OZON tech

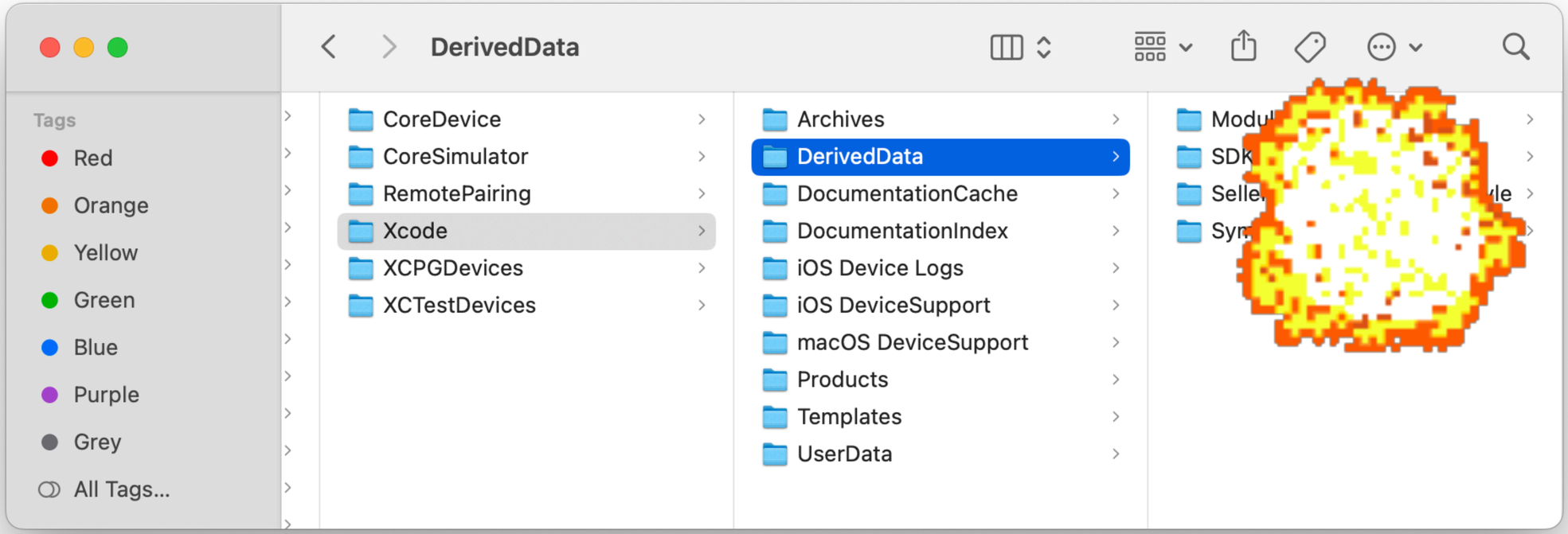
Звук кофемашины, или Кэшируем зависимости SPM

Евгений Рыжов, Старший iOS разработчик

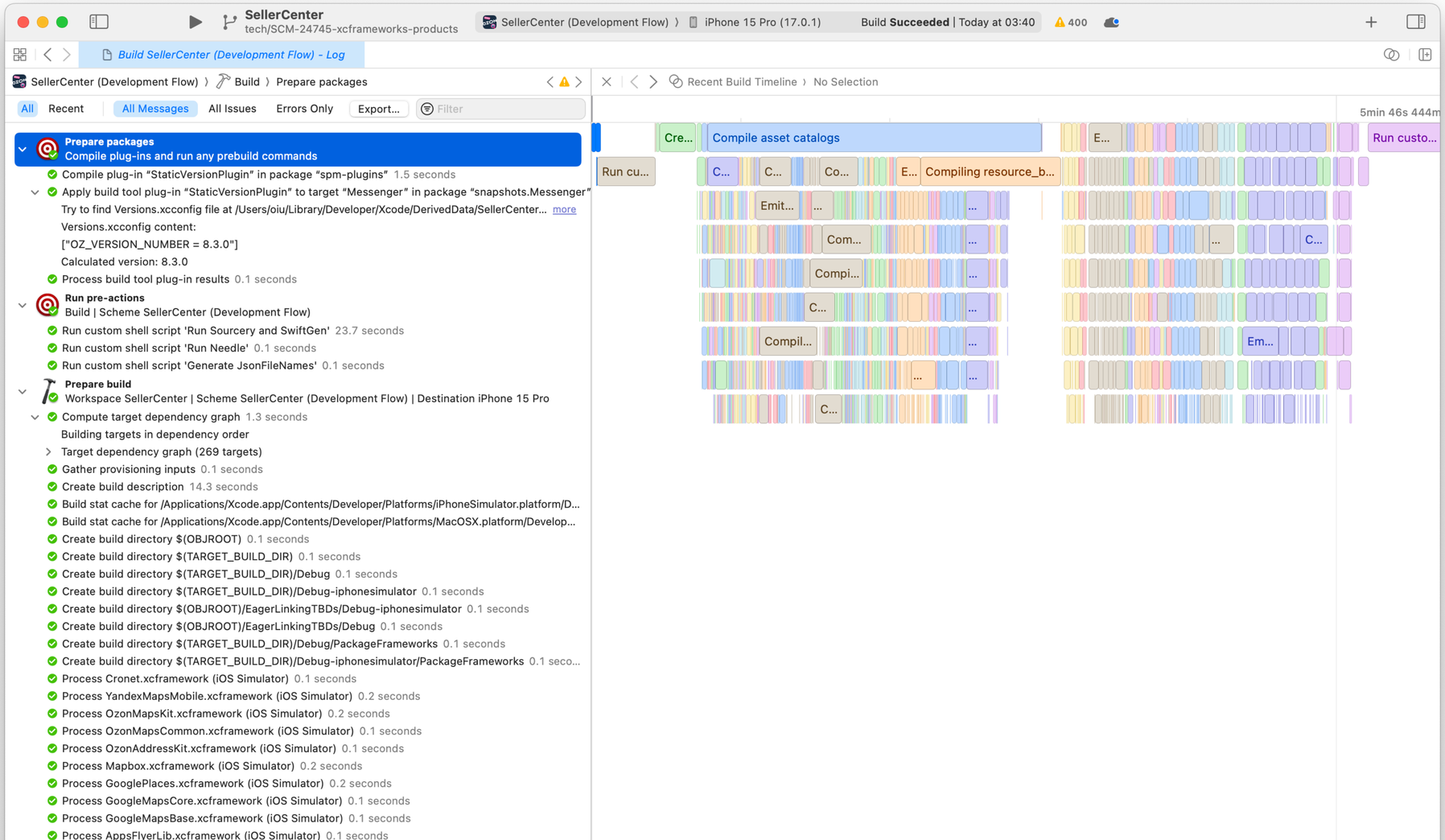
@oiuhr

Shift + ⌘command + K

Знакомо?

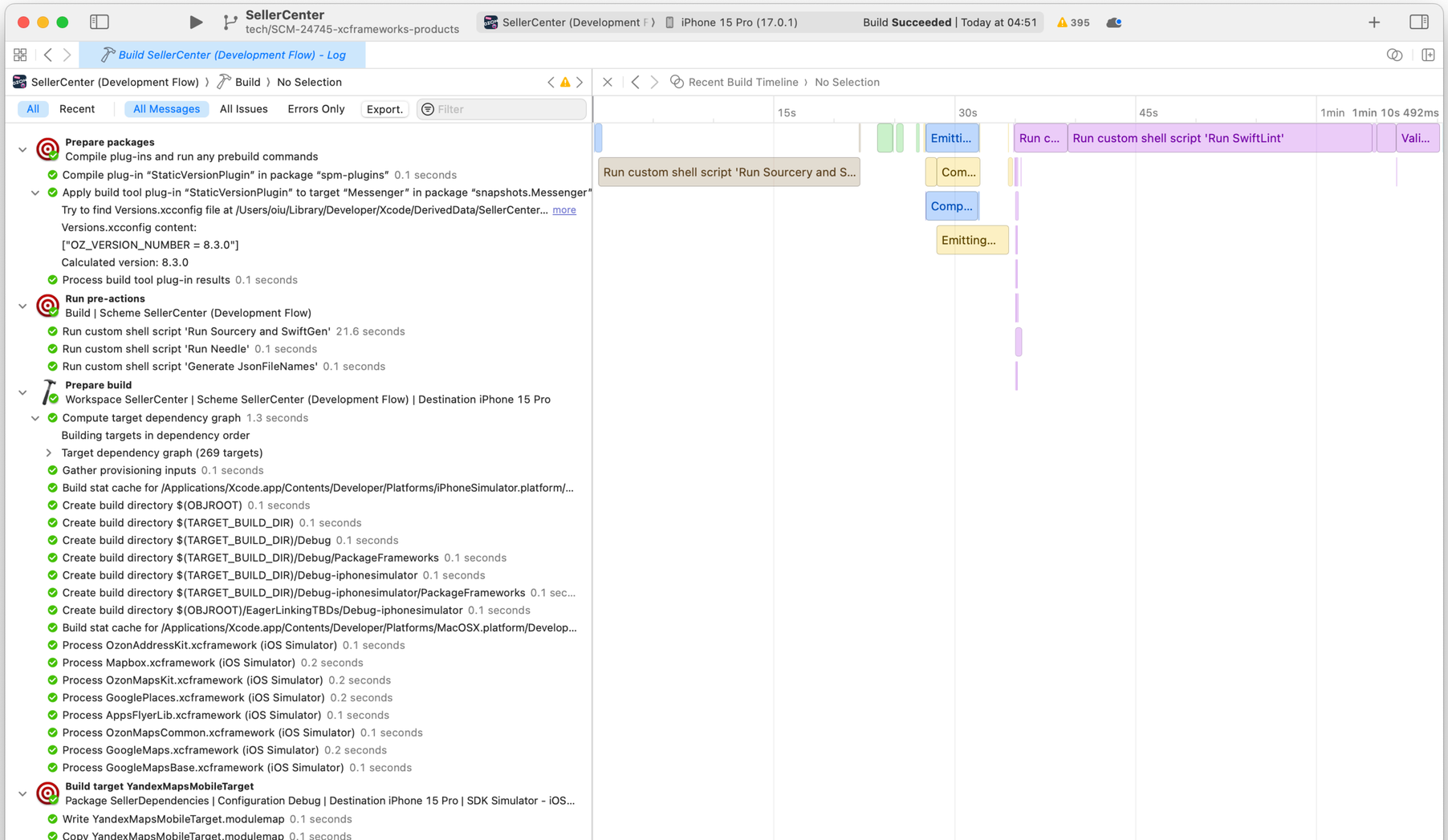

DerivedData
Tags
Red
Orange
Yellow
Green
Blue
Purple
Grey
All Tags...
CoreDevice
CoreSimulator
RemotePairing
Xcode
XCPGDevices
XCTestDevices
Archives
DerivedData
DocumentationCache
DocumentationIndex
iOS Device Logs
iOS DeviceSupport
macOS DeviceSupport
Products
Templates
UserData

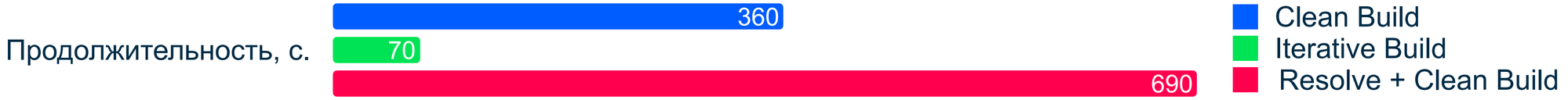
Продолжительность, с. 360 Clean Build



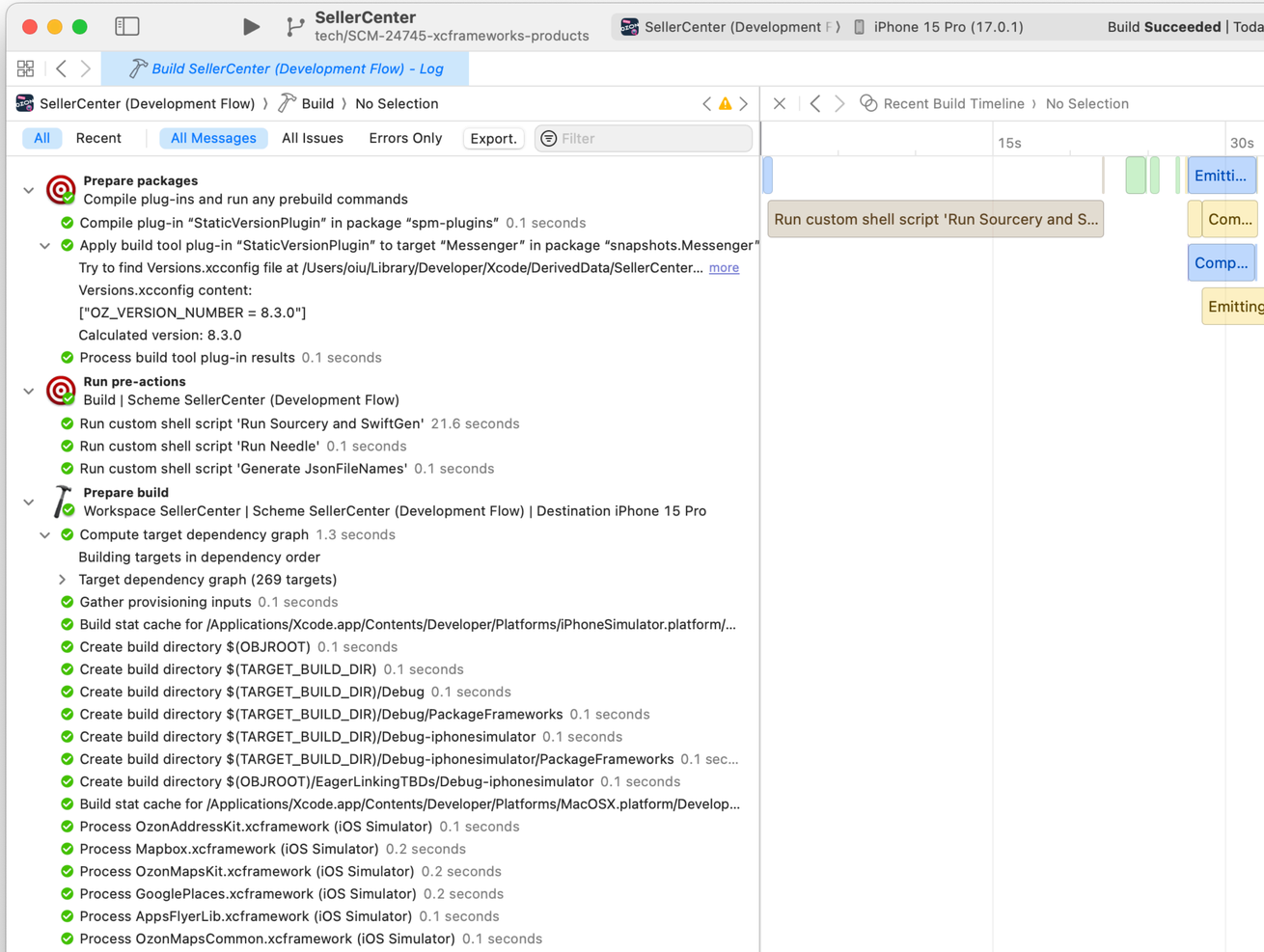
Продолжительность, с.

Clean Build
Iterative Build

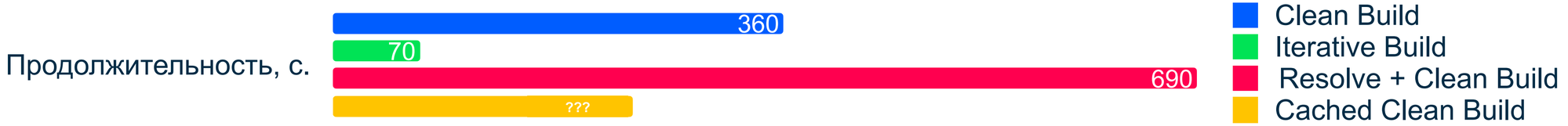


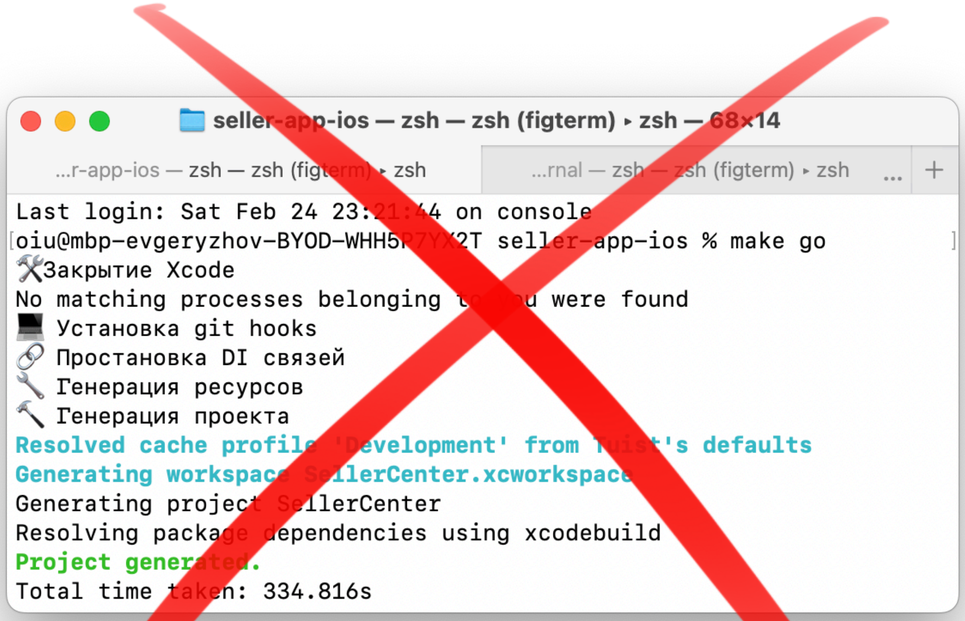


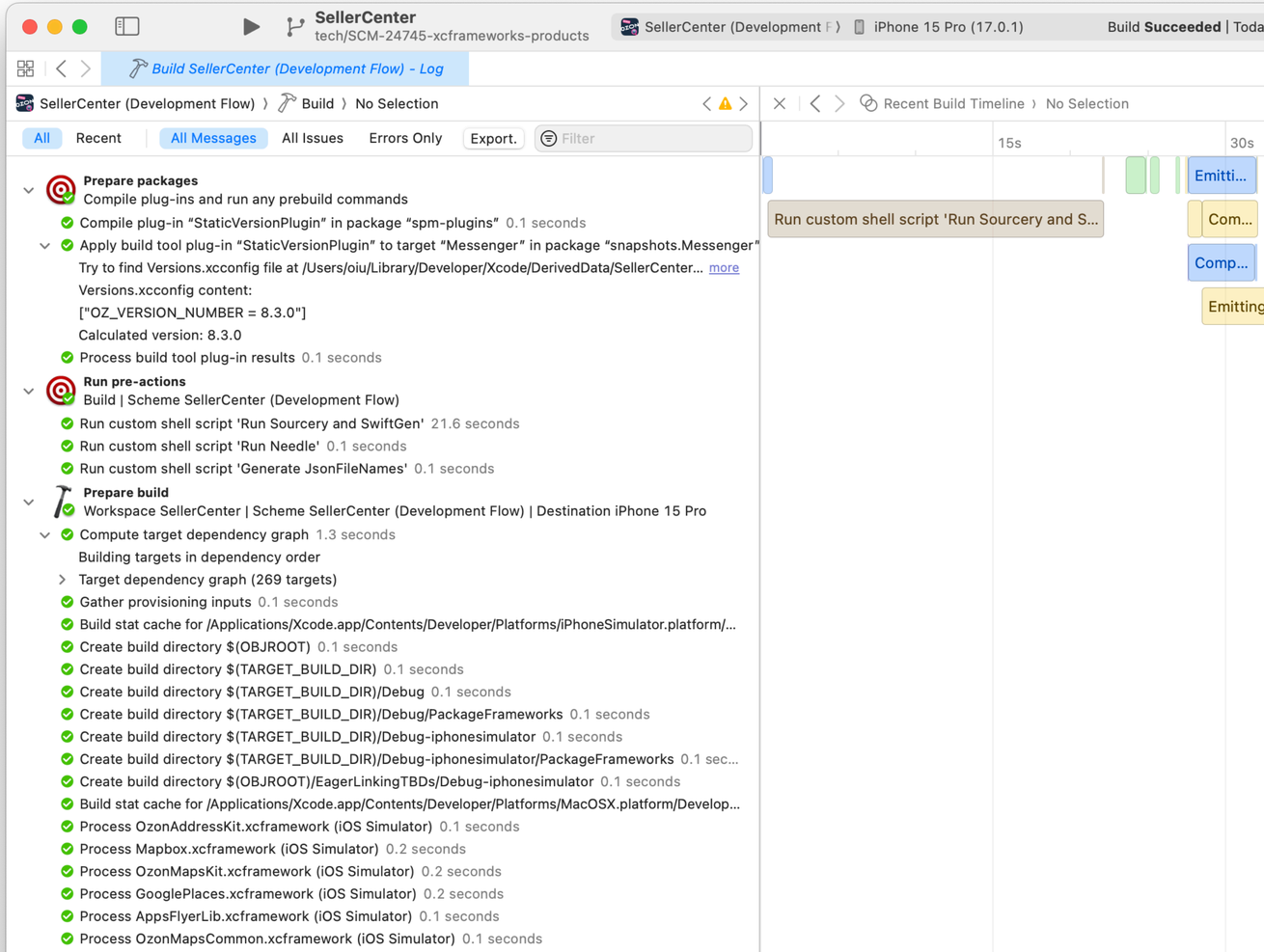

```
seller-app-ios — zsh — zsh (figterm) ▸ zsh — 68×14
...r-app-ios — zsh — zsh (figterm) ▸ zsh
...rnal — zsh — zsh (figterm) ▸ zsh
Last login: Sat Feb 24 23:21:44 on console
oiu@mbp-evgeryzhov-BYOD-WHH5P7YX2T seller-app-ios % make go
🛠️Закрытие Xcode
No matching processes belonging to you were found
💻 Установка git hooks
🔗 Простановка DI связей
🔧 Генерация ресурсов
🔨 Генерация проекта
Resolved cache profile 'Development' from Tuist's defaults
Generating workspace SellerCenter.xcworkspace
Generating project SellerCenter
Resolving package dependencies using xcodebuild
Project generated.
Total time taken: 334.816s
```



