



+



Avoiding Release Anxiety in iOS

Bruno Rocha



The information in this document is not confidential to the person to whom it is addressed, so feel free to disclose it to any other person. It may be reproduced in whole or in part, allowing you to disclose this information without the prior consent of the directors of iFood. It has been used as support material for an oral presentation and, therefore, it does not represent a complete record of the topics presented in the mentioned presentation.



Need to add a feature



Can't do it properly without refactoring everything



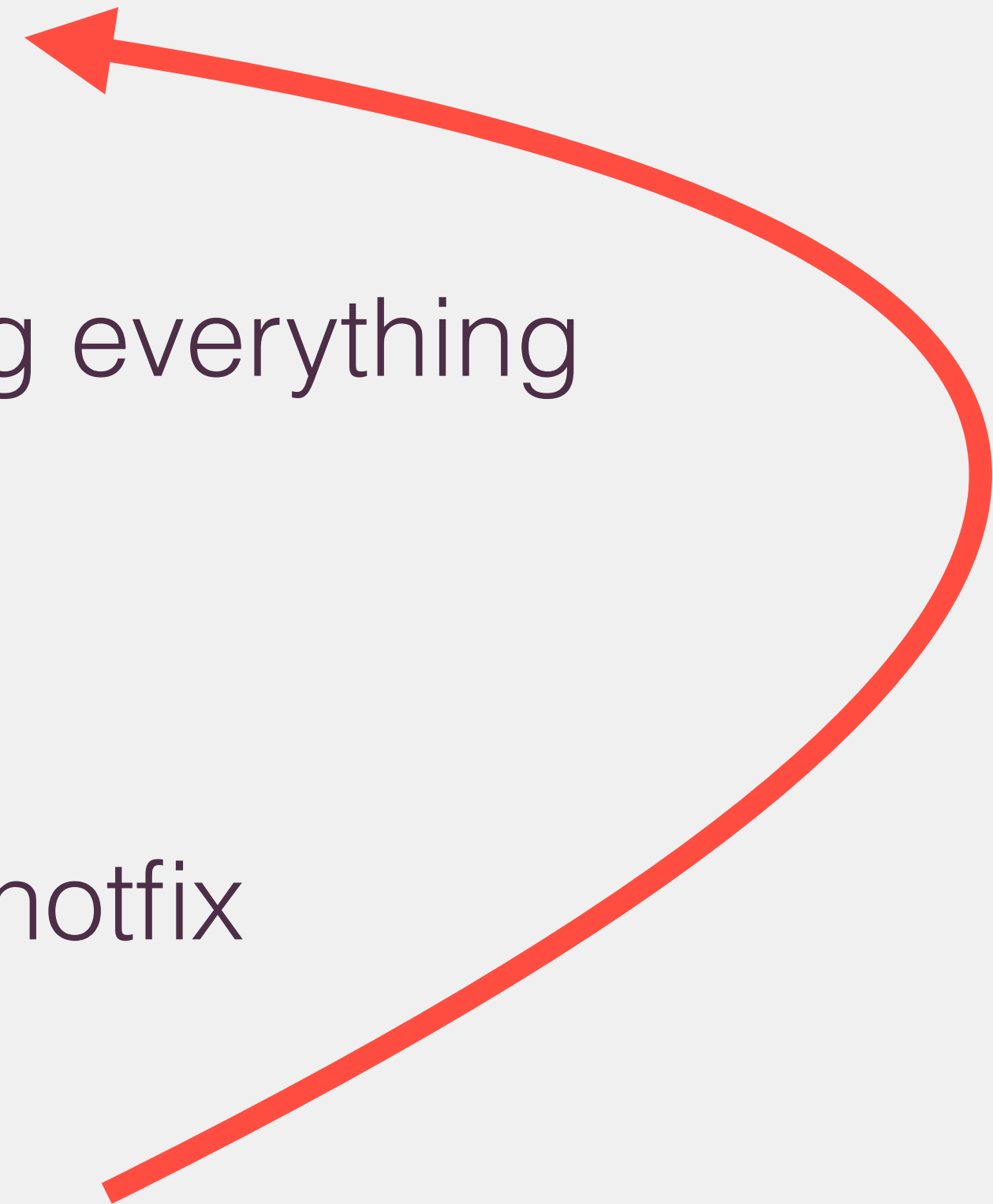
Ignore the problem



Breaks in production, pushes hotfix



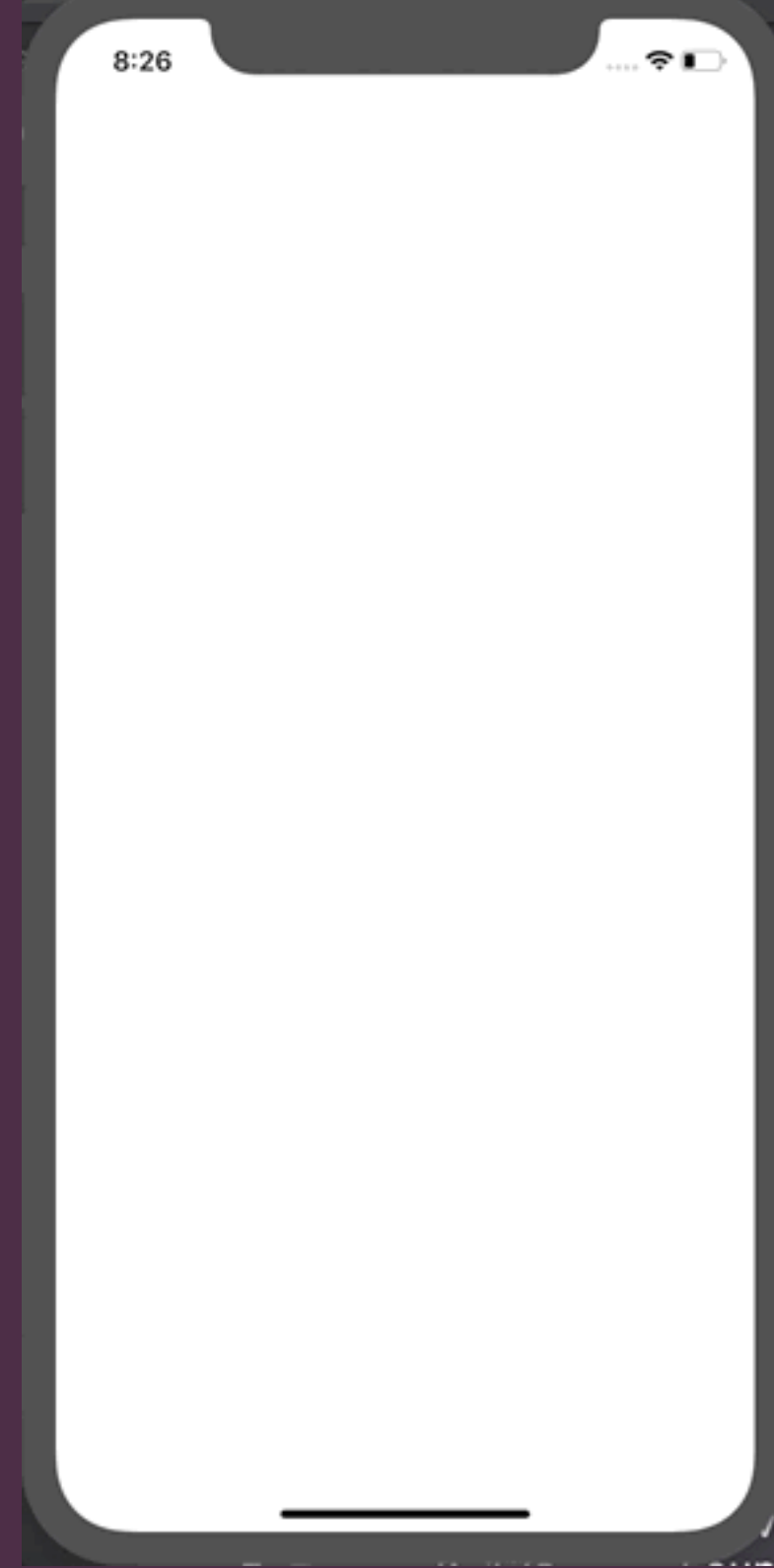
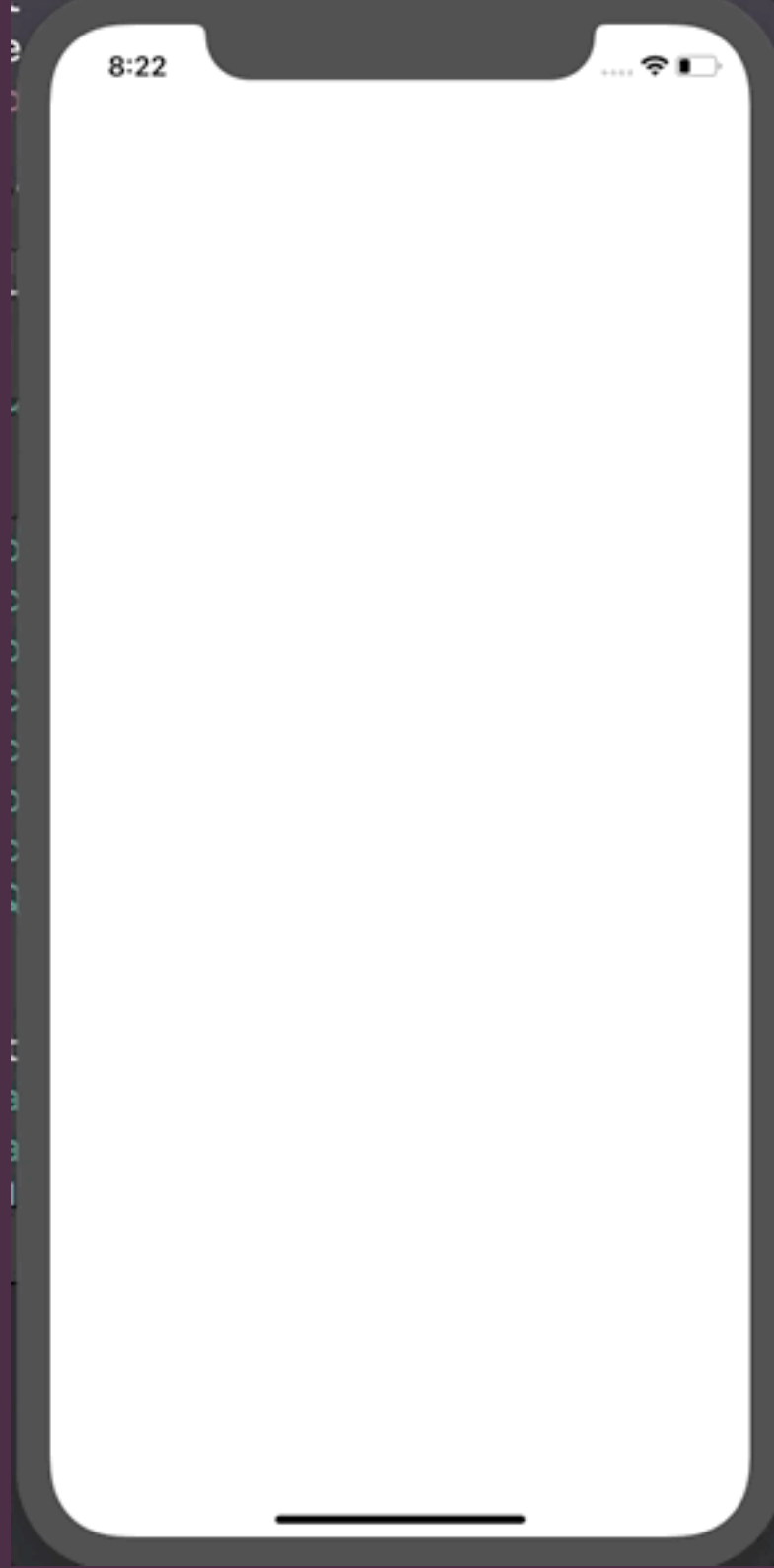
Codebase gets worse





