Building automated modular Xcode projects

Pedro Piñera - @pepibumur

shopify

Pedro Piñera

Production Engineer at Shopify

@pepibumur

ppinera.es

pedro.pinera@shopify.com

It’s a workshop

- **Stop me:**
  - If you got lost.
  - If you can’t follow it.
  - If you want to ask something.
  - If you would like to add something.
  - If you spot the bug I’m struggling to find.
- **How:**
  - Simple: raise your hand ✋

Let’s start

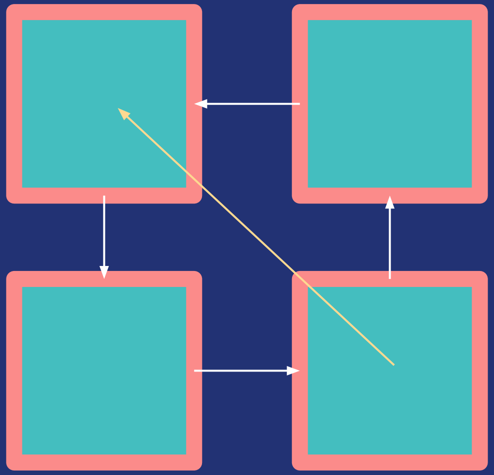
Are you ready?

Modular apps

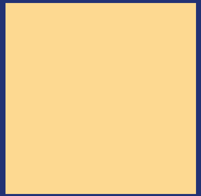
Pedro,

why do I need build my app in modules?