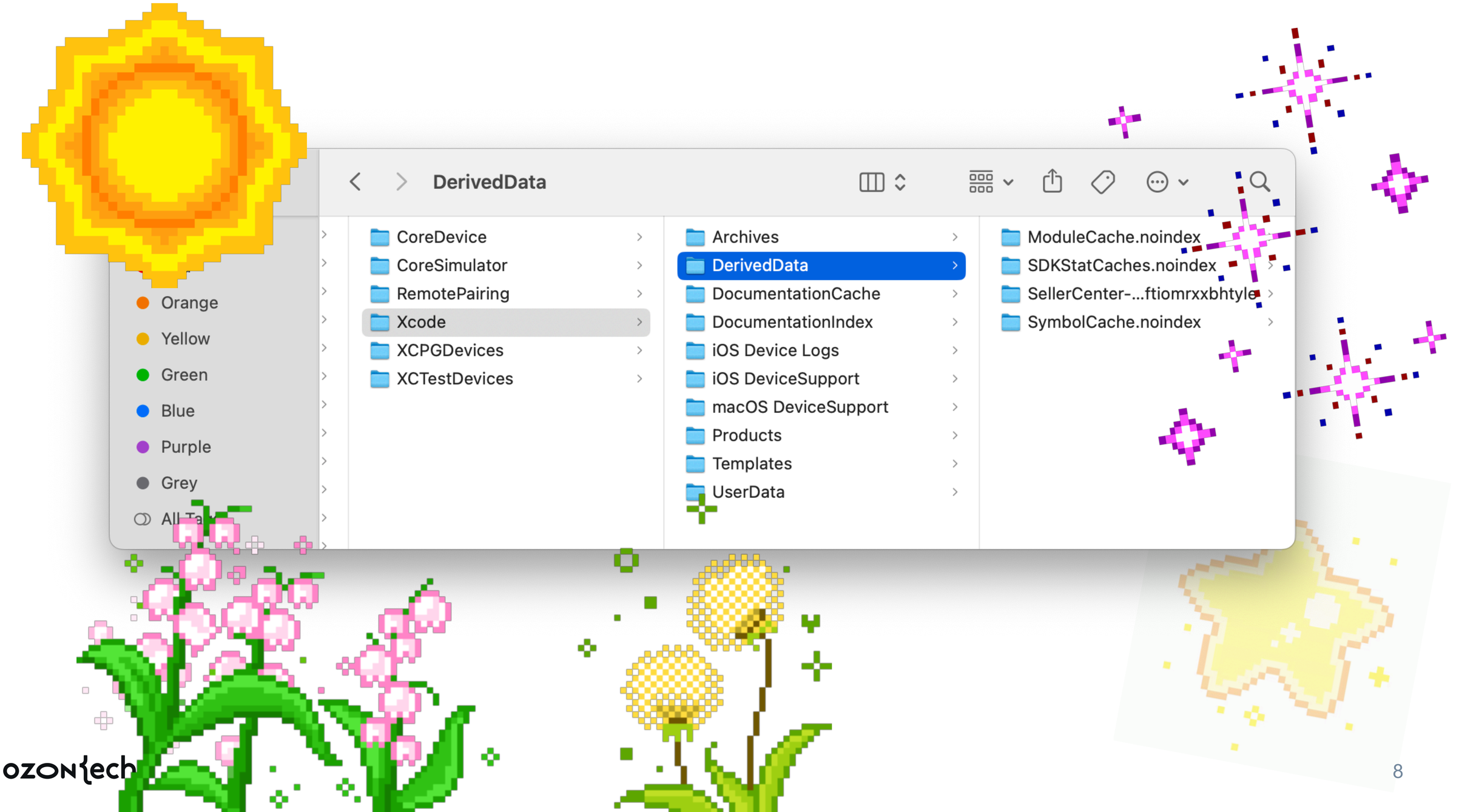
ozontech


Продолжительность, с.
360
70
690
???
Clean Build
Iterative Build
Resolve + Clean Build
Cached Clean Build


seller-app-ios — zsh — zsh (figterm) ▸ zsh — 68×14
...r-app-ios — zsh — zsh (figterm) ▸ zsh
...rnal — zsh — zsh (figterm) ▸ zsh
Last login: Sat Feb 24 23:21:44 on console
oiu@mbp-evgeryzhov-BYOD-WHH5P7YX2T seller-app-ios % make go
🛠️Закрытие Xcode
No matching processes belonging to you were found
💻 Установка git hooks
🔗 Простановка DI связей
🔧 Генерация ресурсов
🔨 Генерация проекта
Resolved cache profile 'Development' from Tuist's defaults
Generating workspace SellerCenter.xcworkspace
Generating project SellerCenter
Resolving package dependencies using xcodebuild
Project generated.
Total time taken: 334.816s


SellerCenter
tech/SCM-24745-xcframeworks-products
SellerCenter (Development F
iPhone 15 Pro (17.0.1)
Build Succeeded | Toda
Build SellerCenter (Development Flow) - Log
SellerCenter (Development Flow) ⟩ Build ⟩ No Selection
Recent Build Timeline ⟩ No Selection
All
Recent
All Messages
All Issues
Errors Only
Export.
Filter
15s
30s
Prepare packages
Compile plug-ins and run any prebuild commands
Compile plug-in "StaticVersionPlugin" in package "spm-plugins" 0.1 seconds
Apply build tool plug-in "StaticVersionPlugin" to target "Messenger" in package "snapshots.Messenger"
Try to find Versions.xcconfig file at /Users/oiu/Library/Developer/Xcode/DerivedData/SellerCenter... more
Versions.xcconfig content:
["OZ_VERSION_NUMBER = 8.3.0"]
Calculated version: 8.3.0
Process build tool plug-in results 0.1 seconds
Run pre-actions
Build | Scheme SellerCenter (Development Flow)
Run custom shell script 'Run Sourcery and SwiftGen' 21.6 seconds
Run custom shell script 'Run Needle' 0.1 seconds
Run custom shell script 'Generate JsonFileNames' 0.1 seconds
Prepare build
Workspace SellerCenter | Scheme SellerCenter (Development Flow) | Destination iPhone 15 Pro
Compute target dependency graph 1.3 seconds
Building targets in dependency order
Target dependency graph (269 targets)
Gather provisioning inputs 0.1 seconds
Build stat cache for /Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/...
Create build directory $(OBJROOT) 0.1 seconds
Create build directory $(TARGET_BUILD_DIR) 0.1 seconds
Create build directory $(TARGET_BUILD_DIR)/Debug 0.1 seconds
Create build directory $(TARGET_BUILD_DIR)/Debug/PackageFrameworks 0.1 seconds
Create build directory $(TARGET_BUILD_DIR)/Debug-iphonesimulator 0.1 seconds
Create build directory $(TARGET_BUILD_DIR)/Debug-iphonesimulator/PackageFrameworks 0.1 sec...
Create build directory $(OBJROOT)/EagerLinkingTBDs/Debug-iphonesimulator 0.1 seconds
Build stat cache for /Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Develop...
Process OzonAddressKit.xcframework (iOS Simulator) 0.1 seconds
Process Mapbox.xcframework (iOS Simulator) 0.2 seconds
Process OzonMapsKit.xcframework (iOS Simulator) 0.2 seconds
Process GooglePlaces.xcframework (iOS Simulator) 0.2 seconds
Process AppsFlyerLib.xcframework (iOS Simulator) 0.1 seconds
Process OzonMapsCommon.xcframework (iOS Simulator) 0.1 seconds
Run custom shell script 'Run Sourcery and S...
Emitti...
Com...
Comp...
Emitting

ozontech


DerivedData
Orange
Yellow
Green
Blue
Purple
Grey
CoreDevice
CoreSimulator
RemotePairing
Xcode
XCPGDevices
XCTestDevices
Archives
DerivedData
DocumentationCache
DocumentationIndex
iOS Device Logs
iOS DeviceSupport
macOS DeviceSupport
Products
Templates
UserData
ModuleCache.noindex
SDKStatCaches.noindex
SellerCenter-...ftiomrxxbhtyle
SymbolCache.noindex

О чём разговор

План действий

А может, лучше кофе? ☕

01 Немного о проекте

02 Base Benchmarking

03 Build Timeline

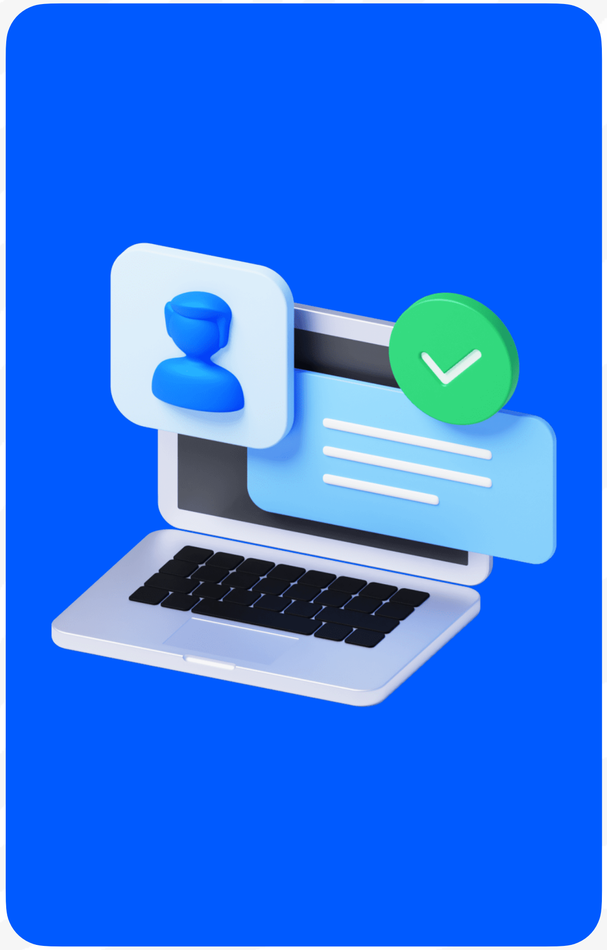
04 Инструменты Build-Cache зависимостей

05 Накладываем схему на SPM

06 Финальные бенчмарки

07 Но какой ценой?

08 Завершение



01

Немного о проекте

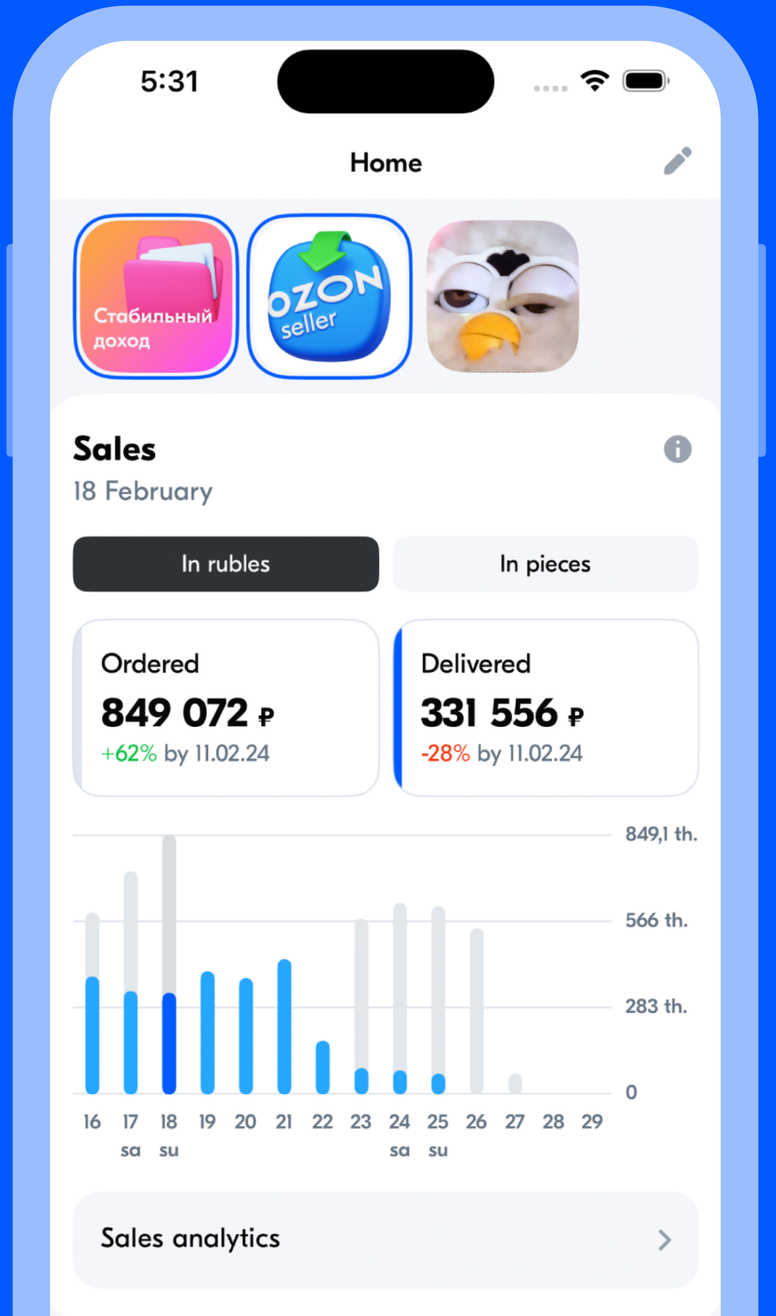
Ода **людям**

Кто мы?

Развиваем приложение продавца

Перевалили за **1 000 000 строк кода**

В команде **~30** красивых умных прекрасных iOS-разработчиков, распредёленных между 5 командами, и тестировщики, пишущие нативные UI-тесты



ozontech

Ода **техничке**

С чем мы работаем?

- **iOS 15+**, **SwiftUI 3**, **Async/Await**, вставьте своё смузи
- **SwiftPM** как менеджер зависимостей и как решение для многомодульности проекта
- **Tuist** для генерации проекта

Swift

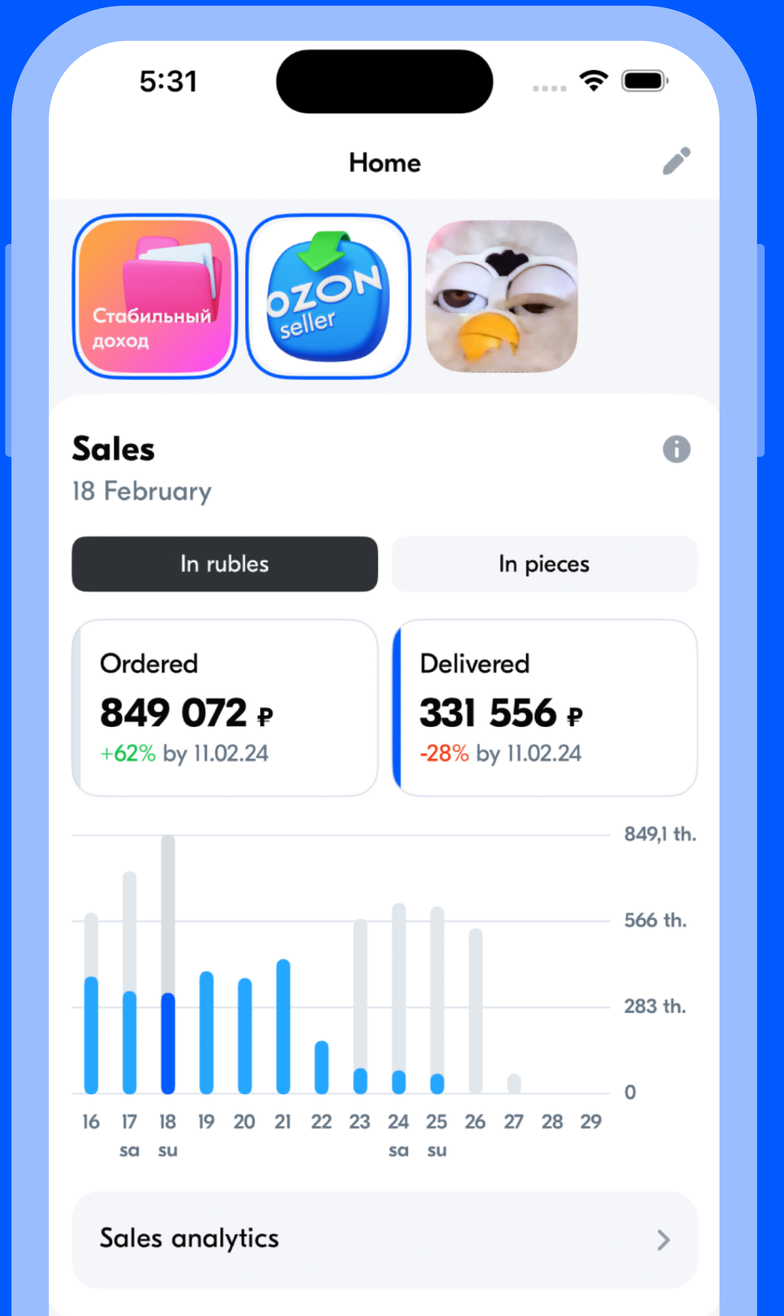
iOS 15+

SwiftUI 3

Gitlab

Tuist

SwiftPM



ozontech

SwiftPM

Менеджер зависимостей

SwiftPM

Историческое решение
для организации
многомодульности

- Рулит зависимостями проекта
- Нативен!
- Проинтегрировали SPM Registry

Tuist

Генератор генерируемого

Tuist

Генерация +
сетап .xcodeproj и
остального

- Пришел на смену XCodeGen
- Убирает возню с мердж-конфликтами
- Облегчает сетапы таргетов и схем

02

Base Benchmarking

Base Benchmarking ⏱️

Формула билдов

Merge Request

(стандартный)

Пробегут 3 пайплайна

1. Изначальный коммит
2. Коммит с фиксом тестов
3. Merge Train MR`a

По 3 билда

1. Тесты (Unit)
2. Сборка для QA
3. Ещё тесты (UI)

Base Benchmarking ⏱️

Формула билдов

$$3 \times 3 - 2 = 7 \quad \times \quad 12 \text{ мин. } 30 \text{ сек.}$$

Упавший пайп — **Стандартная сборка**

...

$$87 \text{ мин. } 30 \text{ сек.} \times 30 \text{ человек}$$

Base Benchmarking ⏱️

Ужасы бытия

≈ 44 часа

Base Benchmarking ⏱️

Формула билдов

12 мин. 30 сек.

Стандартная сборка

87 мин. 30 сек.

Машинное время билдов MP-a

Base Benchmarking ⏱️

Формула билдов

12 мин. 30 сек.

Стандартная сборка

87 мин. 30 сек.