Avoiding Release Anxiety

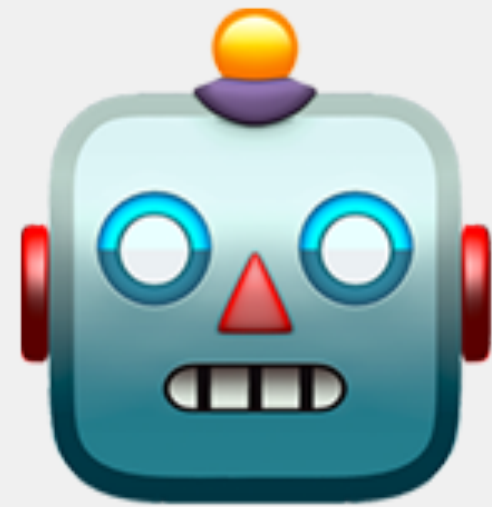
Testing efficiently





Rapiddo's Releases

Covering critical flows with UI Tests



~1 critical hotfix every 2 weeks



2 critical hotfixes in **2 years**

Testing efficiently

Block-based UI Testing


```
func testRechargeFlow() {
    _ = app.tables.staticTexts["Recarga"].waitForExistence(timeout: 0.2)
    app.tables.staticTexts["Recarga"].tap()
    let phoneField = app.textFields["(XX) XXXXX-XXXX"]
    phoneField.tap()
    phoneField.clearAndTypeText("123")
    let rechargeButton = app.buttons["Salvar e recarregar meu celular"]
    expect(rechargeButton.isEnabled) == false
    phoneField.clearAndTypeText("19999999999")
    expect(rechargeButton.isEnabled) == true
    closeKeyboard()
    expect(self.app.navigationBar["Escolha o valor da recarga"].exists) == true
    let productCell = app.collectionViews.cells["R$1"]
    let secondProductCell = app.collectionViews.cells["R$2"]
    let choosePaymentMethodButton = app.buttons["Escolher forma de pagamento"]
    expect(productCell.isSelected) == false
    expect(secondProductCell.isSelected) == false
    expect(choosePaymentMethodButton.isEnabled) == false
    productCell.tap()
    expect(productCell.isSelected) == true
    expect(secondProductCell.isSelected) == false
    expect(choosePaymentMethodButton.isEnabled) == true
    secondProductCell.tap()
    expect(productCell.isSelected) == false
    expect(secondProductCell.isSelected) == true
    expect(choosePaymentMethodButton.isEnabled) == true
    choosePaymentMethodButton.tap()
    expect(self.app.navigationBar["Concluir compra"].exists) == true
}
```

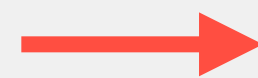
```
func exploreProfileScreen() {
  app.buttons["PROFILE"].tap()
  expectToBe(atScreen: "PROFILE_VIEW")
  // Test the profile screen
  app.tapBackButton()
}
```



```
func testHomeTabs() {
  app.launch()
  exploreProfile()
  exploreRestaurantL
}
```

```
func testHomeTabsWithNoAccount() {
  app.add(argument: User.firstLaunch)
  app.launch()
  exploreOnboarding()
  exploreAccountCreation()
  testHomeTabs()
}
```

```
func exploreRestaurantList() {
  app.buttons["RESTAURANTS"].tap()
  expectToBe(atScreen: "RESTAURANTS_VIEW")
  // Test the restaurants list
  exploreRestaurantMenu()
  app.tapBackButton()
}
```



```
func exploreRestaurantMenu() {
  app.cells["MOCK_RESTAURANT_1"].tap()
  expectToBe(atScreen: "RESTAURANT_MENU_VIEW")
  // Test the restaurant menu
  exploreDishDetails()
  app.tapBackButton()
}
```

```
public enum MockFlags {  
    public enum Environment: MockFlagsConditionable {  
        case isUITest
```

```
extension XCUIApplication {  
    func add(argument flag: MockFlagsConditionable) {  
        self.launchArguments.append(featureFlag.rawValue)  
    }  
}
```

```
extension MockFlagsConditionable {  
    fileprivate var condition: Bool {  
        return CommandLine.arguments.contains(self.rawValue)  
    }  
}
```

```
public func isFlagEnabled(_ flag: MockFlagsConditionable) -> Bool {  
    return flag.condition  
}
```

Testing efficiently

Typesafe Mocks and Protocols



```
protocol Mockable {
    associatedtype MockValue
    var mockedValue: MockValue { get }
}

extension HTTPRequest where Self: Mockable {
    typealias MockValue = Response
}

extension UserBalanceRequest: Mockable {
    var mockedValue: UserBalanceResponse {
        return UserBalanceResponse(balance: hasArgument(User.noBalance) ? 0 : 1_000_000)
    }
}
```



```
protocol HTTPClient: AnyObject {  
    func send<R: HTTPRequest>(_ resource: R) -> Promise<R.Value>  
}
```



```
final class URLSessionHTTPClient: HTTPClient {  
    func send<R: HTTPRequest>(_ resource: R) -> Promise<R.Value> {  
        // Perform a real request  
    }  
}
```



```
final class MockClient: HTTPClient {  
    func send<R: HTTPRequest>(_ request: R) -> Promise<R.Response> {  
        guard let value = (request as? Mockable)?.mockedValue else {  
            Logger.log("Request \(request) has no mocked version!")  
            return Promise(error: HTTPError.generic)  
        }  
        return Promise(value: value)  
    }  
}
```