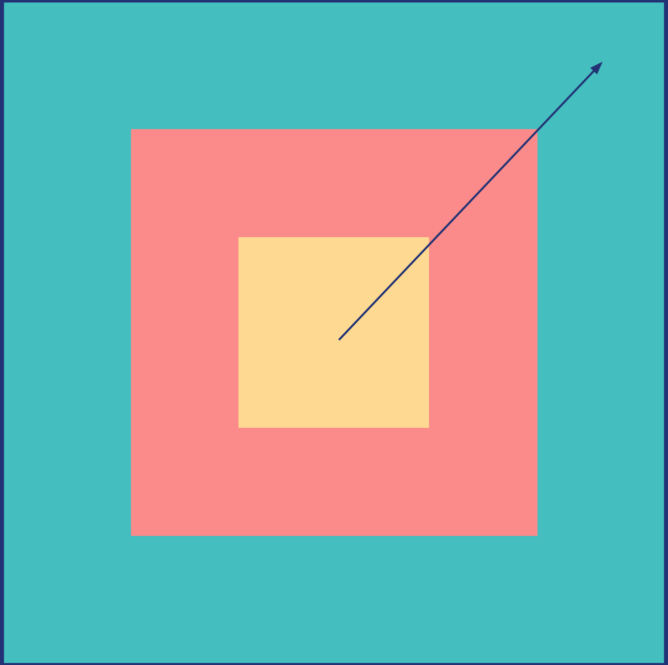
Decoupled code



Faster iterations

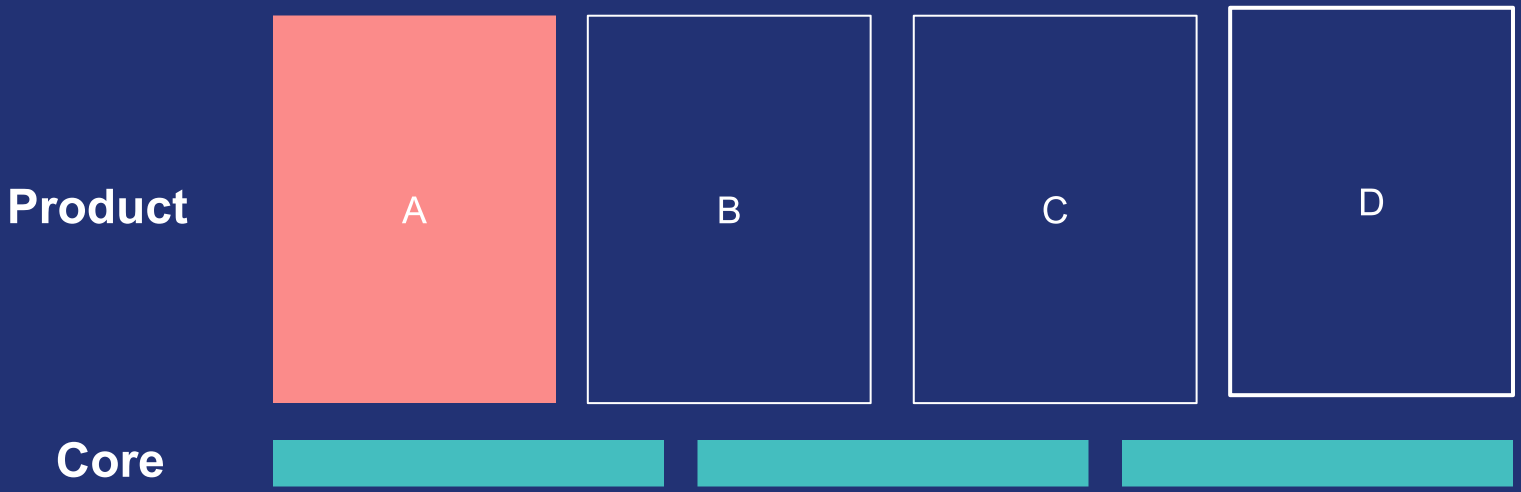


Faster iterations



⌘ + K
Clean

Faster iterations



When should I modularize?

- My project is growing and the build time is having a negative impact on developers **productivity**.
- My company has more than one app and there's a clear need of **sharing code**.
- I'm a freelancer and I want to share code across the projects that I work on.
- I'm a **clean-code** devotee and I'd like my code to be as decoupled as possible.

Xcode quiz

Library/Framework?

Xcode quiz

Static/Dynamic?

Xcode quiz

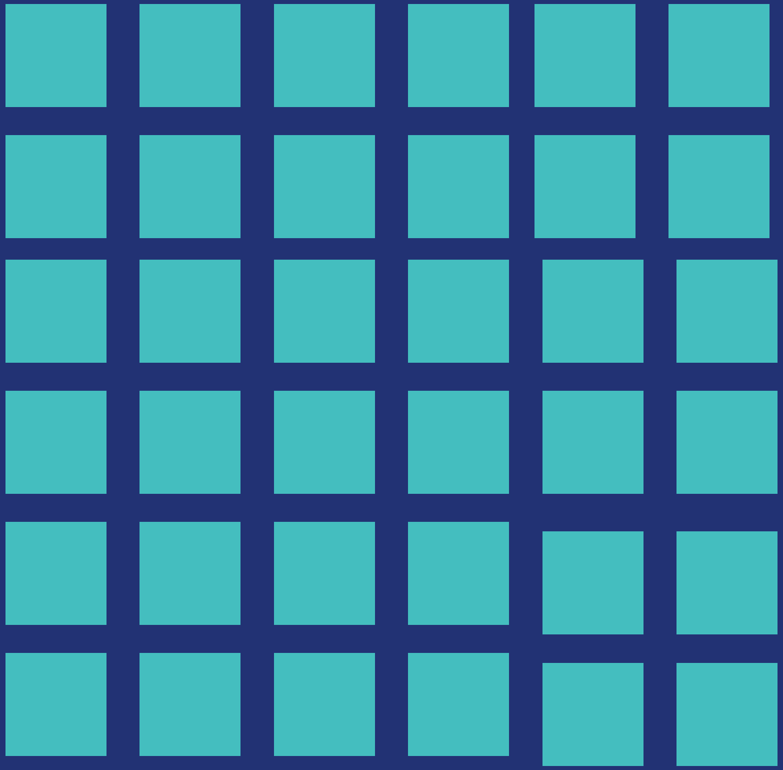
How can we define library/frameworks linking?

Xcode quiz

When is embedding frameworks necessary?

