vector Assets
Improving asset image quality
Override Universal Vector Assets

Improving asset image quality
Image Compression
Lossy and lossless
Capped Images
Challenge of rounded corners

Continue
Capped Images
Challenge of rounded corners
Capped Images
Challenge of rounded corners

Continue
Capped Images
Challenge of rounded corners

Continue
Capped Images
Challenge of rounded corners

Continue
Capped Images
Challenge of rounded corners

Continue
Button Background Image
Interface Builder’s Attributes Inspector

![Interface Builder's Attributes Inspector for a Button with configurable settings including Type, State Config, Title, Font, Text Color, Shadow Color, Image, and Background.](image)
Button Background Image
Interface Builder’s Attributes Inspector
Button Background Image
Interface Builder’s Attributes Inspector
Capped Images
Challenge of rounded corners
Capped Images
Challenge of rounded corners
// Stretchable Images
// Programmatic, Traditional Way:

if let image = UIImage(named: "RoundedRectangle")
    let background = image.stretchableImage(withLeftCapWidth: 10, topCapHeight: 10) {
        button.setBackgroundImage(background, for: [])
    }
Asset Slicing
Asset Slicing
Don’t stretch these edges
Don’t stretch these edges
Repeat these pixels
Repeat these pixels
Capped Images
Preserving of rounded corners
Summary

Asset Catalogs

Use Asset Catalogs

Vector images preserve image quality

Scaled images should be provided for each asset representation (1x, 2x, 3x)

Perform slicing within the asset catalog
Dependency Injection

Pay it forward
UITextFieldDelegate

Methods relating to UITextField

UITextField

UITextFieldDelegate

func textFieldShouldBeginEditing(_ textField: UITextField) -> Bool

func textFieldDidBeginEditing(_ textField: UITextField)
Methods relating to WCSession

```swift
func session(_ session: WCSession,
    activationDidCompleteWith activationState: WCSessionActivationState,
    error: NSError?)
```
App Delegate

Everything else

NS/UIApplication

NS/ApplicationDelegate
func applicationWillResignActive(_ application: UIApplication)
func applicationWillResignActive(_: application: UIApplication)
var sqlite3DB: SQLite3Db?
func applicationWillResignActive(_ application: UIApplication)
var sqlite3DB: SQLite3Db?
var sink = Kitchen(garbageDisposal: true)
func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    return (UIApplication.shared().delegate as? AppDelegate)?.content?.count ?? 0
}
Reach-Back to the App Delegate

Tight coupling

Application Delegate [MyModel]

View Controller A
Mailboxes

View Controller B
Messages

View Controller C
Detail
Dependency Injection
Pay it forward

Application Delegate

MailboxesVC
[Mailbox]()

MessagesVC
[Message]?

MessageVC
Message?
override func prepare(for segue: UIStoryboardSegue, sender: AnyObject?) {
    if let messagesVC = segue.destinationViewController as? MessagesViewController {
        messagesVC.content = model
    }
}
@IBAction func showDetailVC(sender: AnyObject?) {
    let detailVC = DetailVC()
    let indexOfSelectedMessage = 0 // ...
    let message = messages[indexOfSelectedMessage]
    detailVC.message = message
}
Returning from the View Controller

Coming back from Dependency Injection

Create and implement a Protocol

- UIImagePickerControllerDelegate
- MFMailComposeViewControllerDelegate
- Mark the delegate property as weak to prevent memory leaks to prevent circular references

Pass a Closure to the Destination View Controller

Pass Model Objects by Reference

Unwind/Exit Segues, with `prepareForSegue:`
Previously at WWDC

Flashback to 2015

Once upon a time, in a session room not that far away...
Modernizing existing apps with Swift
Modernizing existing apps with Swift

- Playgrounds
- Modern UI
- Interoperability
Live Playgrounds!