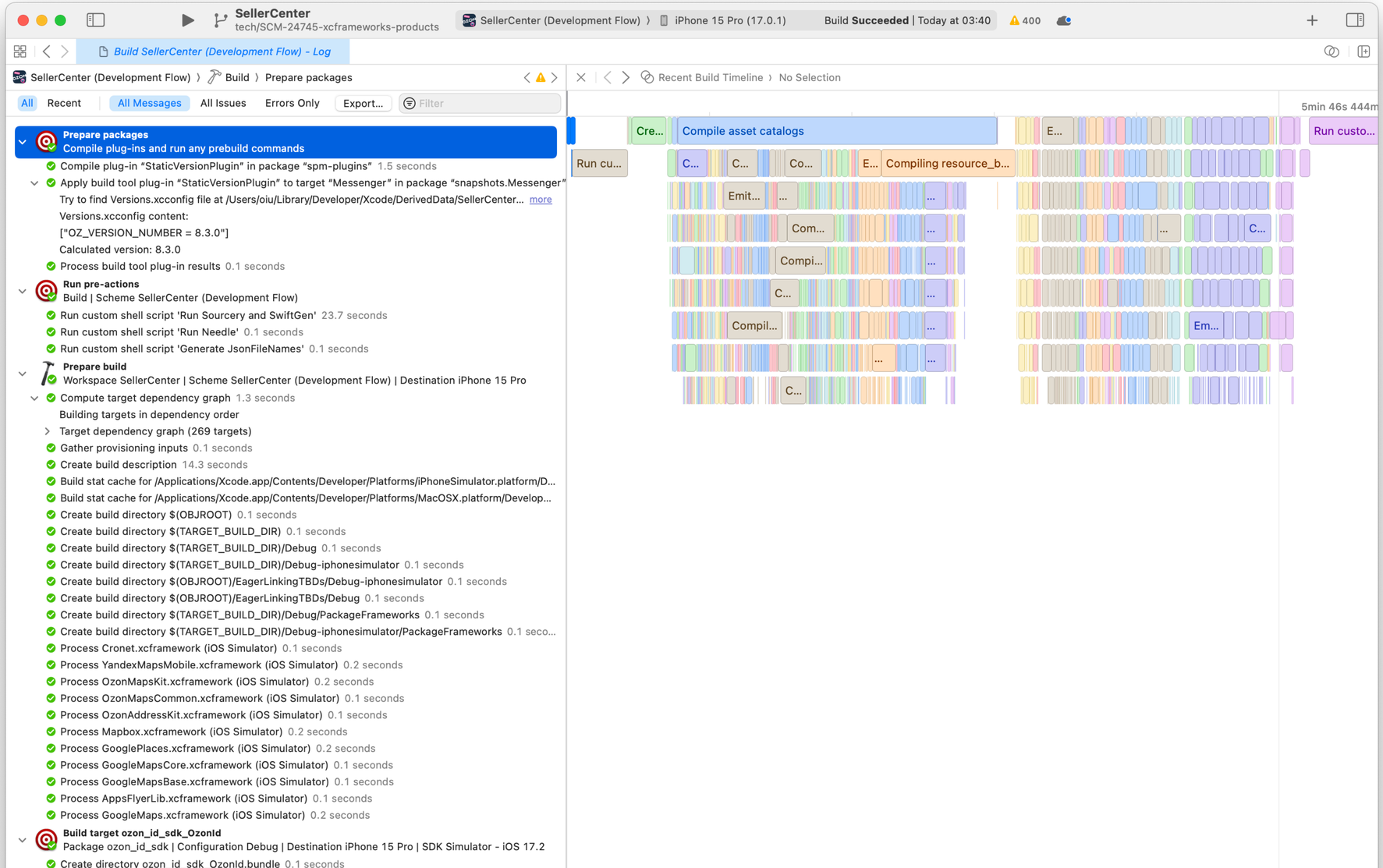
Машинное время билдов MP-а

03

Build Timeline

Build Timeline

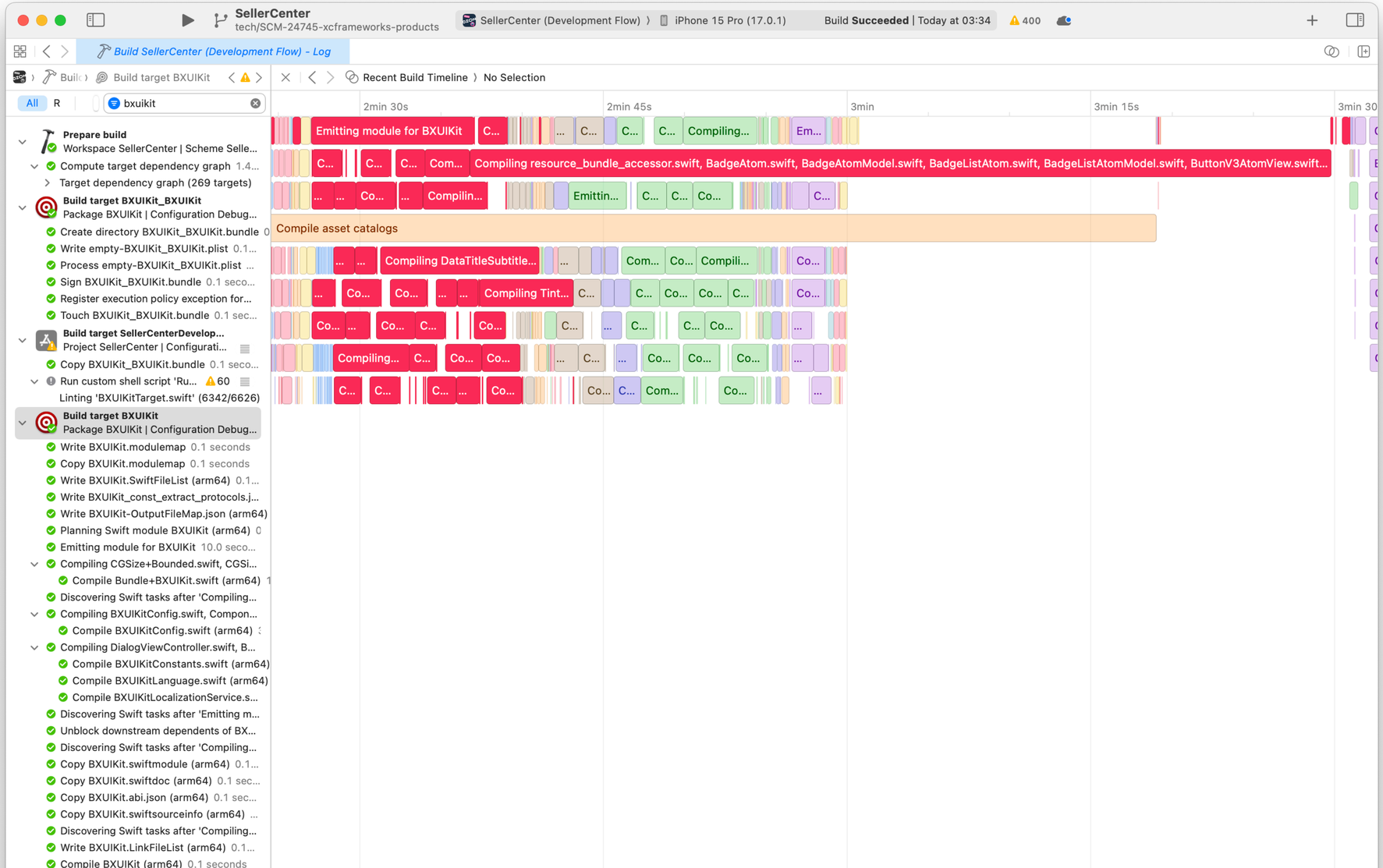
Кто, зачем и сколько?



ozontech

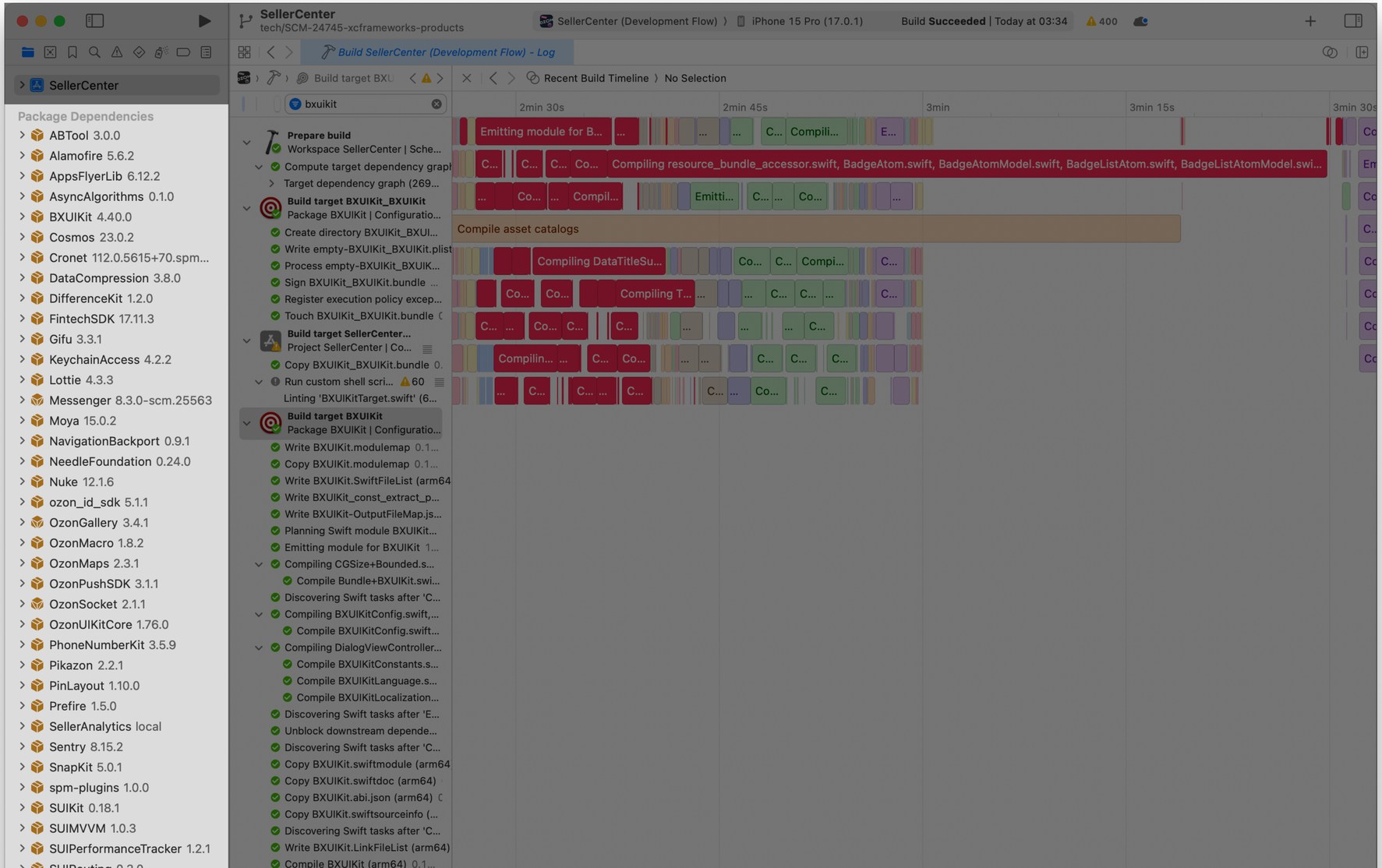
Build Timeline

Кто, зачем и сколько?



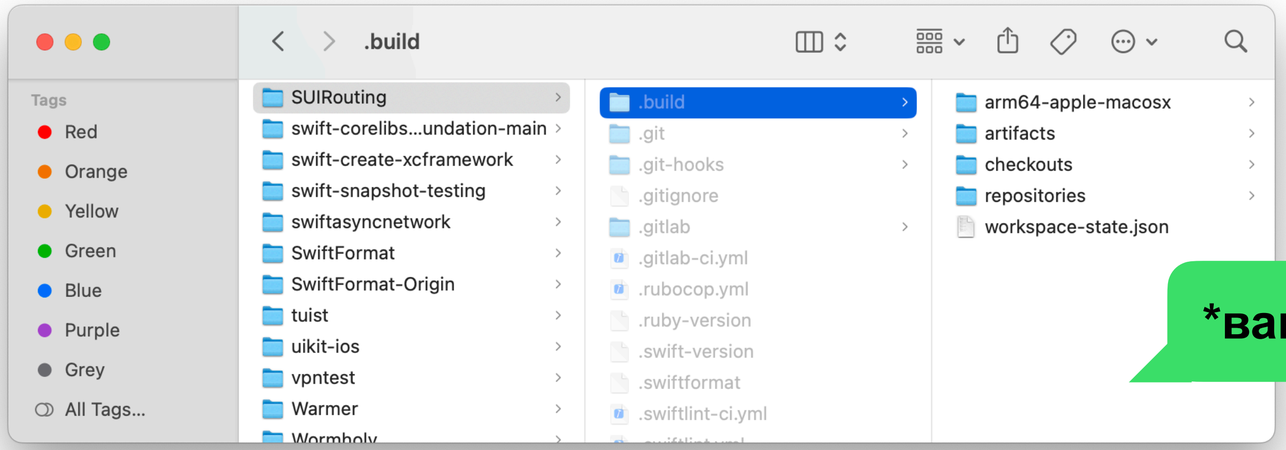
Build Timeline

Кто, зачем и сколько?



Вычеркиваем чекаут зависимостей

— Для plain Package.swift-файла



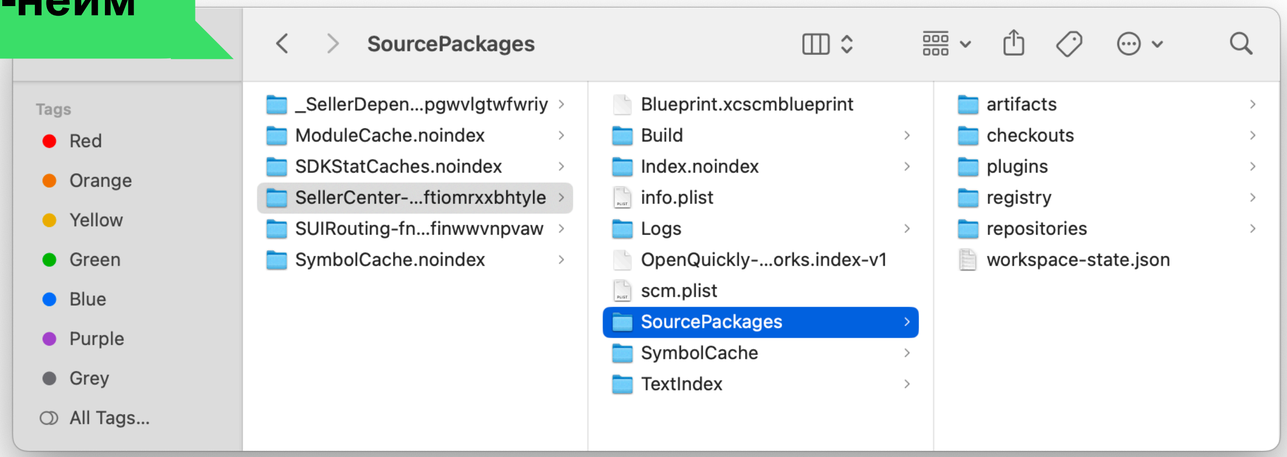

```
1 swift package resolve \
2     --build-path *ваша-папка-нейм* \
3     ...
```

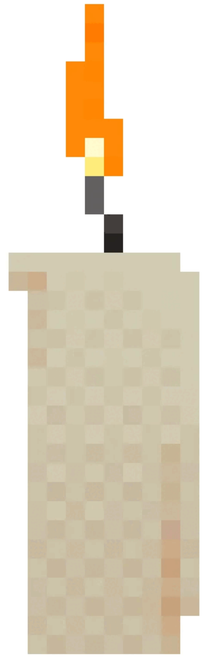
Вычеркиваем чекаут зависимостей

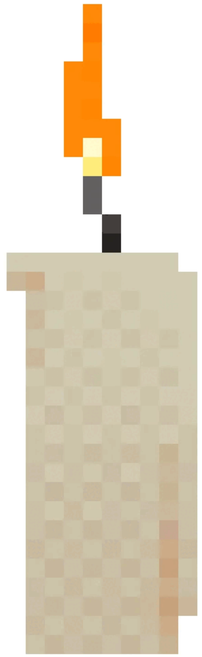
— Для *.xcodeproj-файла

```
1 xcodebuild build \
2     -scheme *ваша-схема-нейм* \
3     -clonedSourcePackagesDirPath *ваша-папка-нейм* \
4     ...
```

ваша-папка-нейм







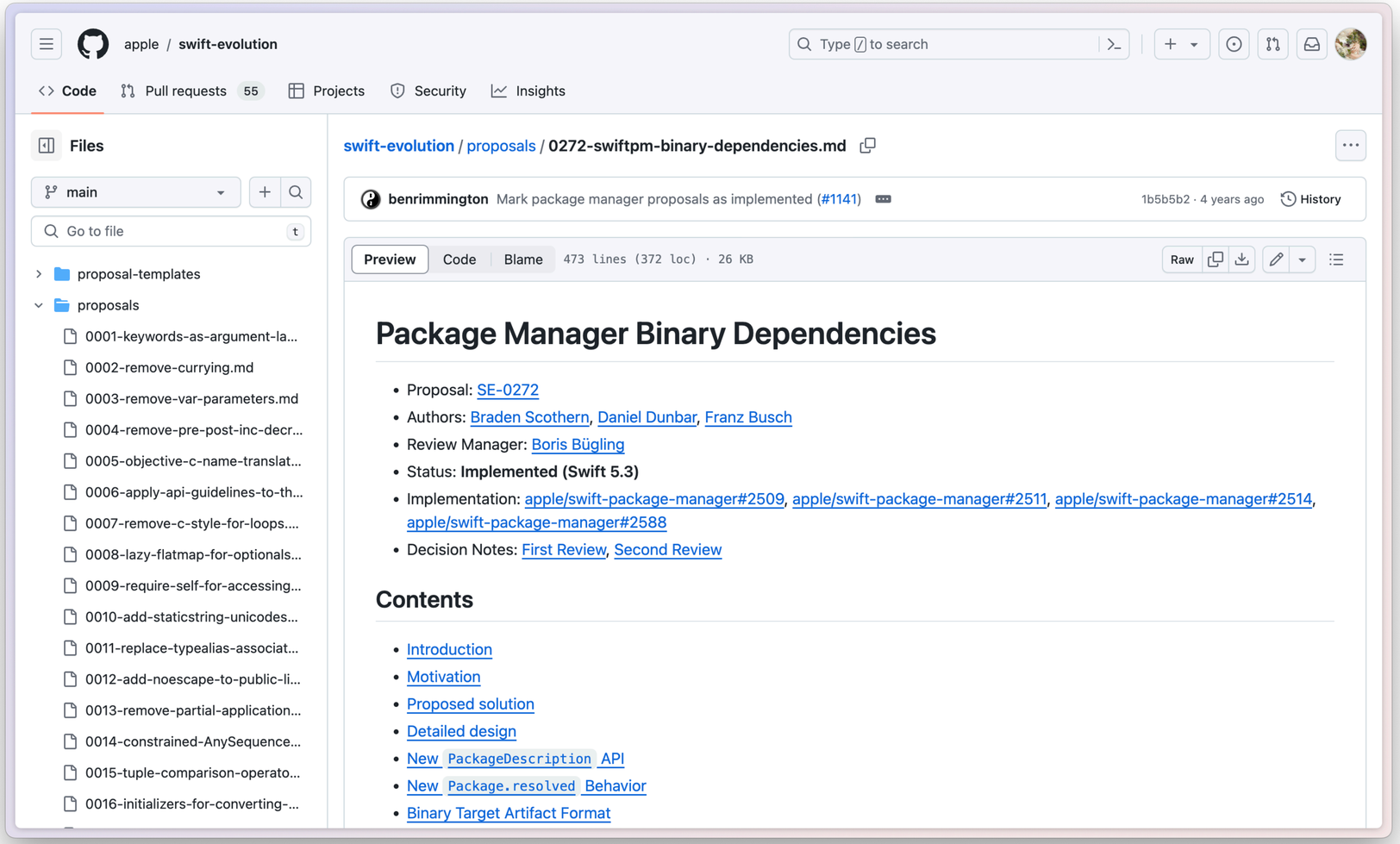
Вычеркиваем билд сурсов зависимостей

Вычеркиваем же, правда?

```swift
// swift-tools-version:5.3
import PackageDescription

let package = Package(
    name: "AppsFlyerLib",
    products: [
        .library(
            name: "AppsFlyerLib",
            targets: ["AppsFlyerLib"]
        )
    ],
    targets: [
        .binaryTarget(
            name: "AppsFlyerLib",
            url: "https://github.com/AppsFlyerSDK/AppsFlyerFramework/releases/download/"
                  + "6.14.3/AppsFlyerLib-Static-SPM.xcframework.zip",
            checksum: "53c824272e8729967af6b22e013577eccbc6e412b7e2da2805ddf3c1ecf7cfc5"
        )
    ]
)
```

Вычеркиваем билд сурсов зависимостей

Вычеркиваем же, правда?