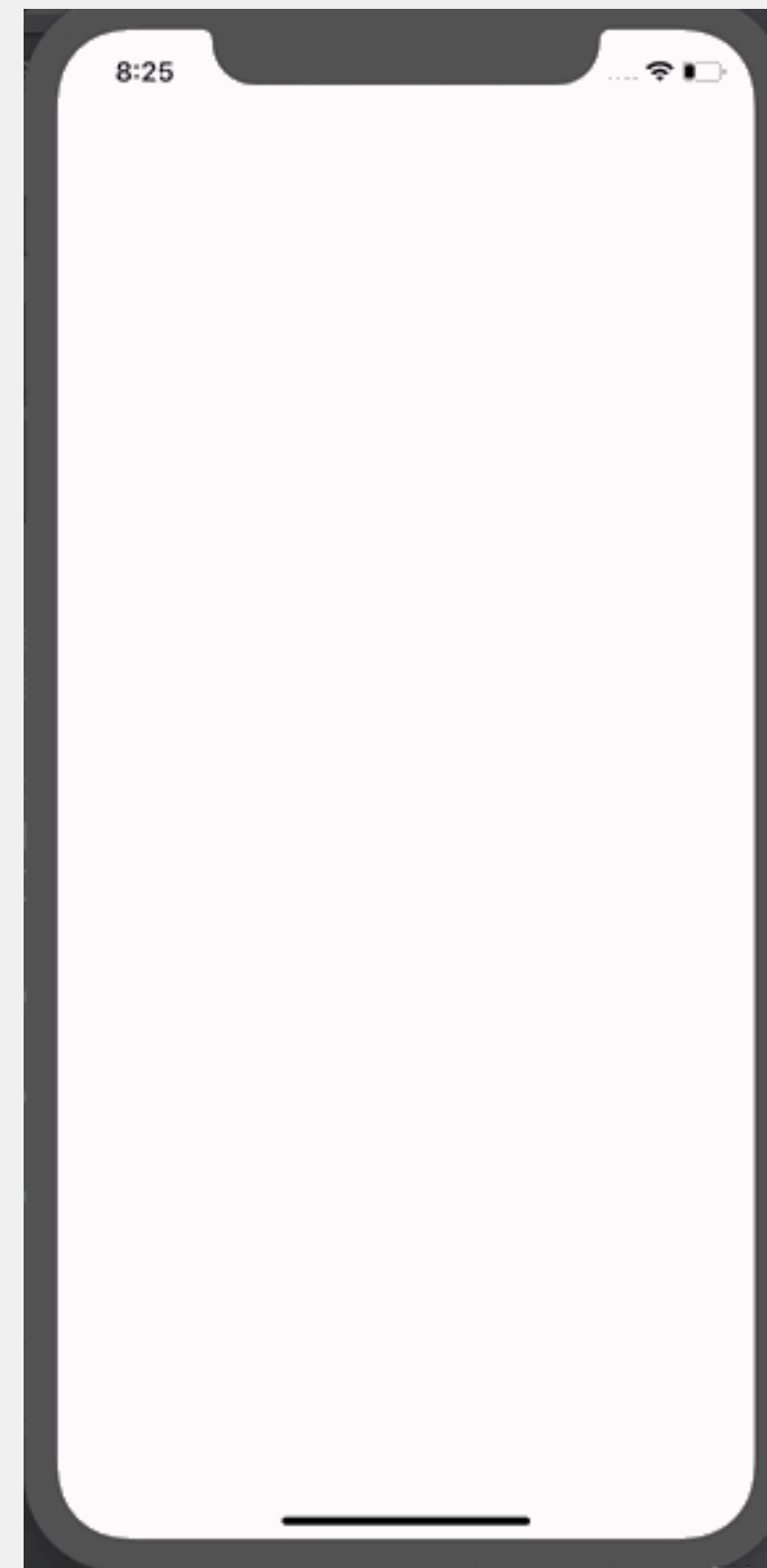

```
enum HTTPClientFactory {
    static func iFood() -> HTTPClient {
        if Environment.current == .mock {
            return MockClient()
        }
        let configuration = URLSessionConfiguration.default
        configuration.timeoutIntervalForResource = 15
        let session = URLSession(configuration: configuration)
        let client = URLSessionHTTPClient(baseUrl: Environment.current.baseUrl,
                                           session: session)
        client.set(defaultHeaders: ["Cache-Control": "no-cache"])
        return client
    }
}
```



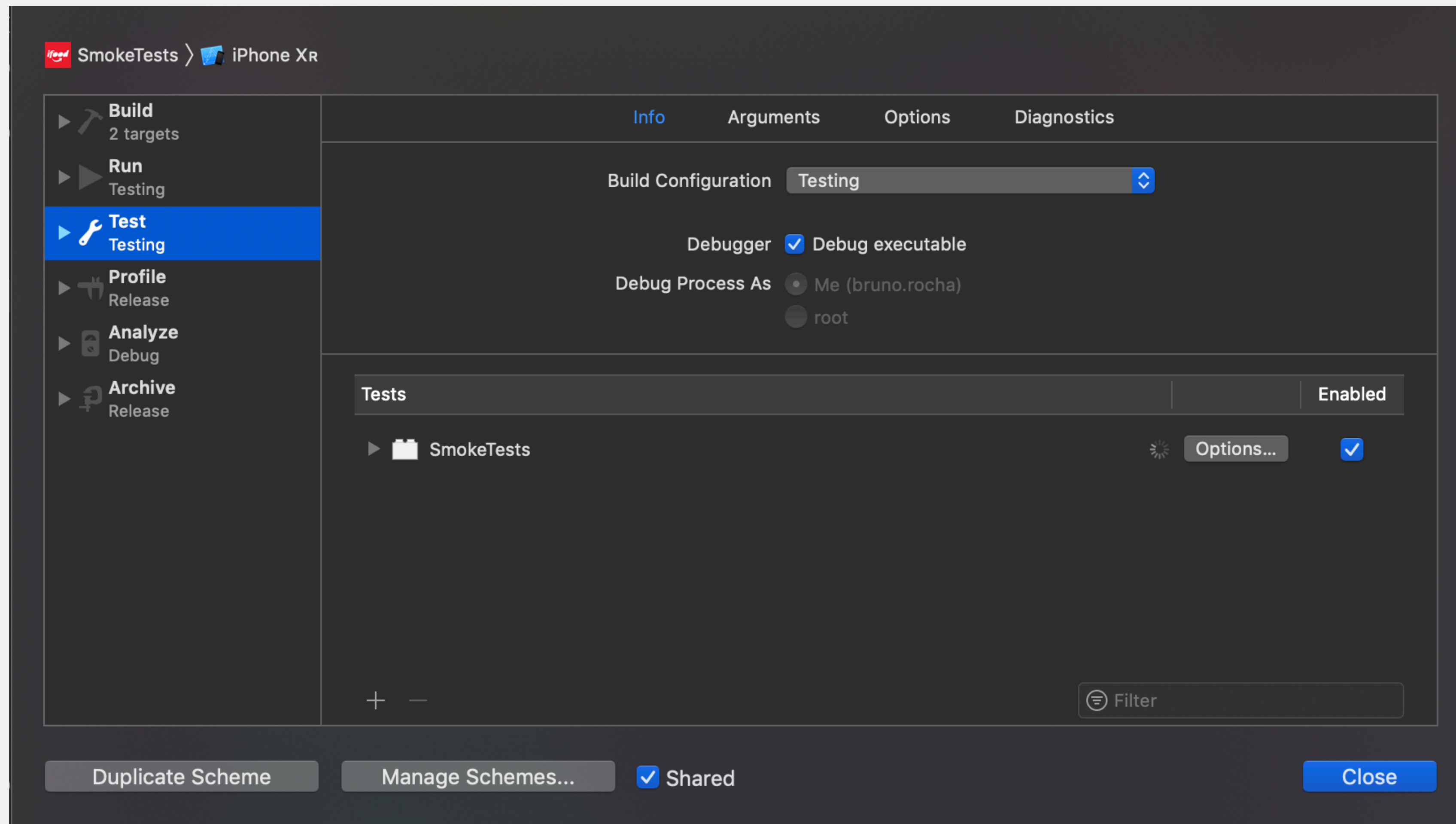
```
@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    fileprivate let client = HTTPClientFactory.iFood()
    fileprivate let persistence = PersistenceFactory.create()
    fileprivate let geolocationProvider = GeolocationProviderFactory.create()

    // ...
}
```



Mocks **!=** Real APIs



Smoke Tests

UI Tests that run on real APIs, triggered by backend changes.