Better than !(Live Playgrounds)
Demo

TableViews in Live Playgrounds

The Needle? We like to move it, move it.
Demo Summary

Live Playground supporting API

Import the Playground Module

```swift
import PlaygroundSupport
```

Indefinite/Asynchronous Execution

```
PlaygroundPage.current.needsIndefiniteExecution = true
```

Assign a Live View

```
PlaygroundPage.current.liveView = customTableViewController.tableView
```
Demo Summary

Tips

Playground Sources and Resources

- Files dragged to these folders are copied, not referenced
- Methods, properties, and data types in the Sources folder must use the Public specifier
Demo Summary
Caches folder

Temporary data that can be recreated if it’s missing

Thumbnails

Downloaded resources

Purged when device under storage pressure

```swift
let fm = FileManager.default()
let cacheUrl = fm.urlsForDirectory(.cachesDirectory, inDomains: .userDomainMask).last
```
NSURLSession

Powerful and performant

Out-of-process networking
Delegation and completion handlers
Supported by tvOS, iOS, macOS, and watchOS platforms
Backed by the expertise of our core networking engineers

NSURLSession: New Features and Best Practices    Pacific Heights    Thursday 10:00AM
Networking for the Modern Internet               Pacific Heights    Thursday 3:00PM
It's time to play the music, it's time to light the lights
// tvOS Conditional Compilation

#if os(tvOS)
    backgroundImageView.adjustsImageWhenAncestorFocused = true
#endif
Adopting Additional Platforms

Human Interface Guidelines

The new Apple TV redefines the living room experience, allowing you to deliver extraordinary, immersive content on the big screen like never before.

Overview
- Design Principles
- Apps
- Home Screen
- Top Shelf
- Focus and Parallax

Design Principles

Apple products are known for connecting people with content. This feeling of connection is expected from Apple TV too, even though it isn't a device you hold in your hand and you don't touch the screen directly. In many households, Apple TV sits in the living room, rather than in a pocket or on a desk. Apps can be used by individuals or entire families, they bring people together for entertainment and to provoke conversation. Apple TV is a unique platform with unique design requirements. Whether you're building a game, a streaming media app, or a utility for the home, keep the following principles in mind as you imagine your app's identity.
Invoicing
Financial Charts
Adding Platforms

Read and use our Human Interface Guidelines
Design to the platform
Pivot your data model
Summary

Modernizing your app is an on-going process
Rely on the frameworks
Start today—No need to wait for Xcode 8
Live Playgrounds allow for more experimentation
Architect your app with few inter-object dependencies
Consider bringing your model layer to our other platforms, with platform-specific UI
<table>
<thead>
<tr>
<th>Related Sessions</th>
<th>Location</th>
<th>Date/Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>What’s New in Cocoa Touch</td>
<td>Presidio</td>
<td>Tuesday 1:40PM</td>
</tr>
<tr>
<td>What’s New in tvOS</td>
<td>Presidio</td>
<td>Tuesday 3:00PM</td>
</tr>
<tr>
<td>What’s New in watchOS 3</td>
<td>Presidio</td>
<td>Tuesday 5:00PM</td>
</tr>
<tr>
<td>What’s New in Swift</td>
<td>Presidio</td>
<td>Tuesday 9:00AM</td>
</tr>
<tr>
<td>Crafting Modern Cocoa Apps</td>
<td>Pacific Heights</td>
<td>Friday 5:00PM</td>
</tr>
<tr>
<td>Increase Usage of Your App with Proactive Suggestions</td>
<td>Mission</td>
<td>Friday 1:40PM</td>
</tr>
<tr>
<td>Labs</td>
<td>Frameworks Lab</td>
<td>Time</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>----------------</td>
<td>---------------</td>
</tr>
<tr>
<td><strong>Cocoa Touch Lab</strong></td>
<td>A</td>
<td>Wednesday 3:00PM</td>
</tr>
<tr>
<td><strong>Cocoa Lab</strong></td>
<td>D</td>
<td>Thursday 2:00PM</td>
</tr>
<tr>
<td><strong>Interface Builder and Auto Layout Lab</strong></td>
<td>C</td>
<td>Friday 9:00AM</td>
</tr>
<tr>
<td><strong>UIKit and UIKit Animations Lab</strong></td>
<td>C</td>
<td>Thursday 1:00PM</td>
</tr>
<tr>
<td><strong>Swift Open Hours</strong></td>
<td>A</td>
<td>Daily</td>
</tr>
</tbody>
</table>
More Information