Вычеркиваем билд сурсов зависимостей

Вычеркиваем же, правда?

```swift
// swift-tools-version:5.3
import PackageDescription

let package = Package(
    name: "AppsFlyerLib",
    products: [
        .library(
            name: "AppsFlyerLib",
            targets: ["AppsFlyerLib"]
        )
    ],
    targets: [
        .binaryTarget(
            name: "AppsFlyerLib",
            url: "https://github.com/AppsFlyerSDK/AppsFlyerFramework/releases/download/"
                  + "6.14.3/AppsFlyerLib-Static-SPM.xcframework.zip",
            checksum: "53c824272e8729967af6b22e013577eccbc6e412b7e2da2805ddf3c1ecf7cfc5"
        )
    ]
)
```

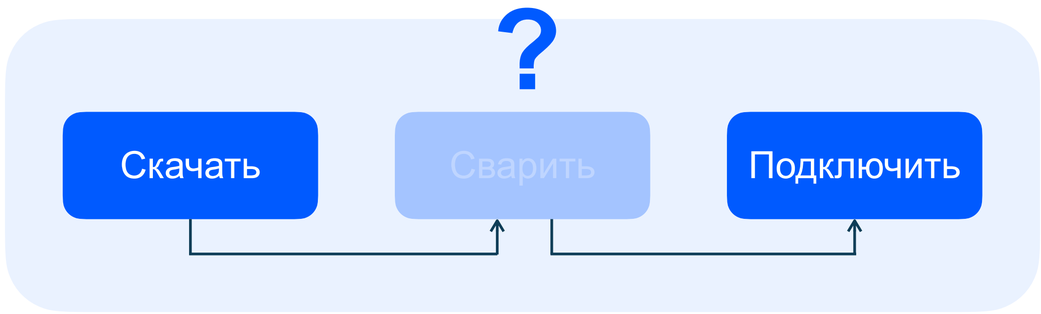
~~Вычеркиваем билд сурсов зависимостей~~

Так просто не выйдет 😭

```swift
// swift-tools-version:5.10
import PackageDescription

let package = Package(
    name: "Alamofire",
    products: [
        .library(name: "Alamofire", targets: ["Alamofire"]),
        .library(name: "AlamofireDynamic", type: .dynamic, targets: ["Alamofire"])
    ],
    targets: [
        .target(
            name: "Alamofire",
            path: "Source",
            exclude: ["Info.plist"],
            resources: [
                .process("PrivacyInfo.xcprivacy")
            ],
            linkerSettings: [
                .linkedFramework(
                    "CFNetwork",
                    .when(platforms: [.iOS, .macOS, .tvOS, .watchOS])
                ),
            ],
        ),
    ]
)
```

~~Нормальные~~ инструменты

SPM




```
1 swift package generate-xcodeproj ...
```

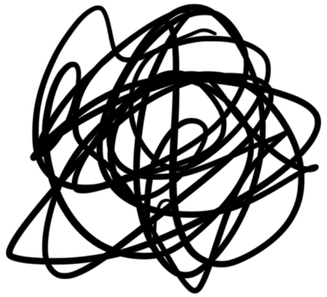
swift-create-xcframework Public

spm-to-xcframework Public

~~Вычеркиваем билд сурсов зависимостей~~

Так просто не выйдет 😭

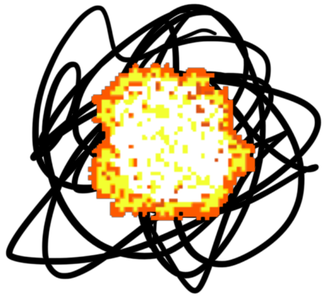
```swift
// swift-tools-version:5.10
import PackageDescription

let package = Package(
    name: "Alamofire",
    products: [
        .library(name: "Alamofire", targets: ["Alamofire"]),
        .library(name: "AlamofireDynamic", type: .dynamic, targets: ["Alamofire"])
    ],
    targets: [
        .target(
            name: "Alamofire",
            path: "Source",
            exclude: ["Info.plist"],
            resources: [
                .process("PrivacyInfo.xcprivacy")
            ],
            linkerSettings: [
                .linkedFramework(
                    "CFNetwork",
                    .when(platforms: [.iOS, .macOS, .tvOS, .watchOS])
                ),
            ],
        ),
    ]
)
```



~~Вычеркиваем билд сурсов зависимостей~~

Так просто не выйдет 😭

```swift
// swift-tools-version:5.3

import PackageDescription

let package = Package(
    name: "Moya",
    platforms: [
        .macOS(.v10_12),
        .iOS(.v10),
        .tvOS(.v10),
        .watchOS(.v3)
    ],
    products: [
        .library(name: "Moya", targets: ["Moya"]),
        .library(name: "CombineMoya", targets: ["CombineMoya"]),
        .library(name: "ReactiveMoya", targets: ["ReactiveMoya"]),
        .library(name: "RxMoya", targets: ["RxMoya"])
    ],
    dependencies: [
        .package(url: "https://github.com/Alamofire/Alamofire.git", .upToNextMajor(from: "5.0.0")),
        .package(url: "https://github.com/ReactiveCocoa/ReactiveSwift.git", .upToNextMajor(from: "6.0.0")),
        .package(url: "https://github.com/ReactiveX/RxSwift.git", .upToNextMajor(from: "6.0.0")),
        .package(url: "https://github.com/Quick/Quick.git", .upToNextMajor(from: "4.0.0")), // dev
        .package(url: "https://github.com/Quick/Nimble.git", .upToNextMajor(from: "9.0.0")), // dev
        .package(url: "https://github.com/AliSoftware/OHHTTPStubs.git", .upToNextMajor(from: "9.0.0")) //
dev
    ],
    targets: [
        .target(
            name: "Moya",
            dependencies: [
                .product(name: "Alamofire", package: "Alamofire")
            ],
            exclude: [
                "Supporting Files/Info.plist"
            ]
        ),
        .target(
            name: "CombineMoya",
            dependencies: [
```

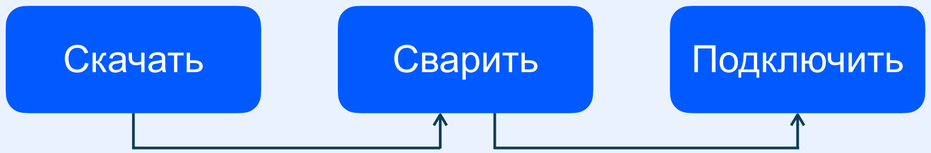


04

Инструменты Build-Cache зависимостей

Нормальные инструменты

Carthage

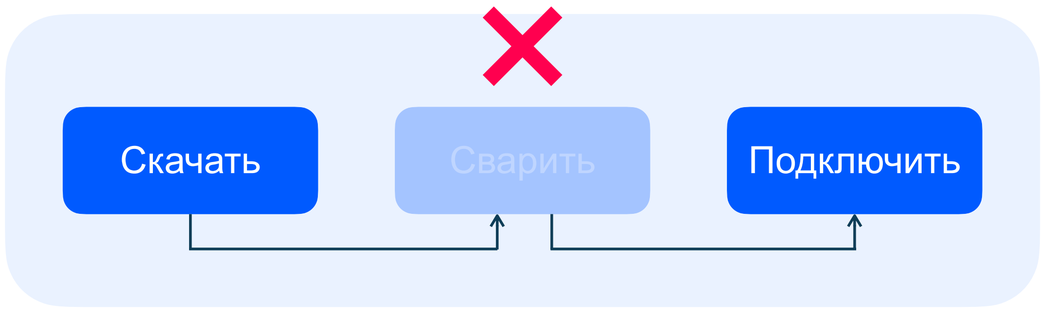



```
# Cartfile

github "Alamofire/Alamofire" ~> 5.5
```

```
oiu@kalkulator oiu % carthage update \
    --use-xcframeworks
```

~~Нормальные~~ инструменты

CocoaPods

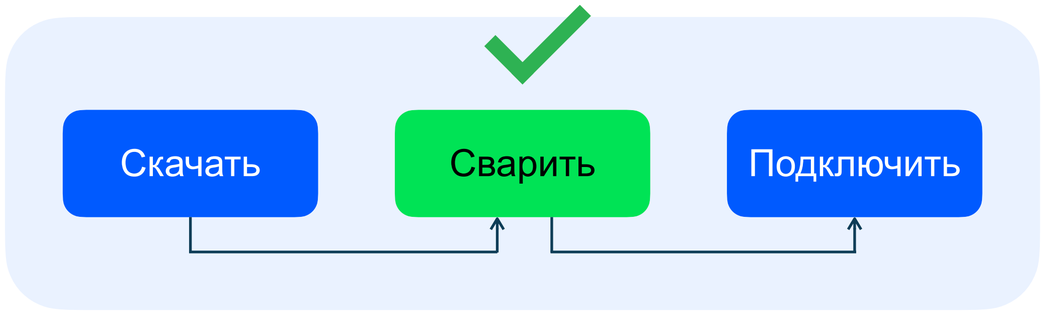



```
workspace 'Examples.xcworkspace'
project 'iOS Example.xcodeproj'

target 'iOS Example' do
  platform :ios, '8.0'
  use_frameworks!
  pod 'Alamofire', :path => './Alamofire'
end
```

```
oiu@kalkulator oiu % pod install
```

Нормальные инструменты

~~CocoaPods~~ Rugby




```
1 workspace 'Examples.xcworkspace'
2 project 'iOS Example.xcodeproj'
3
4 target 'iOS Example' do
5   platform :ios, '8.0'
6   use_frameworks!
7   pod 'Alamofire', :path => './Alamofire'
8 end
```

```
1 oiu@kalkulator oiu % pod install
2 oiu@kalkulator oiu % rugby
```

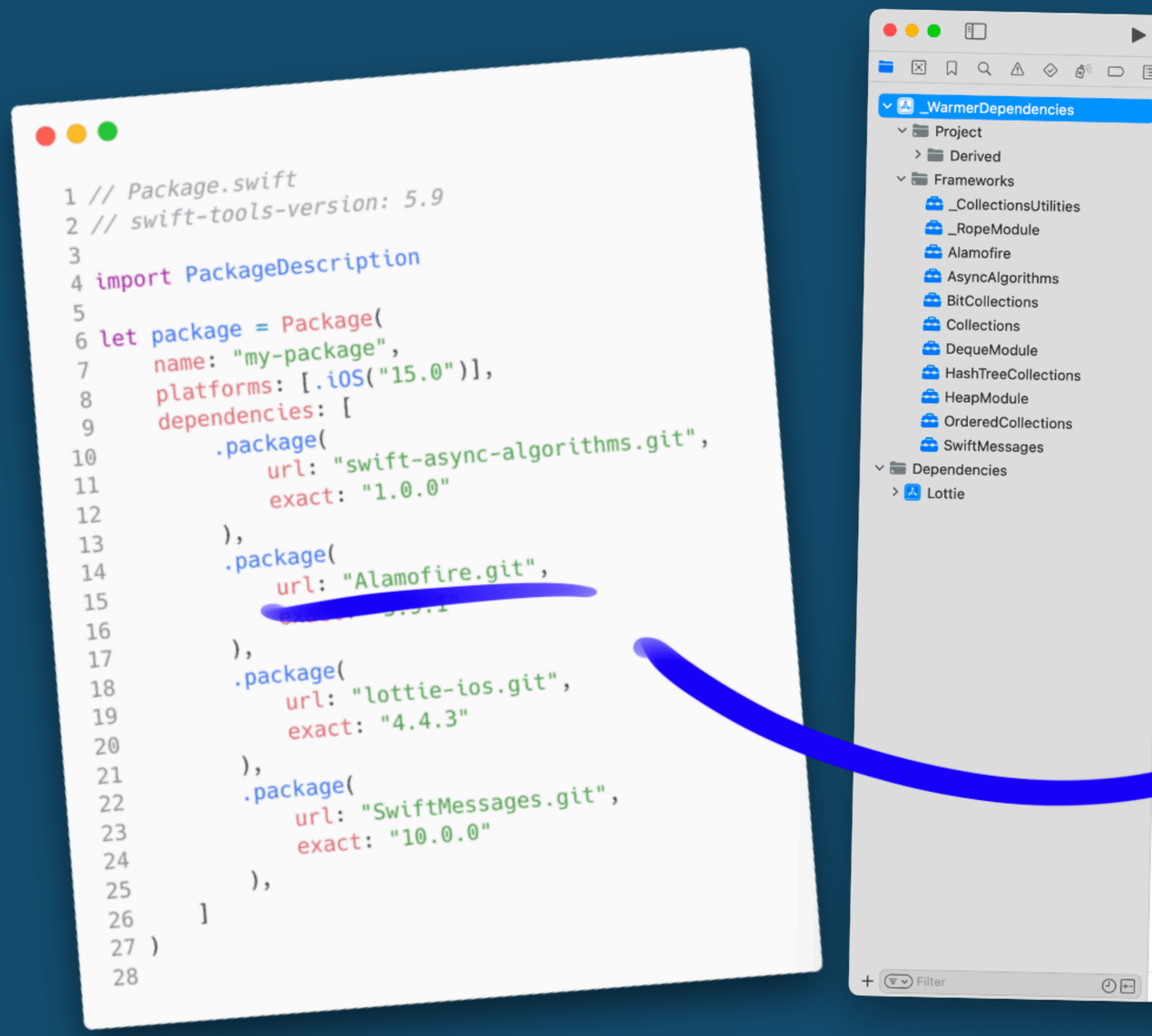
Нормальные инструменты

Tuist

Нормальные инструменты

Tuist

```
1 oiu@kalkulator oiu % tuist install
2 oiu@kalkulator oiu % tuist cache
3 oiu@kalkulator oiu % tuist generate
```



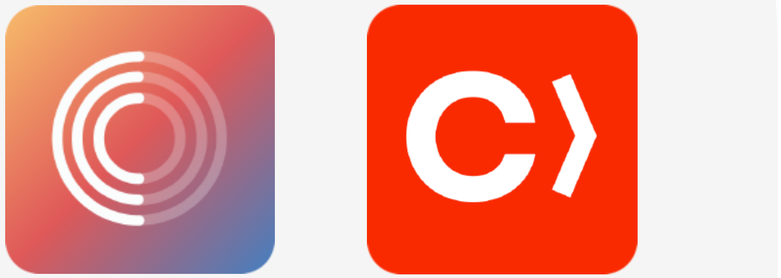
Сводная таблица инструментов

И их возможностей

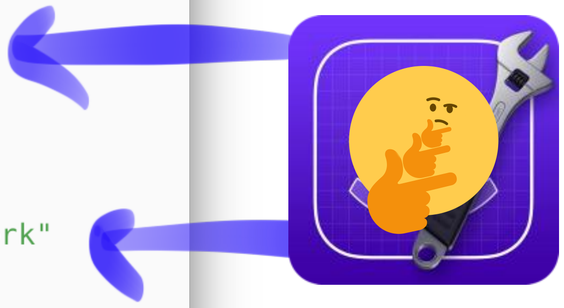
	Скачать	Закэшировать	Подключить
Carthage	✅	✅	✅⛔
CocoaPods	✅	⛔	✅
Rugby	✅	✅	✅
Tuist	✅	✅	✅
Swift Package Manager	✅	⛔	✅

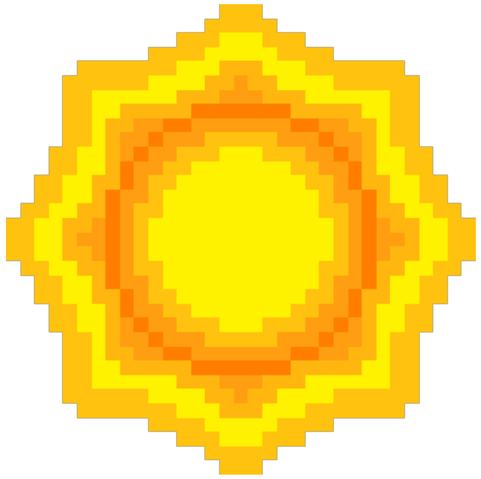
Почему не использовать готовое?

Overview




```swift
// swift-tools-version: 5.9

import PackageDescription

let package = Package(
    name: "my-feature",
    platforms: [.iOS(.v15)],
    products: [
        .library(
            name: "my-feature",
            targets: ["my-feature"]
        ),
    ],
    targets: [
        .target(
            name: "my-feature",
            dependencies: [
                .target(name: "Nuke"),
            ]
        ),
        .binaryTarget(
          name: "Nuke",
          path: "Sources/Nuke.xcframework"
        ),
    ]
)
```



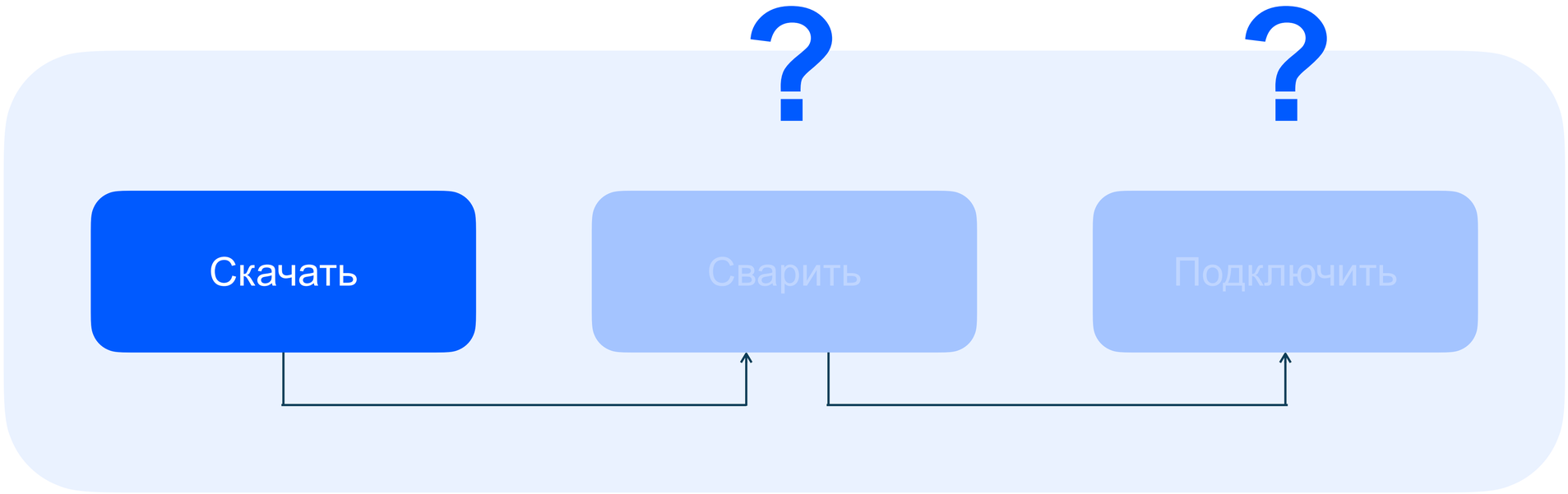


05

Накладываем схему на SPM

Восстанавливаем схему для SPM

Чего нам не хватает?

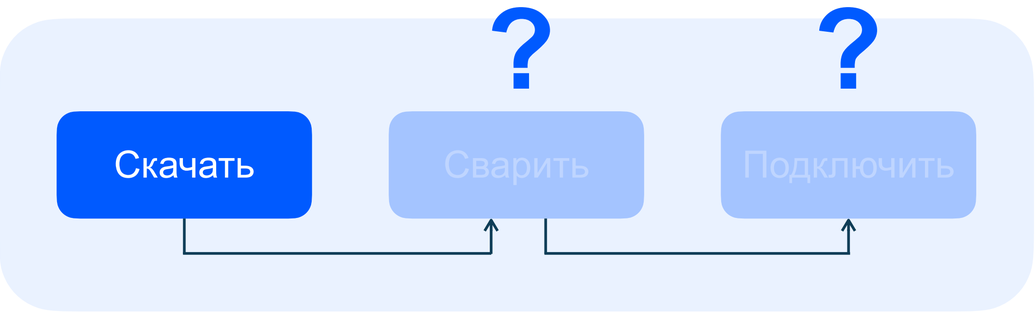


Инструмент может быть любым

Подключение бинарных таргетов в манифест в любом случае придется делать руками, а откуда возьмутся сами бинари — абсолютно неважно

Выборы кандидатов

*в случае нашего проекта



Зачем?