UI Testing Pro

- CI works == User's app works

Avoiding Release Anxiety

Reduce margin of error



Xcode.xcodeproj

A high-performance build tool

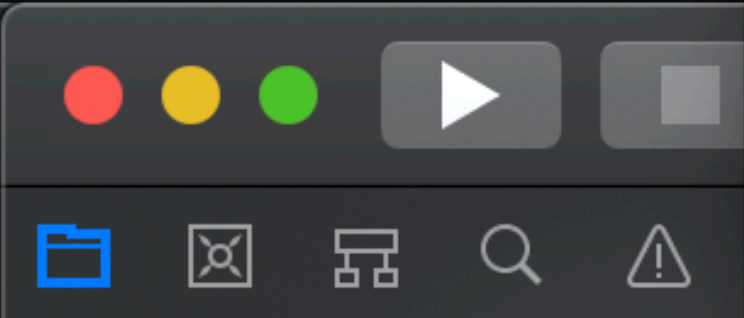
[GETTING STARTED](#)

[GITHUB](#)

```
$ buck build buck
[-] PARSING BUILD FILES...FINISHED 0.6s
[+] BUILDING...9.7s
|=> IDLE
|=> IDLE
|=> //src/com/facebook/buck/rules:rules... 0.4s (running javac[0.4s])
|=> IDLE
|=> IDLE
|=> IDLE
|=> IDLE
|=> IDLE
|=> IDLE
|=> IDLE
|=> IDLE
```

Buck is a build system developed and used by Facebook. It encourages the creation of small, reusable modules consisting of code and resources, and supports a variety of languages on many platforms.


```
apple_binary(  
    name = "iFoodBinary",  
    visibility = [  
        "//iFood:",  
        "//iFood/...",  
    ],  
    configs = app_binary_configs("iFood"),  
    swift_version = SWIFT_VERSION,  
    srcs = [  
        "BuckSupportFiles/Dummy.swift",  
    ],  
    deps = [  
        ":iFoodLibrary",  
    ],  
)
```



- Libraries
 - StaticFrameworks
 - uAnalytics
 - uKitchen
 - uLogin
 - uLoop
 - uPersistency
 - uRemoteConfig
 - uUI
- Pods
- PodsLocal
- iFood
 - Configurations
 - Frameworks
 - IntentsExtension#i..
 - IntentsExtensionUI..
 - IntentsKit
 - NotificationServic...
 - Other
 - Products
 - iFood
 - iFoodApp
 - iFoodAppLibrary
 - iFoodObjCUtils
 - iFoodTests

- AnalyticsApp
 - ✓ iFood (iFood project)
 - iFood (iFood Workspace)
 - iFood+IntentsExtension
 - iFood+IntentsExtensionUI
 - iFood+NotificationService
 - KitchenApp
 - LoopApp
 - PersistencyApp
 - Pods
 - PodsLocal
 - RemoteConfigApp
 - StaticFrameworks
 - uAnalytics
 - UIApp
 - uKitchen
 - uLogin
 - uLoop
 - uPersistency
 - uRemoteConfig
 - uUI
-
- Edit Scheme...
 - New Scheme...
 - Manage Schemes...



Buck is not for beginners!



code style prettier **PASSED** codecov 91% slack 11/27

What's Tuist

Tuist is a command line tool that helps you **generate, maintain and interact** with Xcode projects.

It's open source and written in Swift.

Xcake


gem v0.9.1 contact @igormaka license MIT build passing  test coverage 87%  Plugin Available

merge Xcode project file is so easy <https://fclef.wordpress.com>

python xcode xcode-project-file merge-conflicts

120 commits 3 branches 11 releases 3 contributors View license

Branch: master New pull request Create new file Upload files Find File Clone or download

 truebit	add note for cocoapods users whose project file was converted to XML ...	Latest commit cd8282e on Apr 7, 2017
.gitignore	add more compiled files	5 years ago
LICENSE	add README and LICENSE	5 years ago
README.rst	add note for cocoapods users whose project file was converted to XML ...	2 years ago
setup.py	bump up version	3 years ago
xUnique.py	fix a missing string type convert after applying #28	3 years ago