Уже в проекте

подставьте свой инструмент-нейм

Кусочки недостающего пазла

Pen Pineapple Apple Pen

1

```
1 oiu@kalkulator oiu % tuist install
```

2

```
1 oiu@kalkulator oiu % tuist cache
```

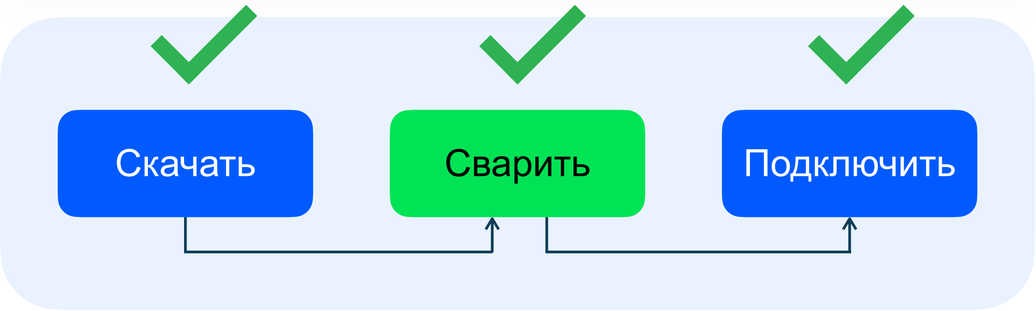
3

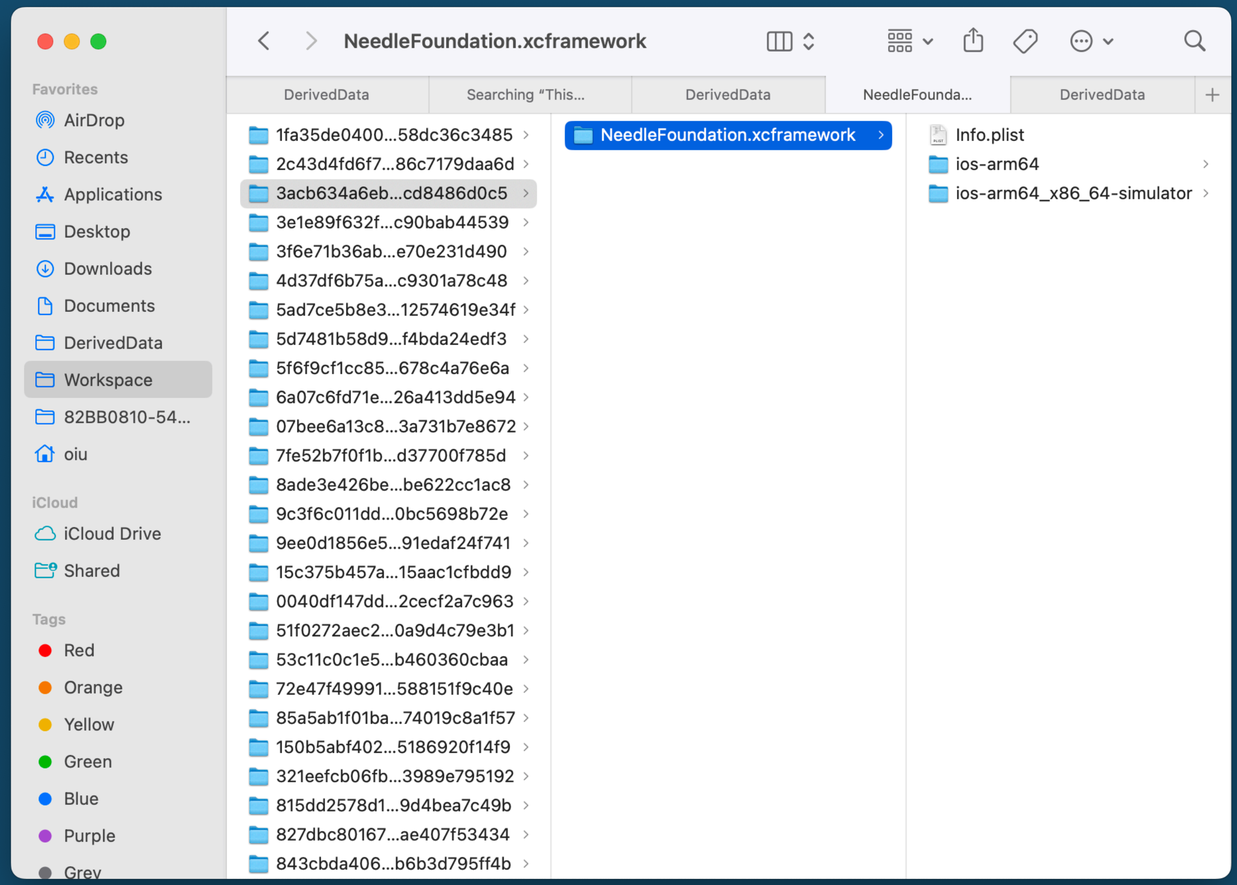
```
1 oiu@kalkulator oiu % tuist generate
```

Нормальные инструменты

Tuist

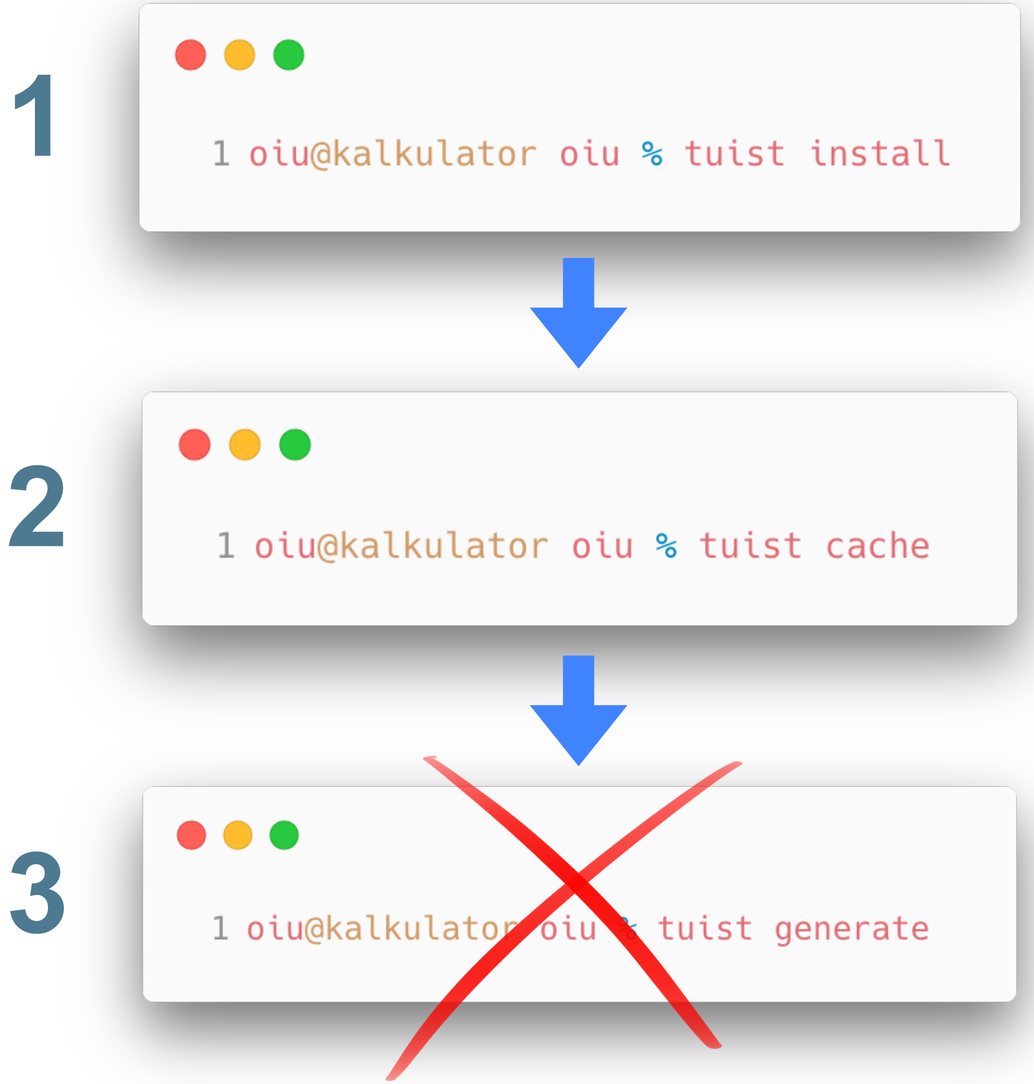
```
1 oiu@kalkulator oiu % tuist install
2 oiu@kalkulator oiu % tuist cache
3 oiu@kalkulator oiu % tuist generate
```





Кусочки недостающего пазла

Pen Pineapple Apple Pen



Варианты подключения

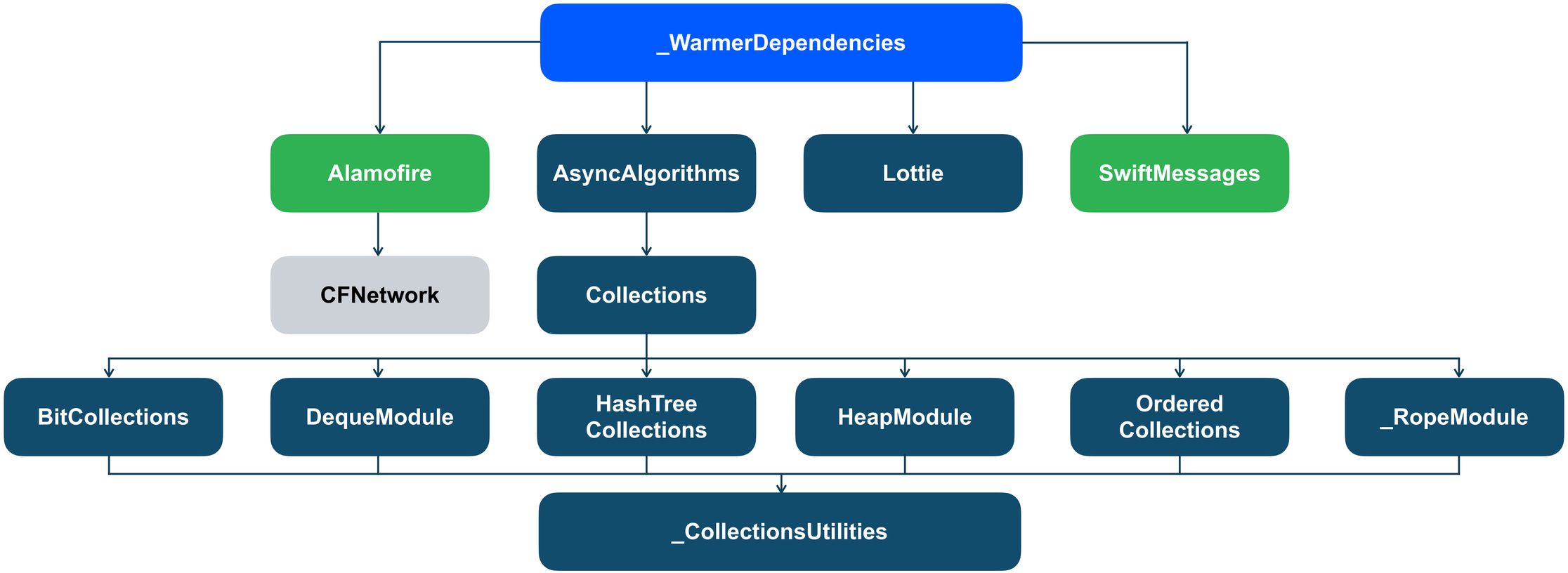
Первый: **попроще**

```swift
// swift-tools-version: 5.9

import PackageDescription

let package = Package(
    name: "my-feature",
    platforms: [.iOS(.v15)],
    products: [
        .library(
            name: "my-feature",
            targets: ["my-feature"]
        ),
    ],
    targets: [
        .target(
            name: "my-feature",
            dependencies: [
                .target(name: "Nuke"),
            ]
        ),
        .binaryTarget(
          name: "Nuke",
          path: "Sources/Nuke.xcframework"
        ),
    ]
)
```

Варианты подключения

Второй: **вариативнее**

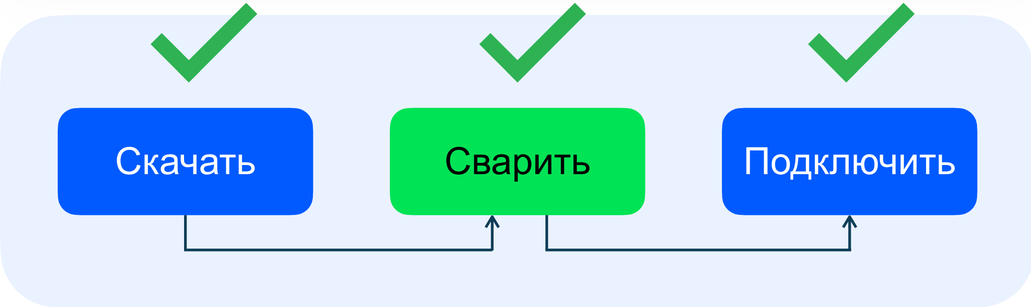
```
oiu@kalkulator oiu % tuist graph
    // опционально: -f json
```

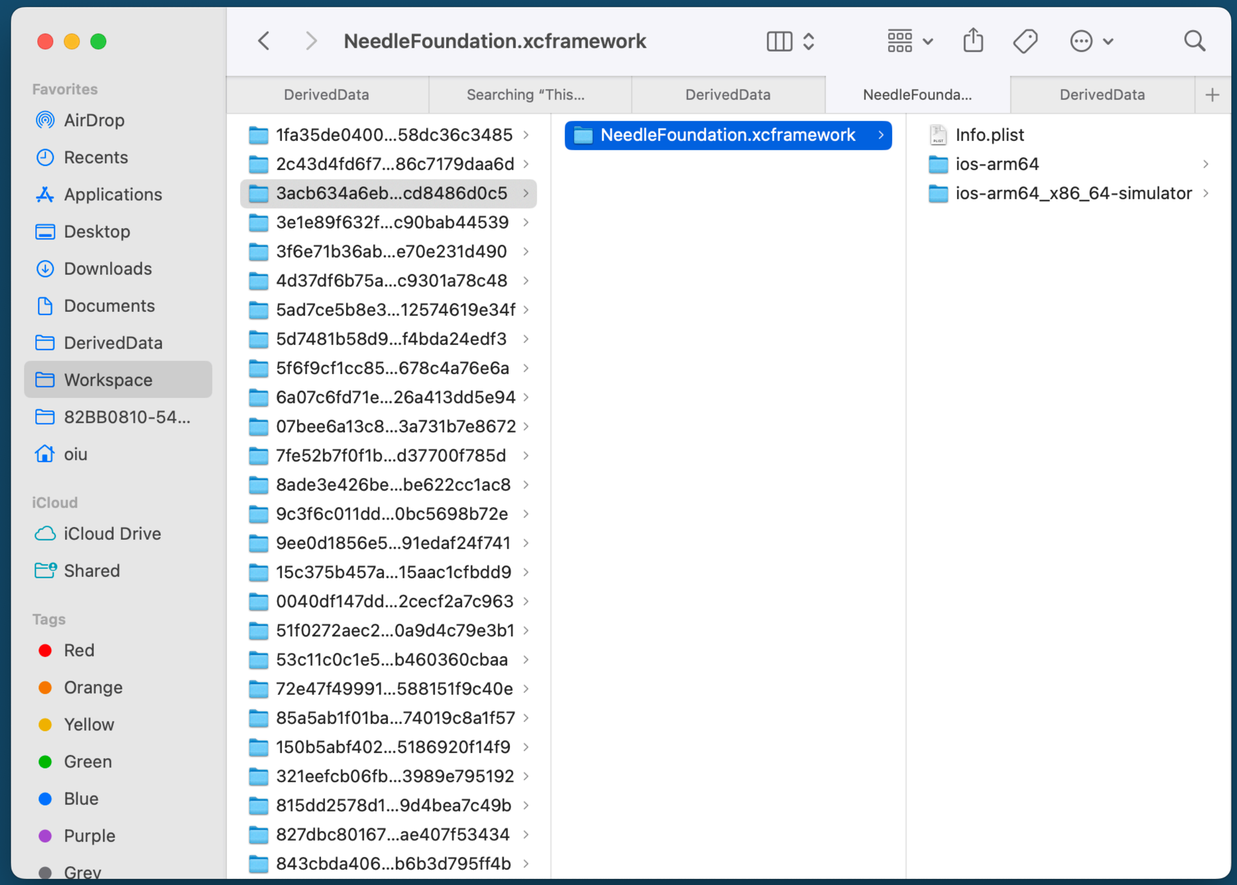


Третий кусочек пазла

Подключение

```
1 oiu@kalkulator oiu % tuist install
2 oiu@kalkulator oiu % tuist cache
3 oiu@kalkulator oiu % tuist generate
```

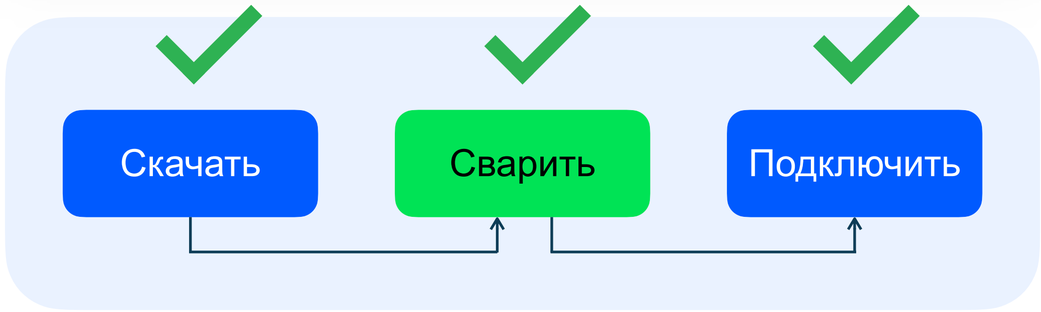


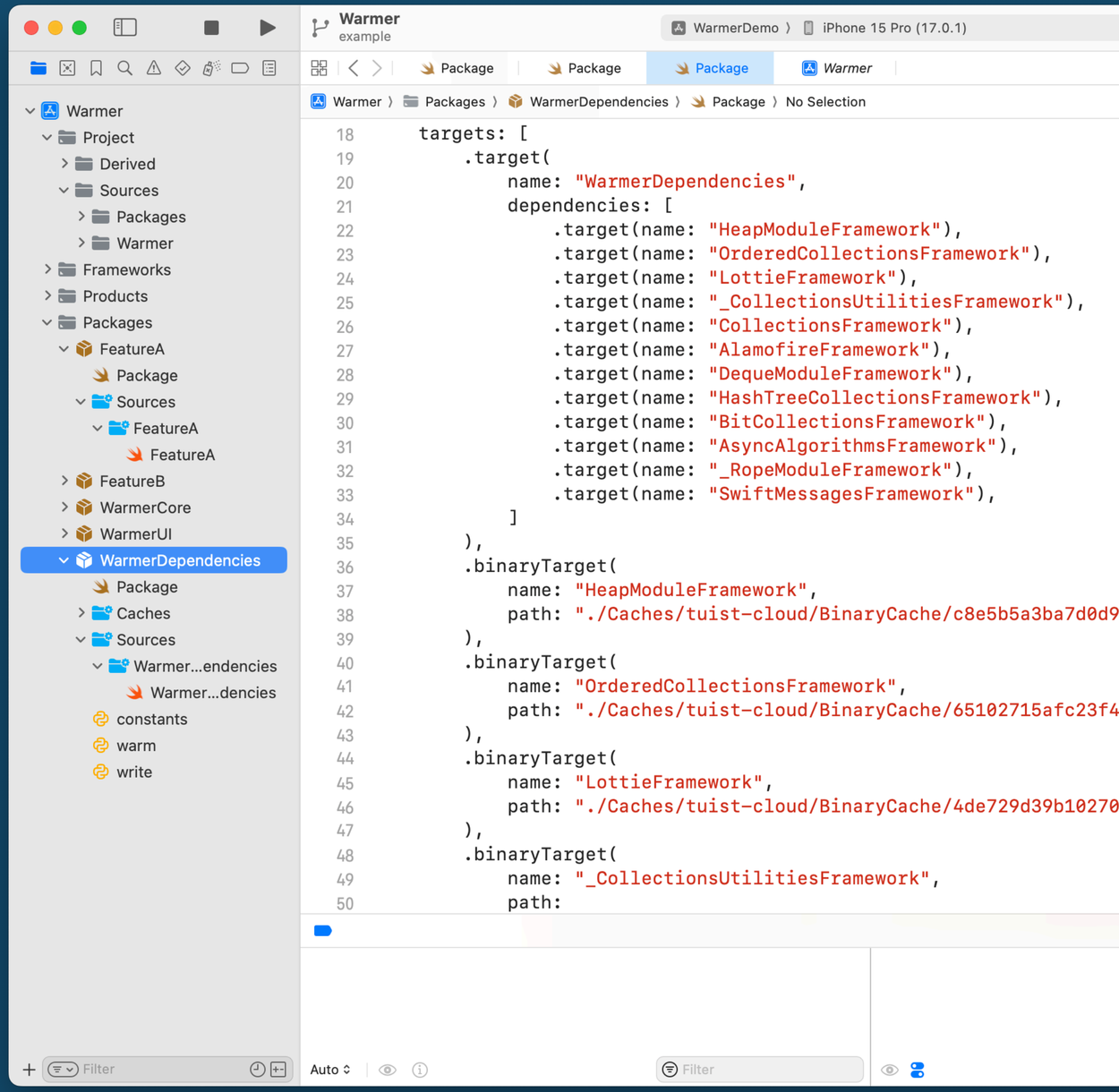


Третий кусочек пазла

Подключение

```
1 oiu@kalkulator oiu % tuist install
2 oiu@kalkulator oiu % tuist cache
3 oiu@kalkulator oiu % tuist generate
```





ozontech

Источники

Вместо лайв-кодинга 😭

1. **Изначальный коммит**
2. **Заводим “пакет с пакетами”**
3. **Обвязка Tuist`a — собираем фреймворки**
4. **Скрипты генерации**
5. Хэширование версий зависимостей
6. Бандлы SPM-зависимостей
7. Генерация проекта исходниками

ozontech

07

Финальные бенчмарки

Базовые планки

Бьём?

12 мин. 30 сек.

Стандартная сборка

87 мин. 30 сек.

Машинное время билдов MP-а

Базовые планки

Рвём

12 мин. 30 сек.

Стандартная сборка

6 мин. 42 сек.

Стандартная сборка

87 мин. 30 сек.

Машинное время билдов MP-a

46 мин. 50 сек.

Машинное время билдов MP-a

Итоговые результаты

Ну по большей части рвём

12мин. 30сек.

Стандартная сборка

6мин. 42сек.

Стандартная сборка

19мин. 10сек.

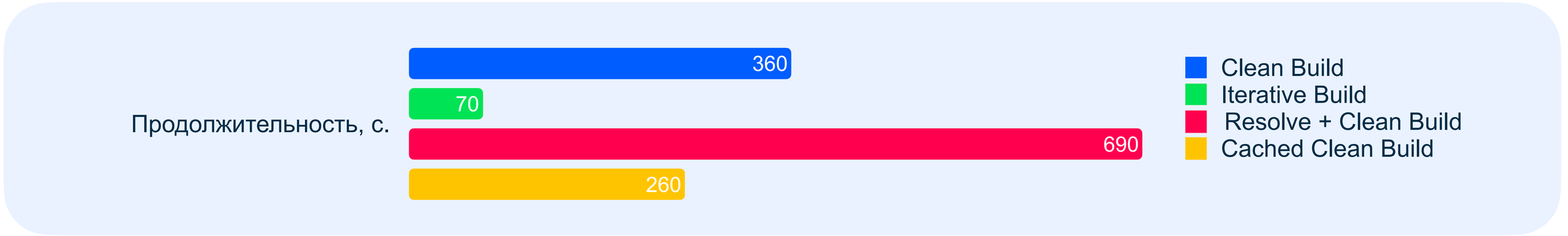
Стандартная сборка (cacheless)

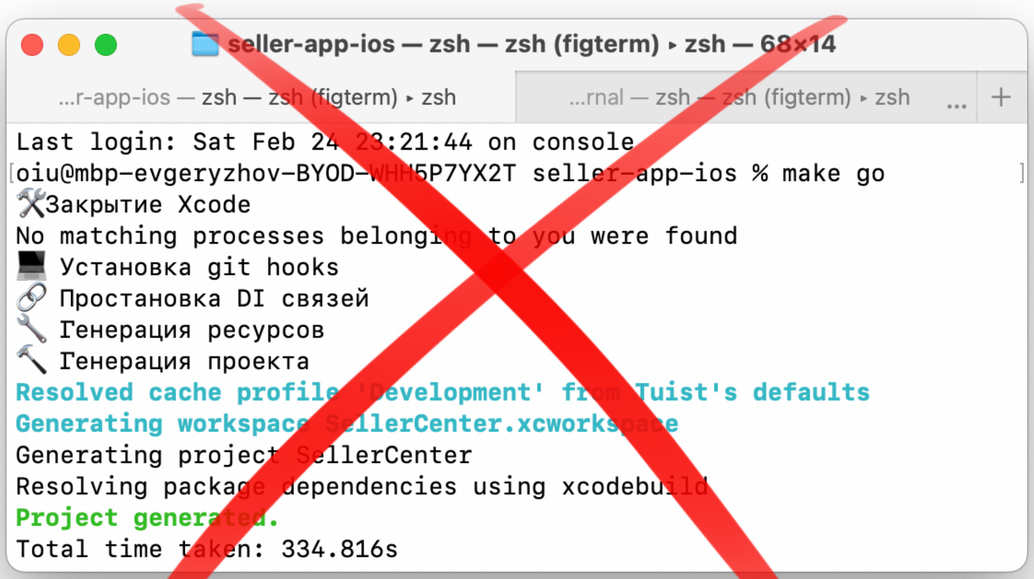
87мин. 30сек.

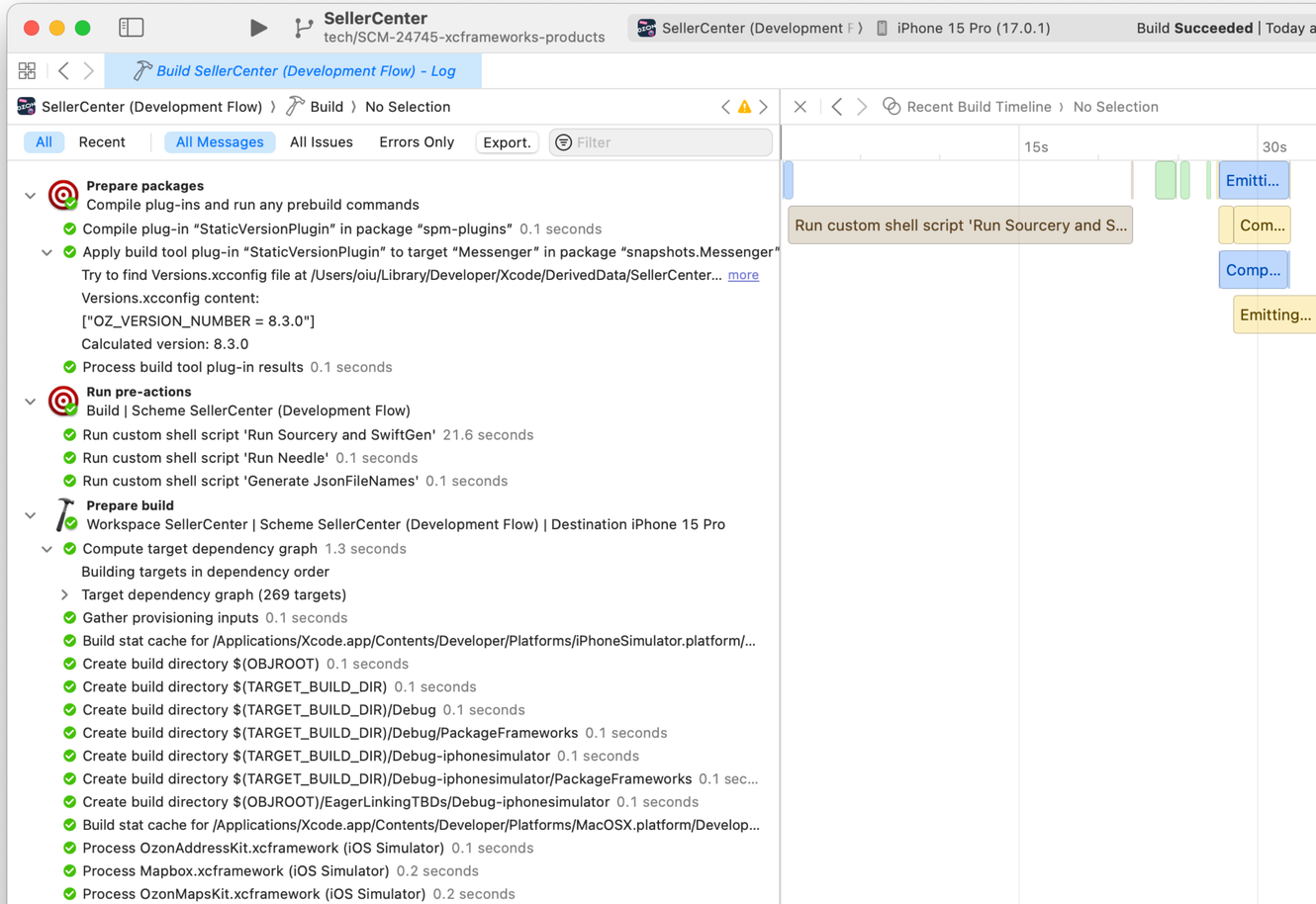
Машинное время билдов MP-a

46мин. 50сек.

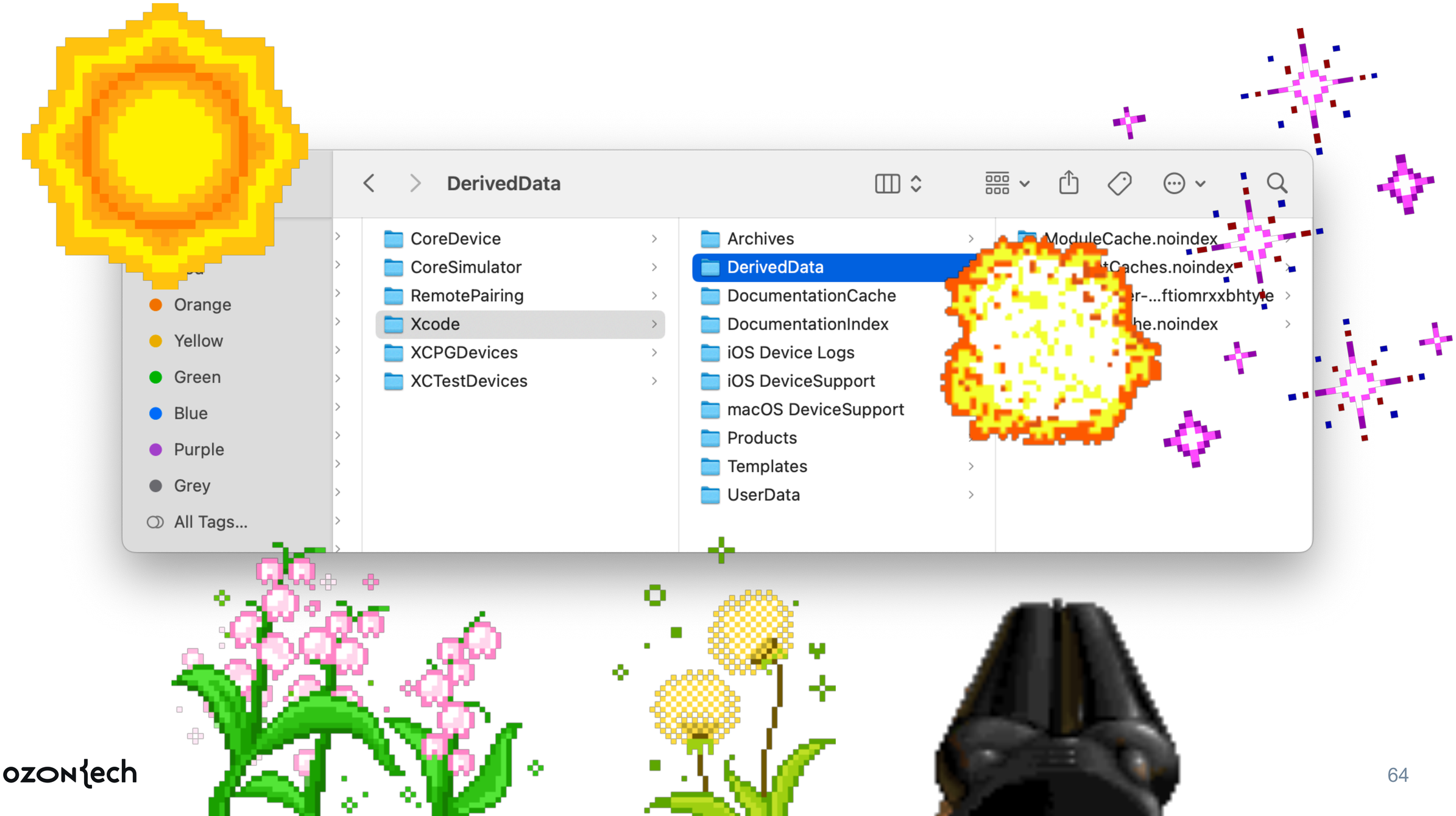
Машинное время билдов MP-a


Продолжительность, с.
360
70
690
260
Clean Build
Iterative Build
Resolve + Clean Build
Cached Clean Build


seller-app-ios — zsh — zsh (figterm) ▸ zsh — 68×14
...r-app-ios — zsh — zsh (figterm) ▸ zsh
...rnal — zsh — zsh (figterm) ▸ zsh
Last login: Sat Feb 24 23:21:44 on console
oiu@mbp-evgeryzhov-BYOD-WHH5P7YX2T seller-app-ios % make go
🛠️Закрытие Xcode
No matching processes belonging to you were found
💻 Установка git hooks
🔗 Простановка DI связей
🔧 Генерация ресурсов
🔨 Генерация проекта
Resolved cache profile 'Development' from Tuist's defaults
Generating workspace SellerCenter.xcworkspace
Generating project SellerCenter
Resolving package dependencies using xcodebuild
Project generated.
Total time taken: 334.816s


SellerCenter
tech/SCM-24745-xcframeworks-products
SellerCenter (Development F)
iPhone 15 Pro (17.0.1)
Build Succeeded | Today a
Build SellerCenter (Development Flow) - Log
SellerCenter (Development Flow) ⟩ Build ⟩ No Selection
Recent Build Timeline ⟩ No Selection
All
Recent
All Messages
All Issues
Errors Only
Export.
Filter
15s
30s
Prepare packages
Compile plug-ins and run any prebuild commands
Compile plug-in "StaticVersionPlugin" in package "spm-plugins" 0.1 seconds
Apply build tool plug-in "StaticVersionPlugin" to target "Messenger" in package "snapshots.Messenger"
Try to find Versions.xcconfig file at /Users/oiu/Library/Developer/Xcode/DerivedData/SellerCenter... more
Versions.xcconfig content:
["OZ_VERSION_NUMBER = 8.3.0"]
Calculated version: 8.3.0
Process build tool plug-in results 0.1 seconds
Run pre-actions
Build | Scheme SellerCenter (Development Flow)
Run custom shell script 'Run Sourcery and SwiftGen' 21.6 seconds
Run custom shell script 'Run Needle' 0.1 seconds
Run custom shell script 'Generate JsonFileNames' 0.1 seconds
Prepare build
Workspace SellerCenter | Scheme SellerCenter (Development Flow) | Destination iPhone 15 Pro
Compute target dependency graph 1.3 seconds
Building targets in dependency order
Target dependency graph (269 targets)
Gather provisioning inputs 0.1 seconds
Build stat cache for /Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/...
Create build directory $(OBJROOT) 0.1 seconds
Create build directory $(TARGET_BUILD_DIR) 0.1 seconds
Create build directory $(TARGET_BUILD_DIR)/Debug 0.1 seconds
Create build directory $(TARGET_BUILD_DIR)/Debug/PackageFrameworks 0.1 seconds
Create build directory $(TARGET_BUILD_DIR)/Debug-iphonesimulator 0.1 seconds
Create build directory $(TARGET_BUILD_DIR)/Debug-iphonesimulator/PackageFrameworks 0.1 sec...
Create build directory $(OBJROOT)/EagerLinkingTBDs/Debug-iphonesimulator 0.1 seconds
Build stat cache for /Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Develop...
Process OzonAddressKit.xcframework (iOS Simulator) 0.1 seconds
Process Mapbox.xcframework (iOS Simulator) 0.2 seconds
Process OzonMapsKit.xcframework (iOS Simulator) 0.2 seconds
Run custom shell script 'Run Sourcery and S...
Emitti...
Com...
Comp...
Emitting...

ozontech


DerivedData
Orange
Yellow
Green
Blue
Purple
Grey
All Tags...
CoreDevice
CoreSimulator
RemotePairing
Xcode
XCPGDevices
XCTestDevices
Archives
DerivedData
DocumentationCache
DocumentationIndex
iOS Device Logs
iOS DeviceSupport
macOS DeviceSupport
Products
Templates
UserData
ModuleCache.noindex

08

Но какой ценой?

Зачем всё это?

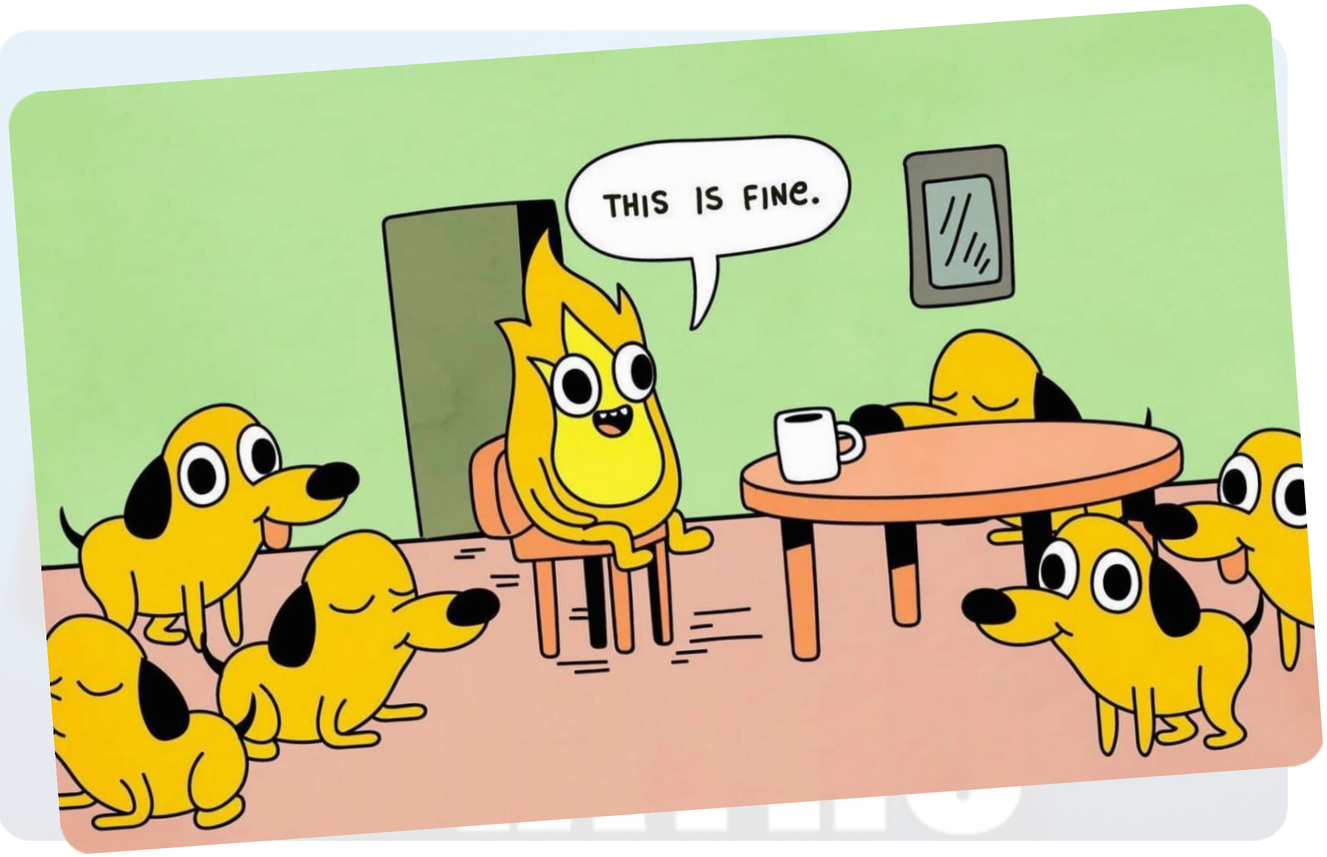
А главное — какой ценой

- **Затраты на разработку**

Зачем всё это?