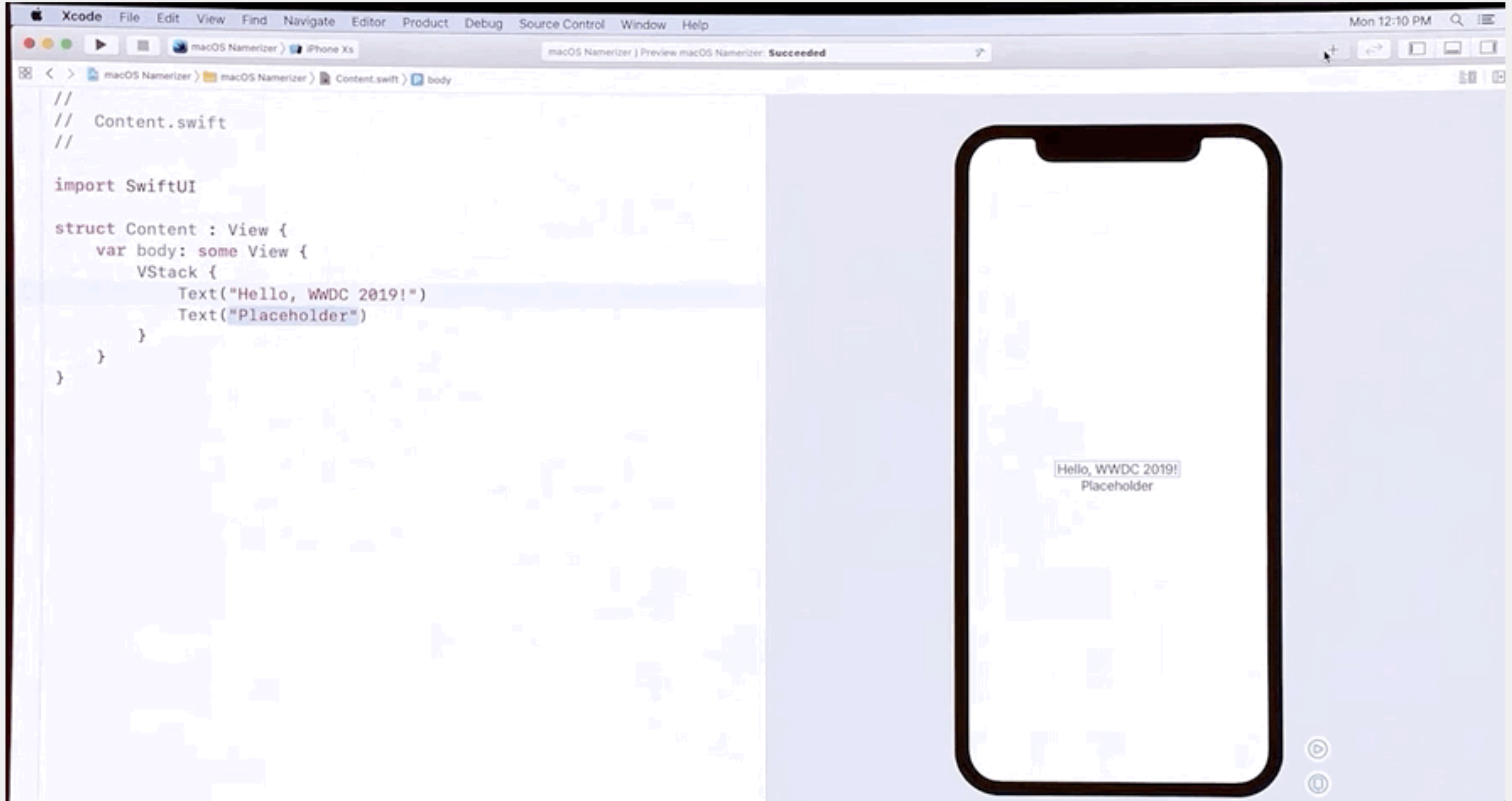


Main.storyboard

2 KB

```
<tableViewController id="Bef-YD-UQ0" customClass="ABTestViewController" customModule="iFood" customModuleProvider="target" sceneMemberI
  <tableView key="view" clipsSubviews="YES" contentMode="scaleToFill" alwaysBounceVertical="YES" dataMode="prototypes" style="plain"
    <rect key="frame" x="0.0" y="0.0" width="375" height="667"/>
    <autoresizingMask key="autoresizingMask" widthSizable="YES" heightSizable="YES"/>
    <color key="backgroundColor" white="1" alpha="1" colorSpace="custom" customColorSpace="genericGamma22GrayColorSpace"/>
    <prototypes>
      <tableViewCell clipsSubviews="YES" contentMode="scaleToFill" preservesSuperviewLayoutMargins="YES" selectionStyle="default"
        <rect key="frame" x="0.0" y="28" width="375" height="80"/>
        <autoresizingMask key="autoresizingMask"/>
        <tableViewCellContentView key="contentView" opaque="NO" clipsSubviews="YES" multipleTouchEnabled="YES" contentMode="cen
          <rect key="frame" x="0.0" y="0.0" width="375" height="79.5"/>
          <autoresizingMask key="autoresizingMask"/>
          <subviews>
            <label opaque="NO" userInteractionEnabled="NO" contentMode="left" horizontalHuggingPriority="251" verticalHuggi
              <rect key="frame" x="20" y="18" width="265" height="20"/>
              <fontDescription key="fontDescription" name="SulSans-Medium" family="Sul Sans" pointSize="16"/>
              <color key="textColor" red="0.2470588235" green="0.24313725489999999" blue="0.24313725489999999" alpha="1"
                <nil key="highlightedColor"/>
            </label>
            <label opaque="NO" userInteractionEnabled="NO" contentMode="left" horizontalHuggingPriority="251" verticalHuggi
              <rect key="frame" x="20" y="43.5" width="265" height="18"/>
              <fontDescription key="fontDescription" name="Menlo-Regular" family="Menlo" pointSize="15"/>
              <nil key="textColor"/>
              <nil key="highlightedColor"/>
            </label>
            <label opaque="NO" userInteractionEnabled="NO" contentMode="left" horizontalHuggingPriority="252" horizontalCor
              <rect key="frame" x="292.5" y="30" width="62.5" height="20"/>
              <color key="backgroundColor" red="1" green="1" blue="1" alpha="1" colorSpace="calibratedRGB"/>
              <constraints>
                <constraint firstAttribute="height" constant="20" id="hjR-CZ-bAk"/>
              </constraints>
              <fontDescription key="fontDescription" name="SulSans-Bold" family="Sul Sans" pointSize="9"/>
              <color key="textColor" red="0.65098039220000004" green="0.63529411759999999" blue="0.62352941179999999" alp
                <nil key="highlightedColor"/>
              <userDefinedRuntimeAttributes>
                <userDefinedRuntimeAttribute type="number" keyPath="cornerRadius">
                  <integer key="value" value="3"/>
                </userDefinedRuntimeAttribute>
                <userDefinedRuntimeAttribute type="color" keyPath="borderColor">
                  <color key="value" red="0.65098039220000004" green="0.63529411759999999" blue="0.62352941179999999" alp
```

```
4
5 private func setup() {
6     backgroundColor = Style.Colors.snowGray
7     setupTableView()
8     setupEmptyStateView()
9     setupLoadingView()
10 }
11
12 private func setupTableView() {
13     addSubview(tableView)
14     constrain(tableView, self) { view, superview in
15         view.edges == superview.edges
16     }
17 }
18
19 private func setupEmptyStateView() {
20     addSubview(emptyStateView)
21     constrain(emptyStateView, self) { view, superview in
22         view.edges == superview.edges
23     }
24 }
25
26 private func setupLoadingView() {
27     addSubview/loadingView)
28     constrain/loadingView, self) { view, superview in
29         view.edges == superview.edges
30     }
31 }
```

SwiftGen

 PASSED pod v6.1.0 platform osx | ios | tvos | watchos

SwiftGen is a tool to auto-generate Swift code for resources of your project:



- [Installation](#)
- [Usage](#)
- [Choosing your template](#)
- [Additional documentation](#)

Then generate constants for:

- [Assets Catalogs](#)
- [Colors](#)
- [Core Data](#)
- [Fonts](#)
- [Interface Builder files](#)
- [JSON and YAML files](#)
- [Plists](#)
- [Localizable strings](#)



```
Localization.loginHeader // "Login"  
Asset.loginBanner.image // UIImage
```

```
add_badge(dark: true)  
increment_build_number(build_number: number_of_commits)  
increment_version_number(version_number: enterprise_version_number)
```



Xcode.xcodeproj



Main.storyboard

2 KB

```
Localization.loginHeader // "Login"  
Asset.loginBanner.image // UIImage
```

```
add_badge(dark: true)  
increment_build_number(build_number: number_of_commits)  
increment_version_number(version_number: enterprise_version_number)
```

Reduce margin of error

CI/CD Damage Control



All checks have passed

5 successful checks

[Hide all checks](#)



ci/bitrise/f8cd86a115fcad49/pr — Passed - ifood-consumer-ios

Required

[Details](#)



codecov/changes — No unexpected coverage changes found.

[Details](#)



codecov/patch — Coverage not affected when comparing 0fe13cb...e122bc0

[Details](#)



codecov/project — 32.12% (+0.01%) compared to 0fe13cb

[Details](#)



danger/danger — All green. Woo!

CI Pipeline for PRs

1. Clone
2. Install Dependencies
3. make ci
4. — bundle exec danger
5. — gen_xcode
6. — test_components
7. — test_ifood
8. — test_ui
9. — run_ifood (just check if the app can be installed)

Danger

Danger runs during your CI process, and gives teams the chance to automate common code review chores.

This provides another logical step in your build, through this Danger can help lint your rote tasks in daily code review.

You can use Danger to codify your teams norms. Leaving humans to think about harder problems.

She does this by leaving messages inside your PRs based on rules that you create with the Ruby scripting language.

Over time, as rules are adhered to, the message is amended to reflect the current state of the code review.



DangerCI commented

1 Error	
	Please include a CHANGELOG entry. You can find it at CHANGELOG.md .
	Please provide a summary in the Pull Request description

1 Warning	
	The file dangerfile_import_plugin.rb does not pass <code>bundle exec danger plugins lint</code> . We want high coverage, as user documentation is auto-generated from it.

1 Message	
	@dangermcshane is not a member of the Danger organisation, would you like an invitation? It's optional, and is part of the Moya Community Continuity .

Generated by danger



rapiddo-bot commented 36 minutes ago



2 Warnings



Please consider adding a CHANGELOG entry. Mark your PR as [TRIVIAL] if your change isn't important.



Please consider modifying the project's unit/UI tests to reflect your changes. Mark your PR as [TRIVIAL] if your change isn't important.

5 Messages



Affects CoreArchitecture



Affects CoreUI



Affects Main



Unit Tests: Executed 43 tests, with 0 failures (0 unexpected) in 1.981 (2.037) seconds



UI Tests: Executed 6 tests, with 0 failures (0 unexpected) in 79.440 (79.452) seconds

SwiftLint found issues

Warnings

File	Line	Reason
RewardsDashboardPresenter.swift	19	TODOs should be resolved (- Styleguide).
RewardsDashboardViewController.swift	12	TODOs should be resolved (- Styleguide).

Generated by Danger

vtsbot c

master

2.3.1

- Addin
- Makir

2.3.0

- Adde

USER



USER

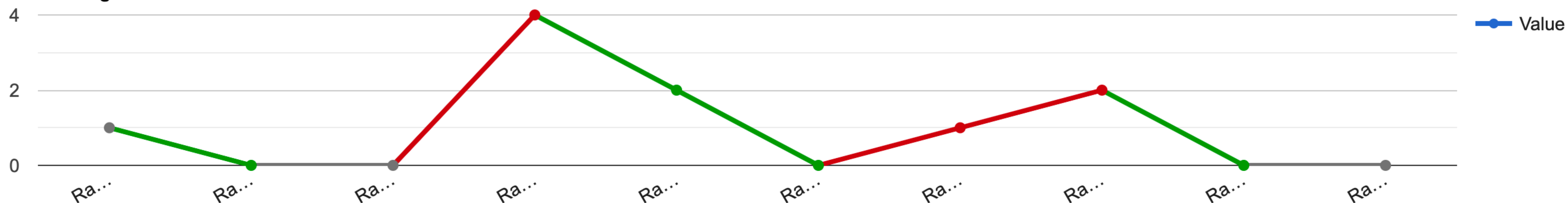


Botty McBotface APP 8:54 PM

SwiftInfo results for MyApp 1.10.11:

📦 app size grew by 1.86 MB (42.29 MB)

Warning Count



tag v2.3.1

SwiftInfo is a CLI tool that extracts, tracks and analyzes metrics that are useful for Swift apps. Besides the default tracking options that are shipped with the tool, you can also customize SwiftInfo to track pretty much anything that can be conveyed in a simple `.swift` script.

By default SwiftInfo will assume you're extracting info from a release build and send the final results to Slack, but it can be used to extract info from individual pull requests as well with the [danger-SwiftInfo danger](#) plugin.



rapiddo-bot commented 2 minutes ago



1 Message

SwiftInfo results for Rapiddo 2.0.3 (243) - Release:



 Time to Build and Run Tests: **Reduced** by 5.58 secs (67.068 secs)

 Test Count: **Increased** by 1 (96)

 Longest Test: -[RapiddoUITests.RechargeUITests testRechargeFlow] (7.263 secs)

Generated by  Danger



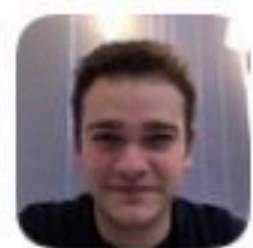
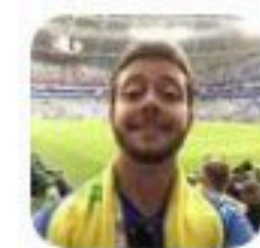
iFoodinho APP 10:21 PM

SwiftInfo results for iFood 8.51.0 (8.51.0.1) -
Release:

📦 .ipa size **grew** by 207.92 MB (607.67 MB)

⚠️ Warning count: Unchanged. (7)

👶 Dependency Count **grew** by 1 (27)



25 replies



Changes approved

2 approving reviews by reviewers with write access. [Learn more.](#)

[Show all reviewers](#)



2 approvals



All checks have passed

2 successful checks

[Hide all checks](#)



ci/bitrise/f8cd86a115fcad49/pr — Passed - ifood-consumer-ios

Required

[Details](#)



danger/danger — All green. Good on 'ya.

Required

[Details](#)



This branch has no conflicts with the base branch

Merging can be performed automatically.

Squash and merge

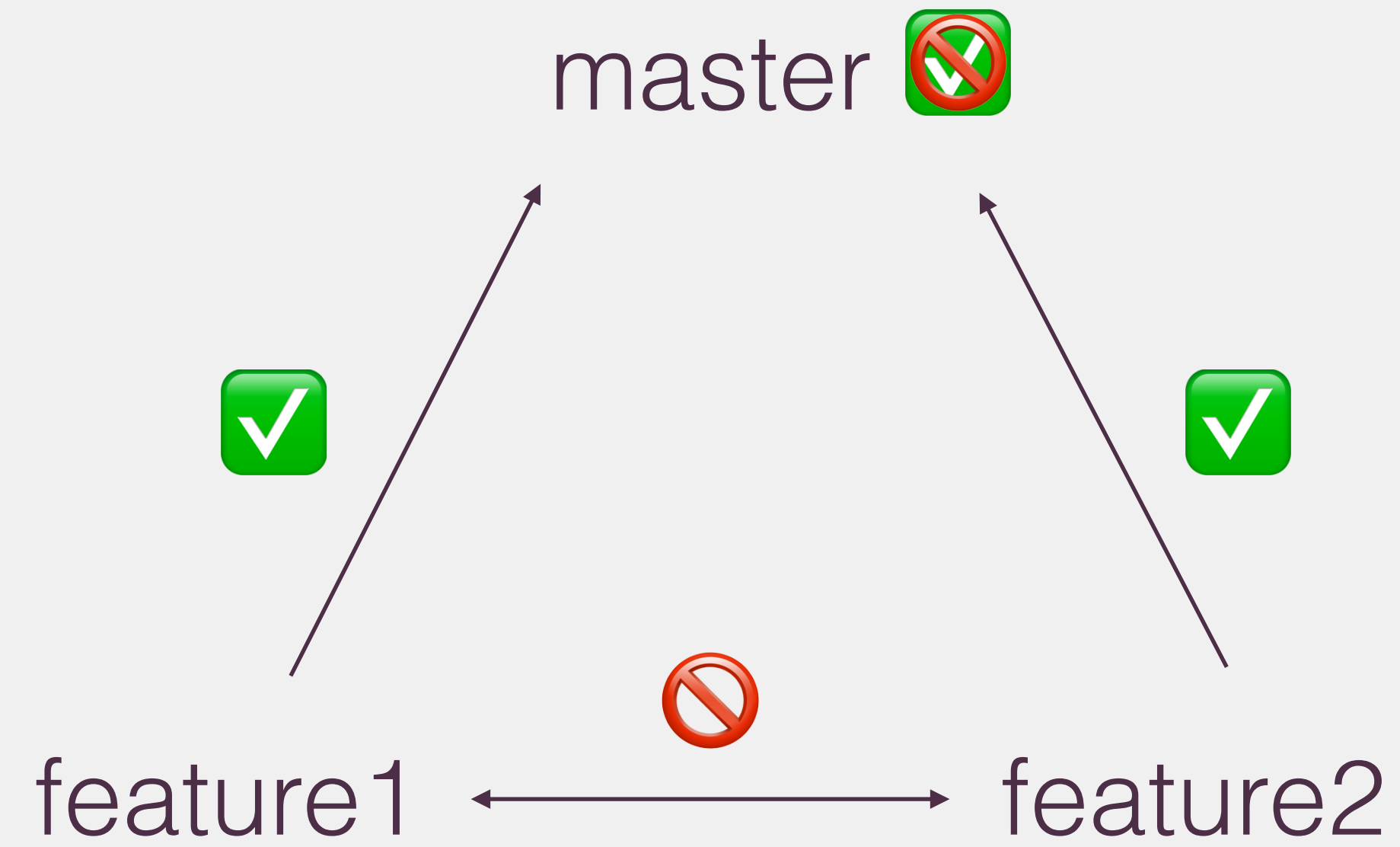


You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

No.



CI/CD Damage Control





rockbruno commented 5 days ago

Author

Owner



ifood-ci r+



ifoodbors bot added a commit that referenced this pull request 5 days ago



Merge #10 ...

✓ 6670afa



ifoodbors bot commented 5 days ago

Contributor



Build succeeded

- [ci/bitrise/93745483555ffe97/push](#)



ifoodbors bot merged commit **b02f11e** into **master** 5 days ago

1 check passed

[View details](#)

Avoiding Release Anxiety

Teams

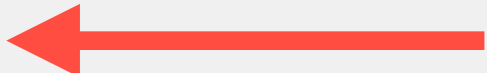
Communication

Teams

Code Review

Review exigency

Too much 🚫
(Toxic)



Too little 🚫
(Instant release anxiety)

Review Guidelines

1. Check if it follows the project's style
2. Check if it works (is **clearly** unit tested)
3. Don't be a dick

...es/SquadPromotions/Voucher/Domain/DataStore/Api/WalletApiDataStore.swi ✖ Hide resolved

ft Outdated

```
54 + let vouchers = self.mapVouchers(entities: voucherEntities
55 +
56 + guard let achievementEntities = apiEntity.achievements,
57 + let achievement = AchievementApiEntityMapper(entities
```



rockbruno 19 hours ago



Nitpick: Weird formatting in this line

Value your team's relationship over nitpicking non-critical details

Review Etiquette

```
func isUserSubscribed() -> Bool {  
    if let subscription = currentSubscription() {  
        return subscription.isActive  
    } else {  
        return false  
    }  
}
```

You should **guard** instead of **if** here.

We could use **guard** instead of **if** here.

Use guard **instead** of **if** here.

Review Etiquette

Questions instead of demands

Do **this**.

What do you think of **this**?

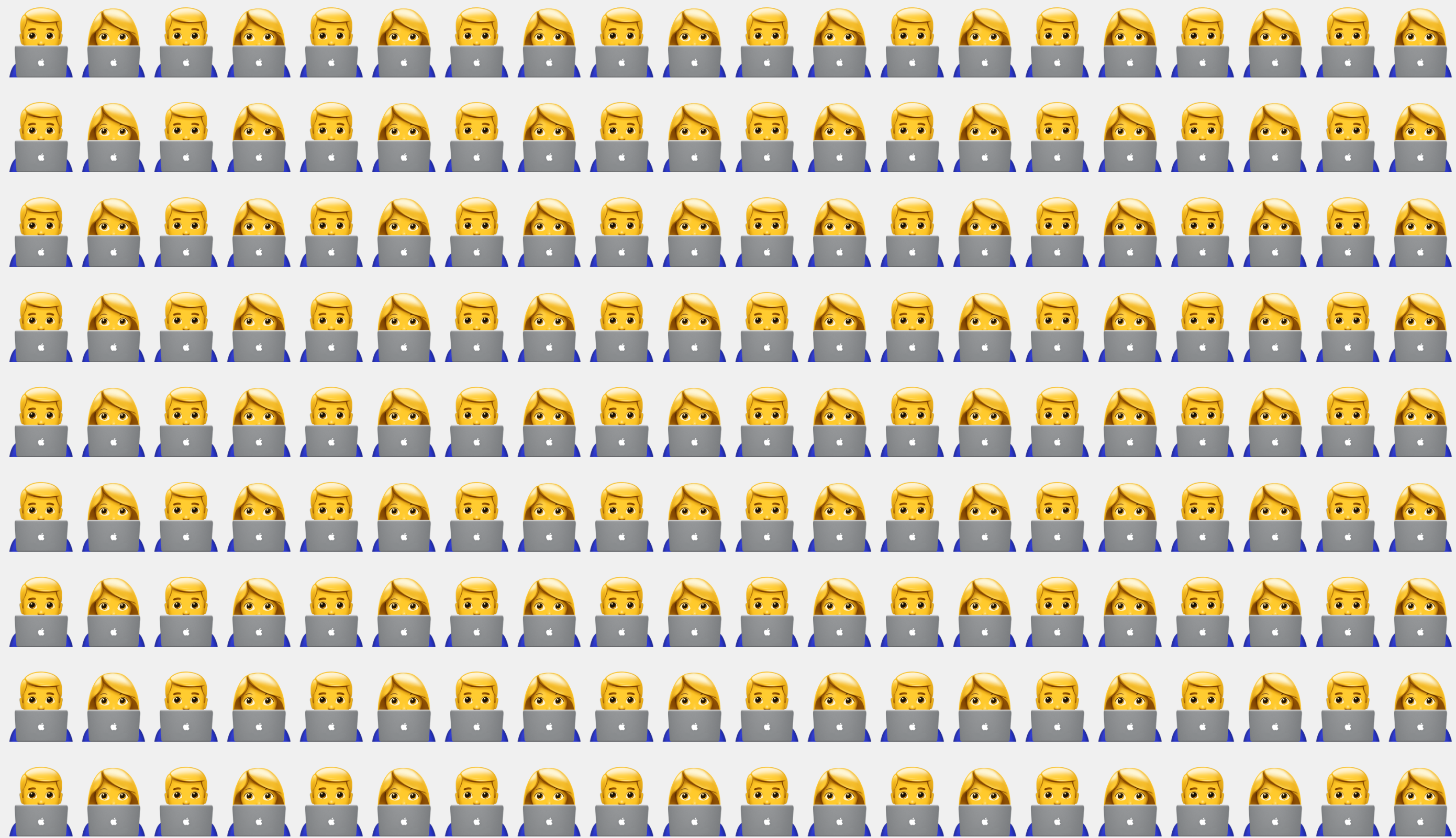
Review Etiquette: Being reviewed

1. Explain why the change was needed
2. Understand the reviewer's perspective
- 3. Don't merge if you're not confident it works**