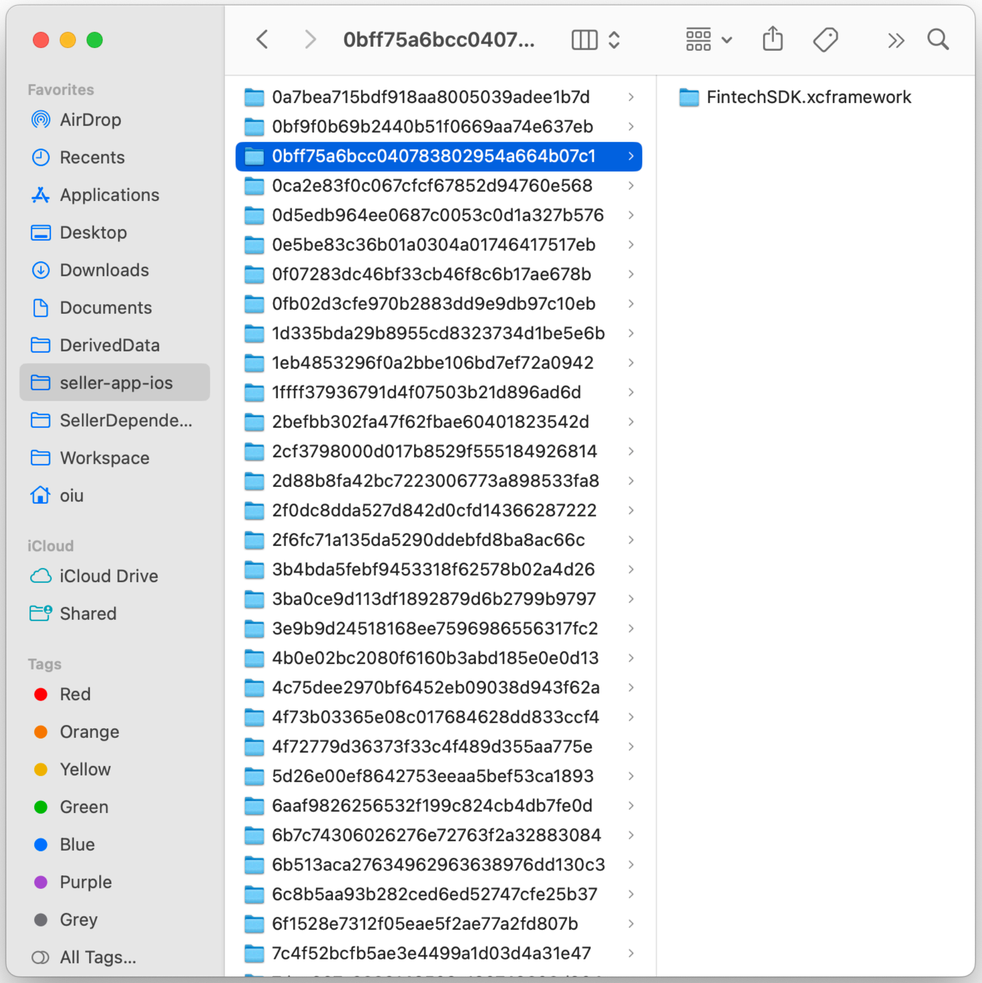
А главное — какой ценой

- Затраты на разработку
- Затраты на поддержку



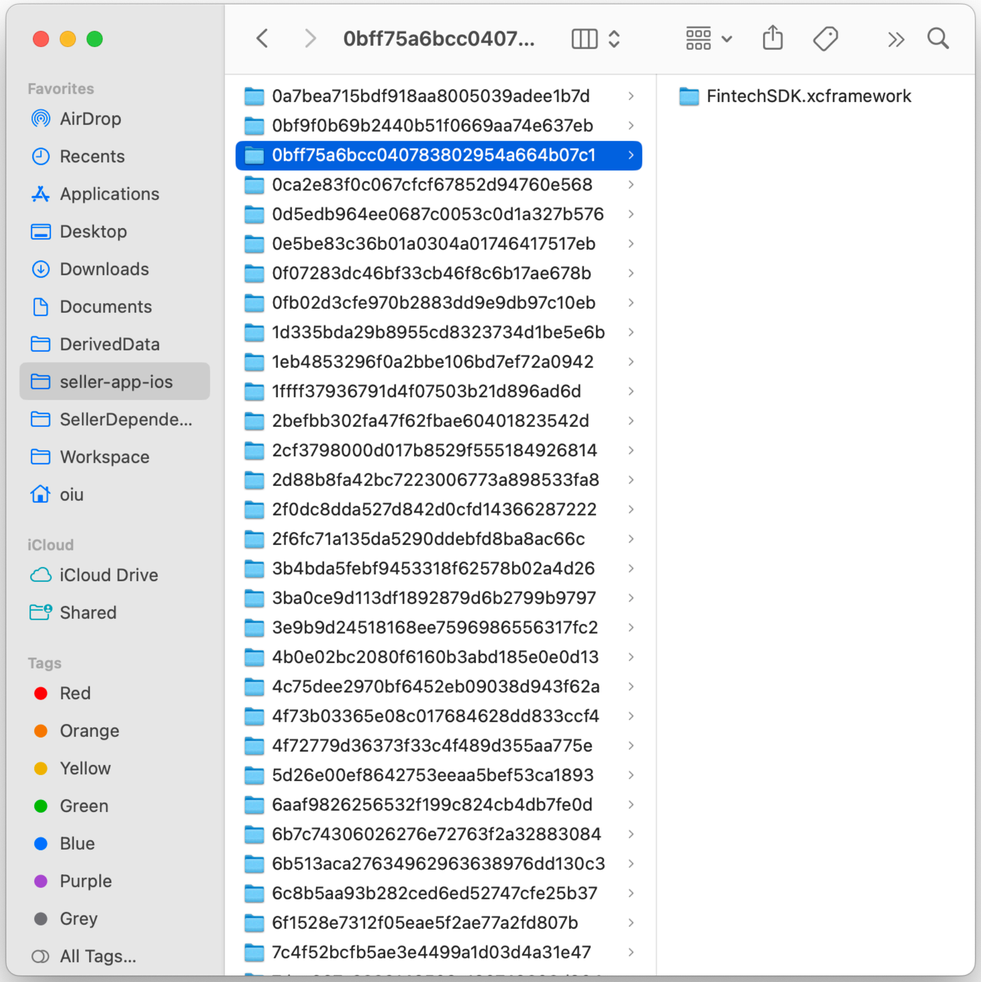
Хэши прогретых зависимостей

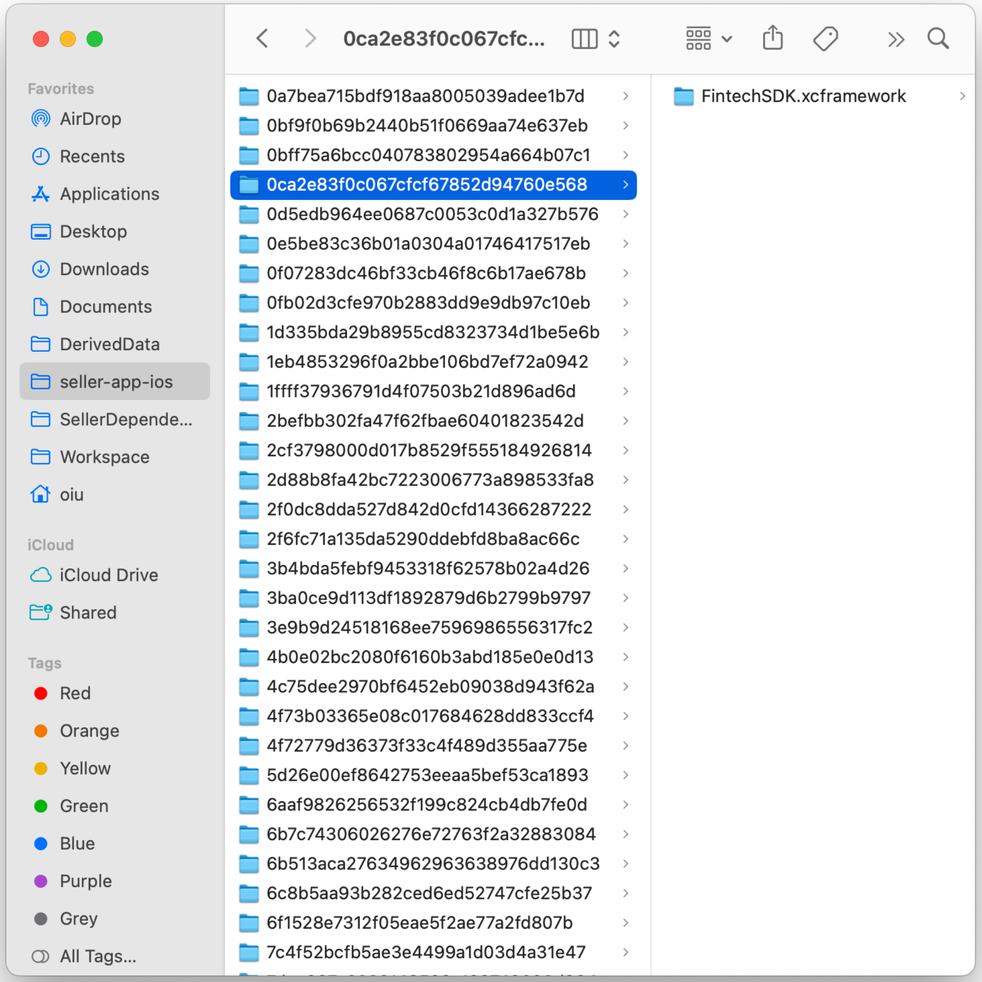
И их дубликаты



Хэши прогретых зависимостей

И их дубликаты





Хэши прогретых зависимостей

И их дубликаты

1. Изначальный коммит
2. Заводим “пакет с пакетами”
3. Обвязка Tuist`a — собираем фреймворки
4. Скрипты генерации
5. **Хэширование версий зависимостей**
6. Бандлы SPM-зависимостей
7. Генерация проекта исходниками

@oiuhr

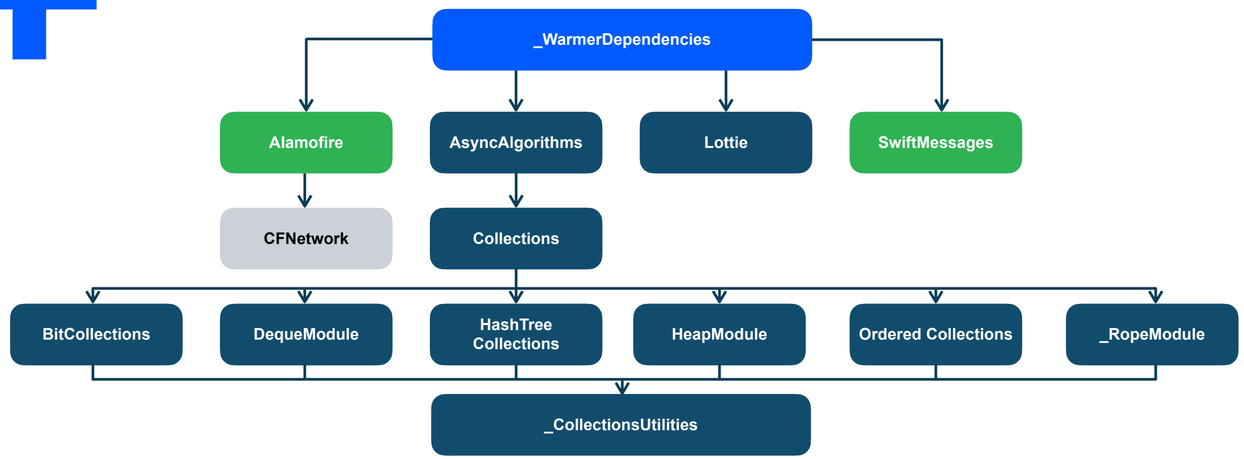
ozontech

Хэши прогретых зависимостей

И их дубликаты

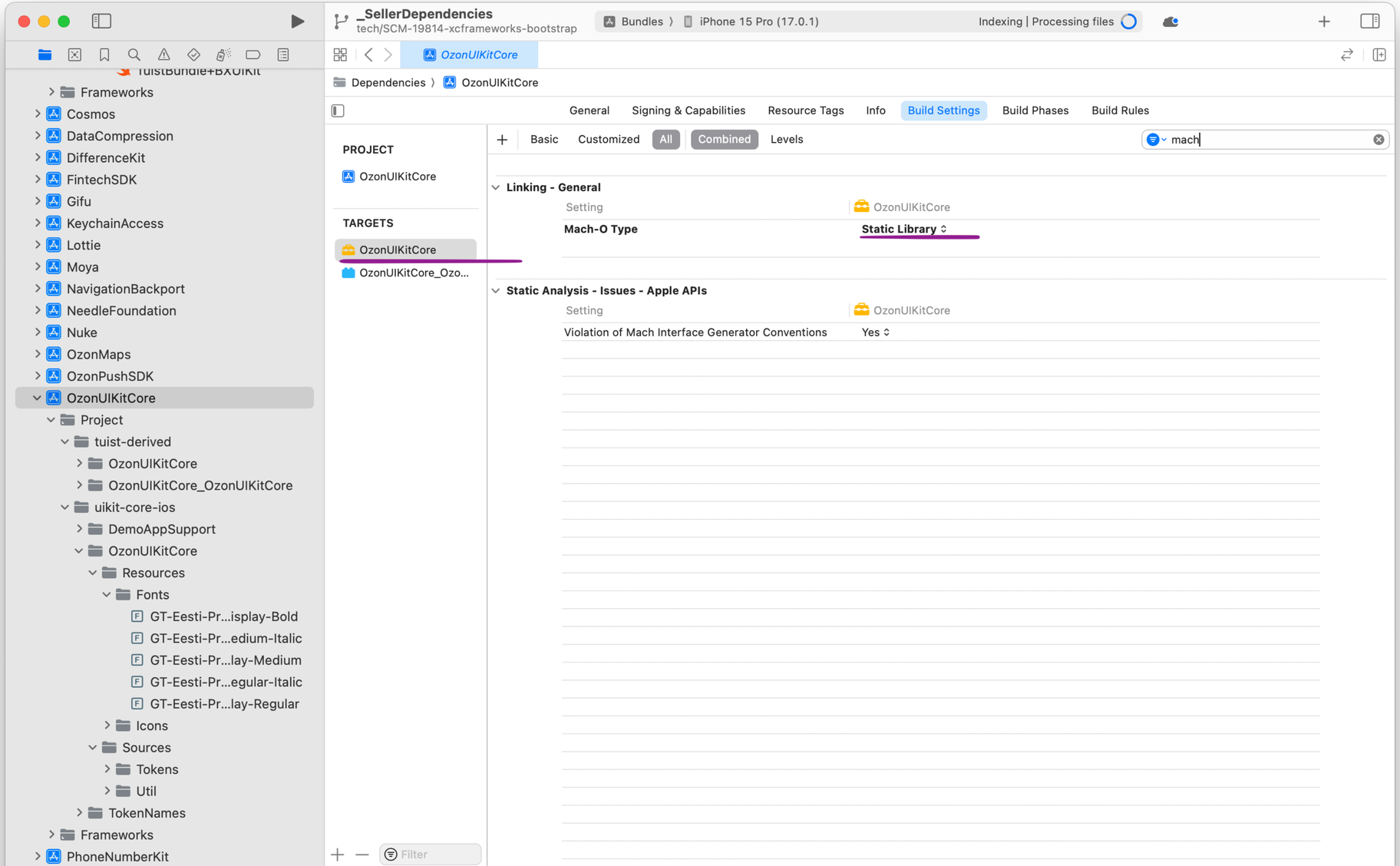
```
oiu@kalkulator .internal % tuist cache --print-hashes
Loading and constructing the graph
It might take a while if the cache is empty
Alamofire - c435572b0a7ad22c114f8d51a31ae436
AsyncAlgorithms - 720331ea3ae0fe949c993e79ce0189e8
BitCollections - 28a87950d21c2bd32a186e8af9b18ae3
Collections - 8092a7cf321e5baa360287afd327c627
DequeModule - 80aba64f69a2fabe9f88b498ce53a4ce
HashTreeCollections - fd1f6a880f83254e0eef63c6f05cc70b
HeapModule - c8e5b5a3ba7d0d90bf949a80ab4c6fa6
Lottie - 4de729d39b102705ae7e5cec6f575324
Lottie_Lottie - 5624029cc39cae4926d23ad86a16fec4
OrderedCollections - 65102715afc23f47960497675b59e8a1
SwiftMessages - df635f71fc370f397c7943196c0f3069
_CollectionsUtilities - 285fcd1cd13e3e5e246a86e7e03bb625
_RopeModule - 145ab29585d0c66ba147747e3319ba6f
Total time taken: 1.533s
```

```
oiu@kalkulator oiu % tuist graph
    // опционально: -f json
```



Бандлы сторонних проектов

А точнее — их отсутствие



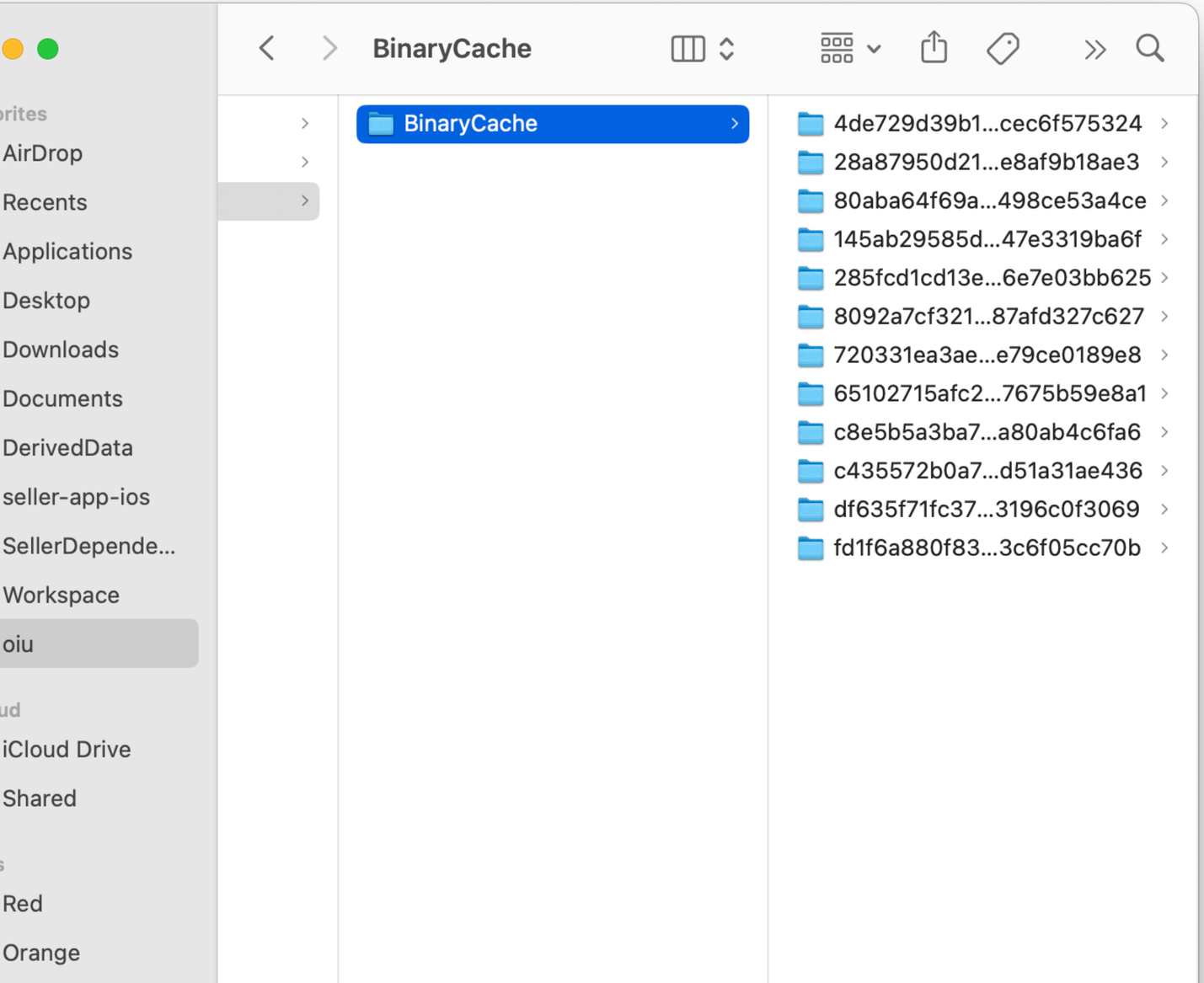
Бандлы сторонних проектов

А точнее — их отсутствие

```
oiu@kalkulator .internal % tuist cache --print-hashes
Loading and constructing the graph
It might take a while if the cache is empty
Alamofire - c435572b0a7ad22c114f8d51a31ae436
AsyncAlgorithms - 720331ea3ae0fe949c993e79ce0189e8
BitCollections - 28a87950d21c2bd32a186e8af9b18ae3
Collections - 8092a7cf321e5baa360287afd327c627
DequeModule - 80aba64f69a2fabe9f88b498ce53a4ce
HashTreeCollections - fd1f6a880f83254e0eef63c6f05cc70b
HeapModule - c8e5b5a3ba7d0d90bf949a80ab4c6fa6
Lottie - 4de729d39b102705ae7e5cec6f575324
Lottie_Lottie - 5624029cc39cae4926d23ad86a16fec4
OrderedCollections - 65102715afc23f4796049767 5b59e8a1
SwiftMessages - df635f71fc370f397c7943196c0f3069
_CollectionsUtilities - 285fcd1cd13e3e5e246a86e7e03bb625
_RopeModule - 145ab29585d0c66ba147747e3319ba6f
Total time taken: 1.533s
```