What should be reviewed?

Teams

Style Guide



Not having a Style Guide can mean:

1. Every component is different
2. Changing squads has a learning curve
3. Code review becomes hard
4. "Legacy" code plagues the app



Style Guide Pros

1. All code feels it was written by you
2. Changing others' code is easy
3. Code review is easier/faster

499 lines (354 sloc) | 19.6 KB

Raw Blame History   

The MovilePay iOS Style Guide

If you are a MovilePay employee, you can find documentation that explains how the project itself works in the main repository's README.

Creating new screens in MovilePay

We use MVVM-C (Model View View Model with Coordinators) as our architecture with our own Coordinator implementation (which you can find documentation for in the main project's README)

Swift Style Guide

This style guide is based on Apple's excellent Swift standard library style and also incorporates feedback from usage across multiple Swift projects within Google. It is a living document and the basis upon which the formatter is implemented.

The Official raywenderlich.com Swift Style Guide.

Updated for Swift 4.2

This style guide is different from others you may see, because the focus is centered on readability for print and the web. We created this style guide to keep the code in our books, tutorials, and starter kits nice and consistent — even though we have many different authors working on the books.

Our overarching goals are clarity, consistency and brevity, in that order.

Style Guide Topics

Architecture

MyScreenCoordinator

In MVVM-C, The Coordinator is the object responsible for handling screen transitions. It retains its inner `UIViewController` and delegates it in order to know when to transition to another Coordinator.

```
import UIKit

final class MyScreenCoordinator: Coordinator {
    init(client: HTTPClient, persistence: Persistence) {
        let viewModel = MyScreenViewModel(client: client, persistence: persistence)
        let viewController = MyScreenViewController(viewModel: viewModel)
        super.init(rootViewController: viewController)
        viewController.delegate = self
    }
}

extension MyScreenCoordinator: MyScreenViewControllerDelegate {
    func continue() {
        let coordinator = NextCoordinator()
        push(coordinator, animated: true)
    }
}
```

If your screen needs to retain or pass a `MovilePayDelegate` delegate forward, you should use `MovilePayCoordintor` instead as it holds an unowned reference to the delegate.

Clean Code & Naming

Naming

Promote Clear Usage

- **Include all the words needed to avoid ambiguity** for a person reading code where the name is used.

▼ LESS DETAIL

For example, consider a method that removes the element at a given position within a collection.

```
extension List {  
  public mutating func remove(at position: Index) -> Element  
}  
employees.remove(at: x)
```



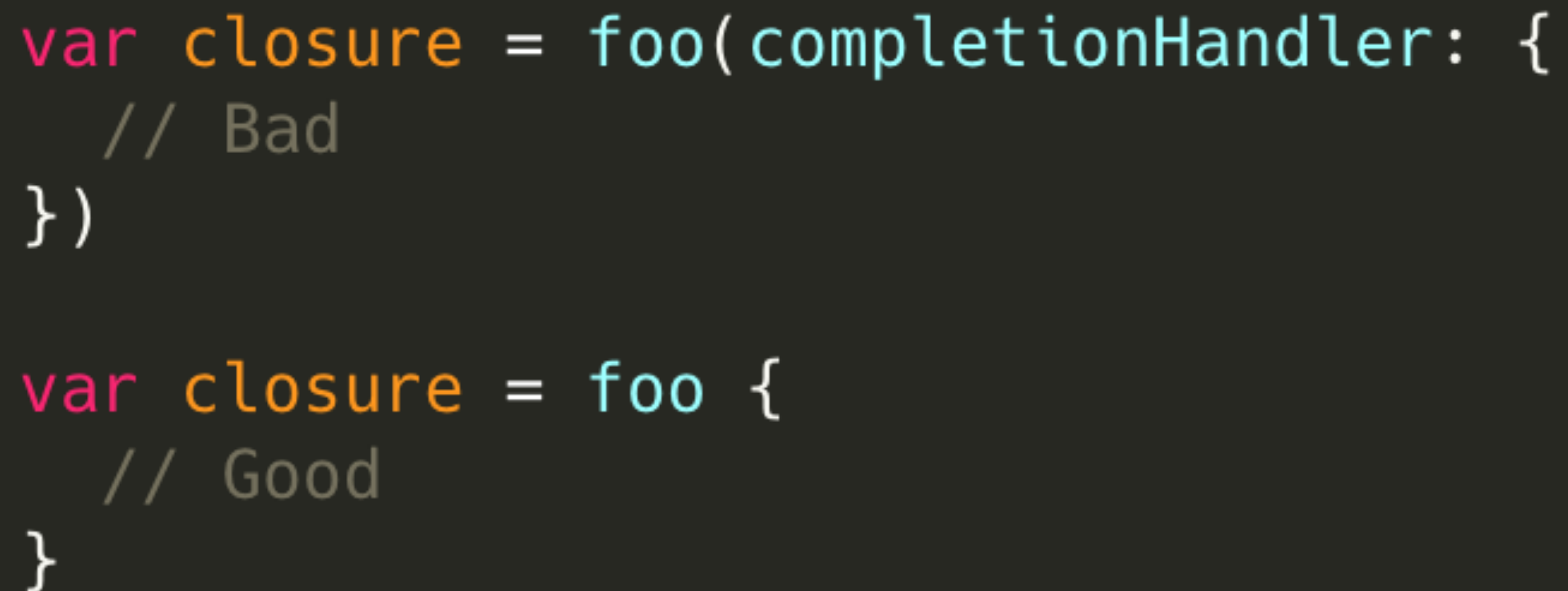
If we were to omit the word `at` from the method signature, it could imply to the reader that the method searches for and removes an element equal to `x`, rather than using `x` to indicate the position of the element to remove.

```
employees.remove(x) // unclear: are we removing x?
```




Style Guide Topics

Desired Usage of Compiler Syntax Sugar





```
var closure = foo(completionHandler: {  
    // Bad  
})  
  
var closure = foo {  
    // Good  
}
```

Style Guides are Incremental


_SDK/Sources/Screens/TransactionsList/TransactionsListViewModel.swift  Hide resolved

Outdated

```
128     127     }
129     128
130     129     private func updateActivityResult(with newItems: WalletActivityPa
131     -         guard let activityPage = transaction.activityPage else {
132     -             transaction.activityPage = newItems
130     +         guard !segments[currentCategoryIndex].data.isEmpty else {
```