Бандлы сторонних проектов

А точнее — их отсутствие




```
oiu@kalkulator .internal % tuist cache
Loading and constructing the graph
It might take a while if the cache is
Alamofire - c435572b0a7ad22c114f8d51a3
AsyncAlgorithms - 720331ea3ae0fe949c99
BitCollections - 28a87950d21c2bd32a186
Collections - 8092a7cf321e5baa360287af
DequeModule - 80aba64f69a2fabe9f88b498
HashTreeCollections - fd1f6a880f83254e
HeapModule - c8e5b5a3ba7d0d90bf949a80a
Lottie - 4de729d39b102705ae7e5cec6f575
Lottie_Lottie - 5624029cc39cae4926d23a
OrderedCollections - 65102715afc23f479
SwiftMessages - df635f71fc370f397c7943
_CollectionsUtilities - 285fcd1cd13e3e
_RopeModule - 145ab29585d0c66ba147747e
Total time taken: 1.533s
```

Бандлы сторонних проектов

А точнее — их отсутствие

1. Изначальный коммит
2. Заводим “пакет с пакетами”
3. Обвязка Tuist`a — собираем фреймворки
4. Скрипты генерации
5. Хэширование версий зависимостей
6. **Бандлы SPM-зависимостей**
7. Генерация проекта исходниками

@oiuhr

ozontech

Артефакты линковки

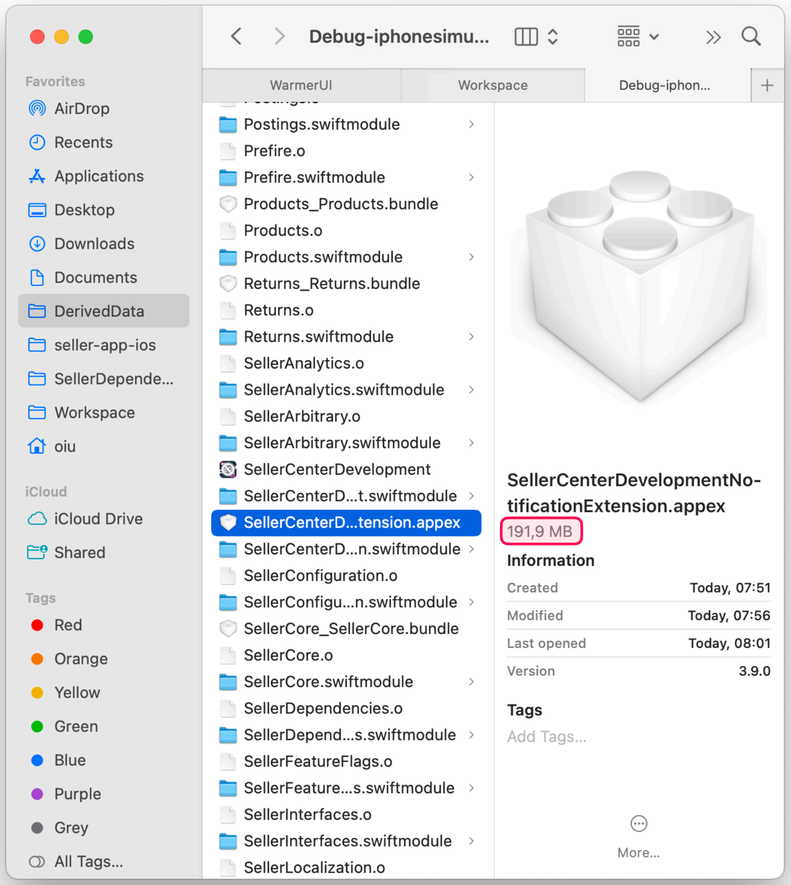
На этот раз действительно без костылей

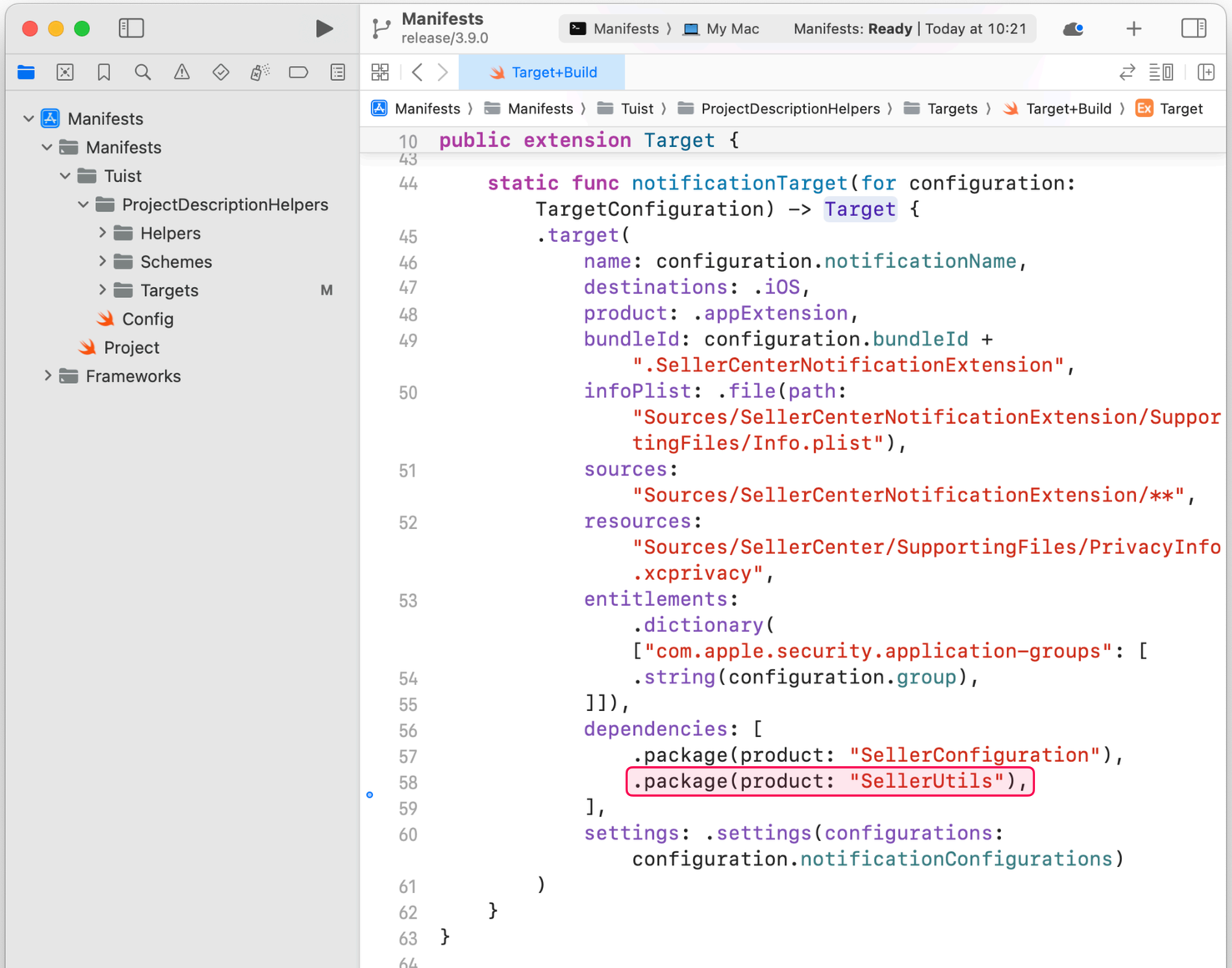
```
_ = UIApplicationMain(
    CommandLine.argc,
    UnsafeMutableRawPointer(C
        .bindMemory(to: UnsafeMutablePointer?.self, capacity: Int(CommandLine.argc)),
    nil,
    appDelegateClass
```

Thread 1: Fatal error: NSArray element failed to match the Swift Array Element type
Expected YMKSuggestItem but found YMKSuggestItem

Артефакты линковки

На этот раз действительно без костылей

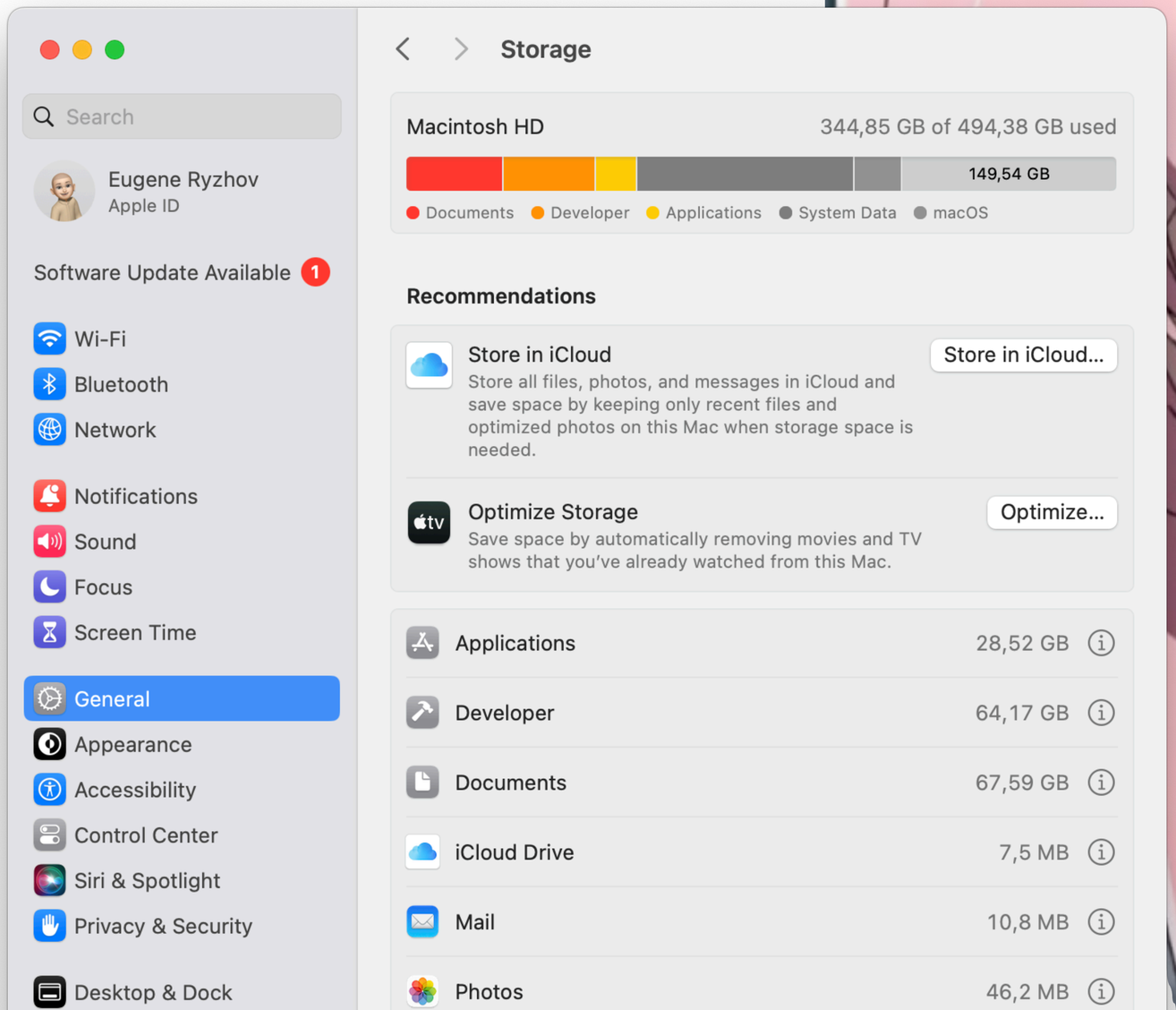


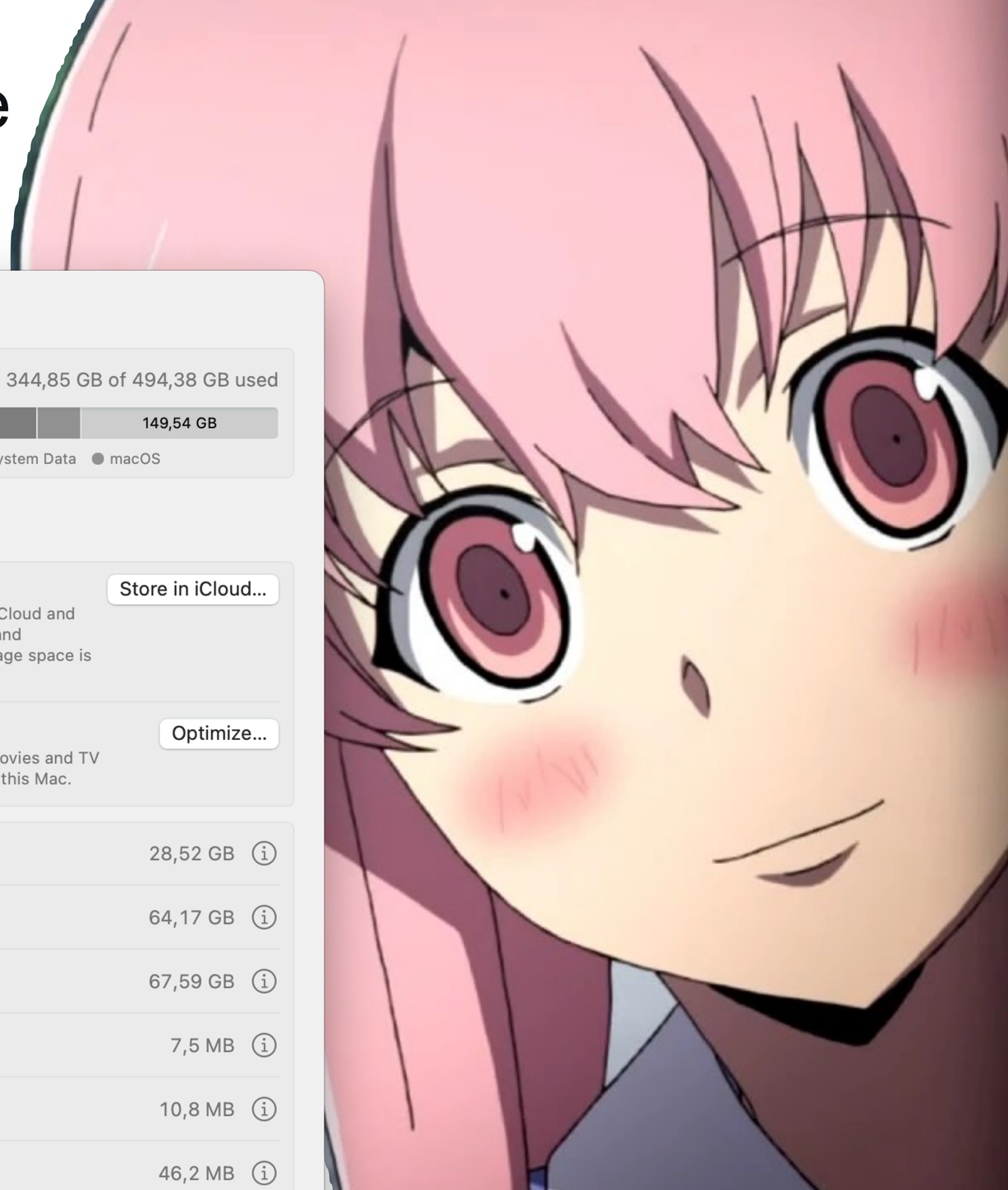


ozontech

Взрыв на макаронной фабрике

Место на машинках CI





ozontech

Генерация проекта исходниками

На семь бед — один ответ

```
1 f "SPM_WARMER_GENERATE_WITH_SOURCES" in os.environ:
2     print(
3         f'🥷 Поймано значение окружения для генерации из
  сурсов: используем внутренний манифест',
4         flush=True
5     )
6     subprocess.call(['cp', f'{manifest_path}', './'])
7     exit(0)
```

Генерация проекта исходниками

На семь бед — один ответ

1. Изначальный коммит
2. Заводим “пакет с пакетами”
3. Обвязка Tuist`a — собираем фреймворки
4. Скрипты генерации
5. Хэширование версий зависимостей
6. Бандлы SPM-зависимостей
7. **Генерация проекта исходниками**

@oiuhr

ozontech

Зачем всё это?

А главное — какой ценой

- Затраты на разработку
- Затраты на поддержку
- **Целесообразность написания собственного решения**

Зачем кастом?

Или почему не проприетарное решение?

Pros

- Кэширование бинарных артефактов из коробки

Cons

- Огромный пласт работ на переезд
- Очень частые обновления
  - Так себе документация
  - Стабильность релизов
- Закрытые (пейволлом) исходники Tuist Cloud

Зачем всё это?

А главное — какой ценой

- Затраты на разработку
- Затраты на поддержку
- **Целесообразность написания собственного решения**

09

Всё!

Сегодня ты узнал…

… что лучше бы выбрал кофе вместо прослушки доклада

01 Что билдтайм можно измерять
в граммах (зерна кофе) и часах (сна)

Сегодня ты узнал…

… что лучше бы выбрал кофе вместо прослушки доклада

01 Что билдтайм можно измерять в граммах (зерна кофе) и часах (сна)

02 Что SPM, в общем-то, ничего не умеет

Сегодня ты узнал…

… что лучше бы выбрал кофе вместо прослушки доклада

01 Что билдтайм можно измерять в граммах (зерна кофе) и часах (сна)

02 Что SPM, в общем-то, ничего не умеет

03 Как использовать готовый инструмент в виде костыля

Сегодня ты узнал…

… что лучше бы выбрал кофе вместо прослушки доклада

01 Что билдтайм можно измерять в граммах (зерна кофе) и часах (сна)

02 Что SPM, в общем-то, ничего не умеет

03 Как использовать готовый инструмент в виде костыля

67 Сборку SPM-проекта с использованием сурсов и бинарных артефактов

Сегодня ты узнал…

… что лучше бы выбрал кофе вместо прослушки доклада

01 Что билдтайм можно измерять в граммах (зерна кофе) и часах (сна)

02 Что SPM, в общем-то, ничего не умеет

03 Как использовать готовый инструмент в виде костыля

67 Сборку SPM-проекта с использованием сурсов и бинарных артефактов

68 Варианты подключения бинарных артефактов к фичевым пакетам с помощью собственного инструмента

Сегодня ты узнал…

… что лучше бы выбрал кофе вместо прослушки доклада

01 Что билдтайм можно измерять в граммах (зерна кофе) и часах (сна)

02 Что SPM, в общем-то, ничего не умеет

03 Как использовать готовый инструмент в виде костыля

67 Сборку SPM-проекта с использованием сурсов и бинарных артефактов

68 Варианты подключения бинарных артефактов к фичевым пакетам с помощью собственного инструмента

69 Результаты нашего аттракциона, включая всю боль

Что дальше?

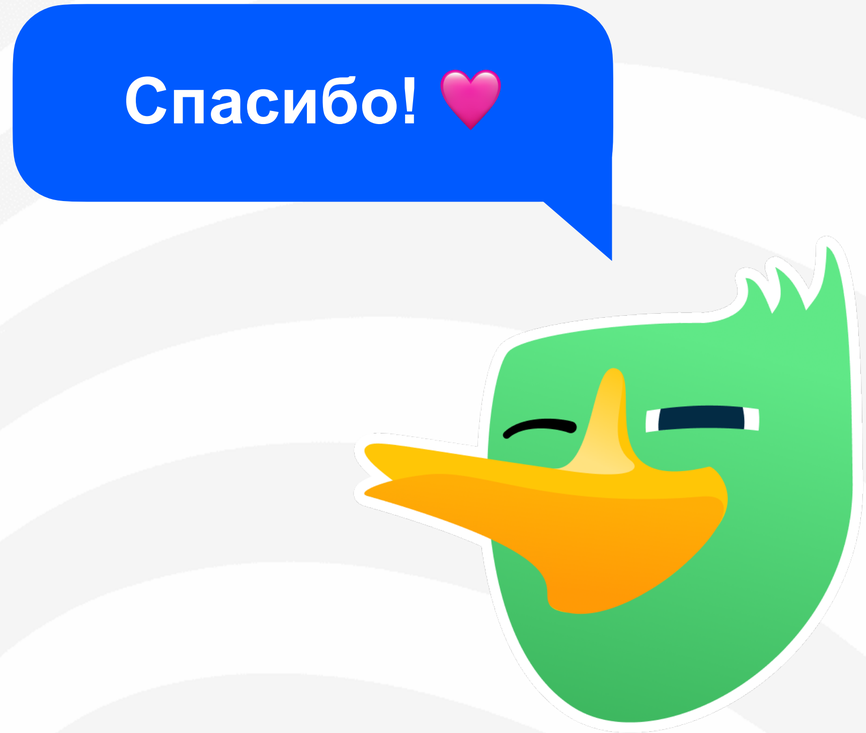
Наш роадмап

- ~~Разнести всё из одного пакета по разным (вес, инкапсуляция)~~ **Сделано за время подготовки!**
- ~~Убить постоянный ребилд кэша~~ **Сделано за время подготовки!**
- **Remote build cache**
- **Кэширование локальных фичевых пакетов**
- **Интерфейс с кнопочками**
- **???**

ozontech

OZON tech

А каков **ваш** роадмап?



Евгений Рыжов, Старший iOS разработчик

@oiuhr