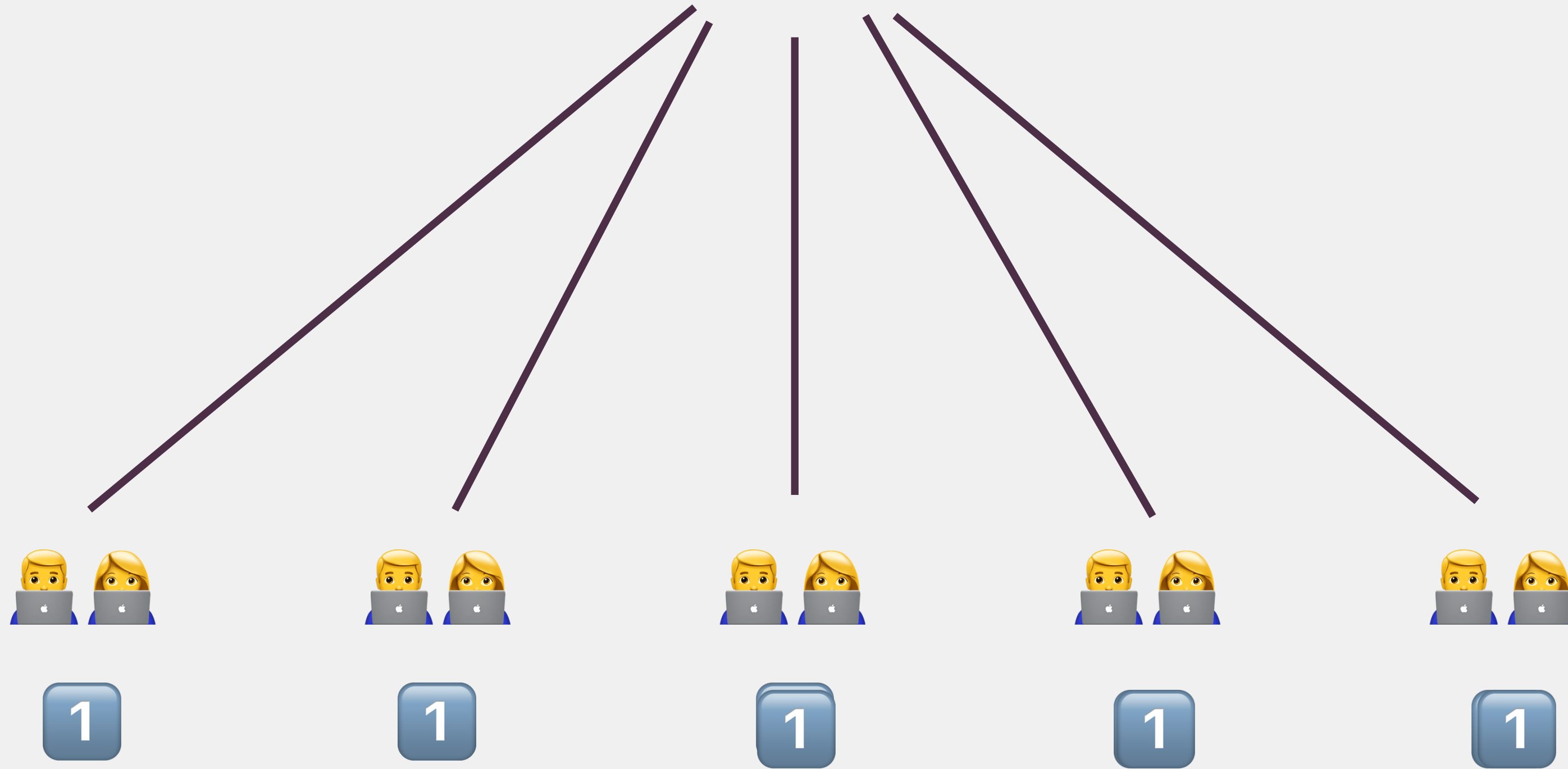
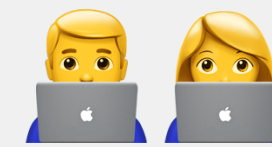
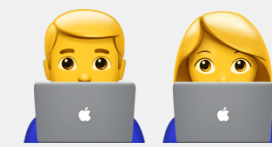
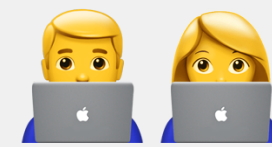
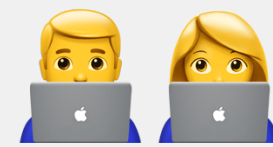
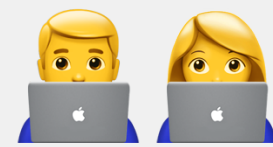
 **rockbruno** 12 hours ago +  ...

Style guide nitpick: I personally think `segments[currentCategoryIndex].data.isEmpty == false` is more readable than `!segments[currentCategoryIndex].data.isEmpty`. What do you guys think?

 1



App



“But I don’t have time!”

A high-performance build tool

[GETTING STARTED](#)

[GITHUB](#)

```
$ buck build buck
[-] PARSING BUILD FILES...FINISHED 0.6s
[+] BUILDING...9.7s
|=> IDLE
|=> IDLE
|=> //src/com/facebook/buck/rules:rules... 0.4s (running javac[0.4s])
|=> IDLE
|=> IDLE
|=> IDLE
|=> IDLE
|=> IDLE
|=> IDLE
|=> IDLE
|=> IDLE
```

Buck is a build system developed and used by Facebook. It encourages the creation of small, reusable modules consisting of code and resources, and supports a variety of languages on many platforms.

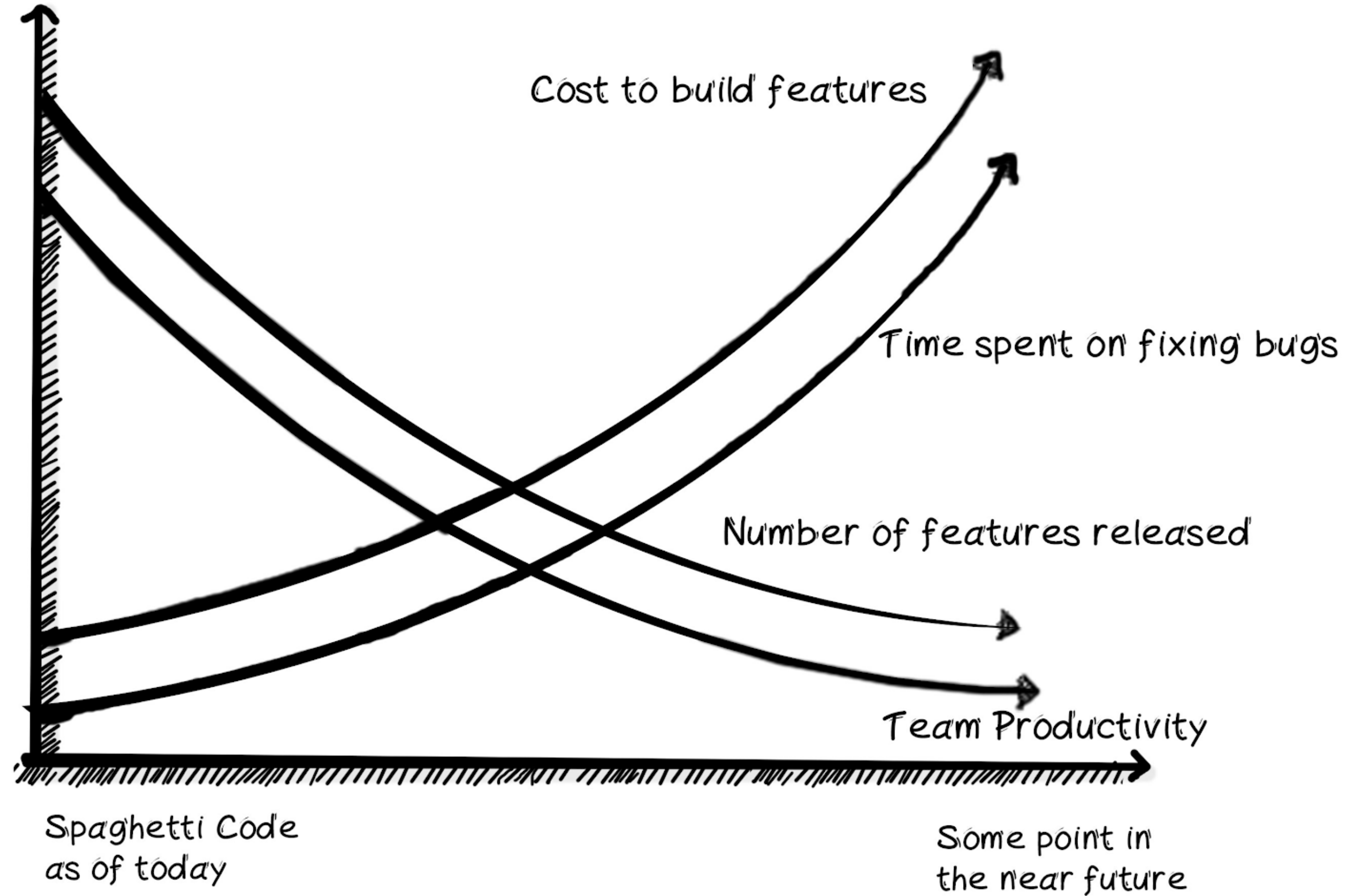
iFood with Xcode

1. CI taking over 20 minutes to run
2. Infinite xcodeproj conflicts
3. Xcode failing to build due to swiftc argument size
4. Multiple bugs in production as tests from the inner Monorepo modules aren't checked in CI

iFood with Buck

1. Local/CI builds can take only a few seconds to run thanks to Buck HTTP Cache
2. Creating new Monorepo modules is just a matter of creating the folders
3. Little to no conflicts thanks to Buck project generation
4. The main targets run the tests from all modules thanks to project generation

Reasons to Refactor



Your health



+



Obrigado!



[swiftrocks.com](https://www.swiftrocks.com)



[@rockthebruno](https://twitter.com/rockthebruno)