There’s a little downside...

maintaining multiple Xcode projects and targets is very cumbersome



Build settings

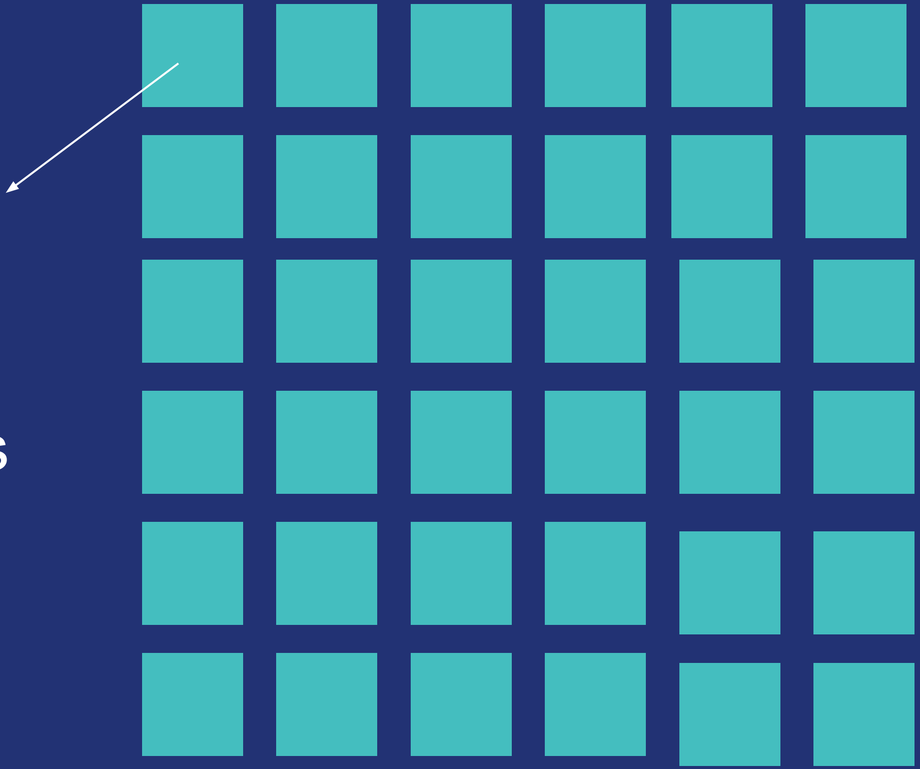
Schemes

Targets

Resources

Dependencies

...



Don’t worry □

After the workshop, it won’t be an issue anymore

Workshop’s goals

- Have a better understanding of a modular architecture *(linking, embedding, frameworks, library, static & dynamic)*.
- Automate the generation of Xcode projects using XcodeGen.
- Share code across multiple products.
- Automate tasks in Swift using Sake.

Modularization

Project Generation

Automation using Swift

Are you ready? 🚀

Let’s get our hands dirty

Before continuing...

- ✓ Do you have Xcode 9.2 installed?
- ✓ Do you have Carthage installed?
- ✓ Your Ruby environment is working and has Bundler installed?
- ✓ XcodeGen is installed: *github.com/yonaskolb/xcodegen*
- ✓ Sake is installed: *github.com/xcodeswift/sake*

Clone repository

github.com/pepibumur/xcode-modular-apps-workshop

git checkout 0.1.0

#1
Getting the project running

Before continuing...

- ✓ Your app compiles
- ✓ You are able to login using GitHub.
- ✓ You see the issues assigned to you on the screen.

#2
Automating the generation of the GitHubKit project

Before continuing...

- ✓ GitHubKit project is not part of the git repository anymore
- ✓ GitHubKit scheme compiles and the tests run.
- ✓ Issues app compiles and runs on the simulator.

Git tag: 0.2.0

#3
Automating the generation of the IssuesKit project

Before continuing...

- ✓ IssuesKit project is not part of the git repository anymore
- ✓ IssuesKit scheme compiles and the tests run.
- ✓ Issues app compiles and runs on the simulator.

Git tag: 0.3.0

#4 Automating the generation of the App project

Before continuing...

- ✓ App project is not part of the git repository anymore
- ✓ App scheme compiles and the tests run

Git tag: 0.4.0

#5
Creating an
iMessage app

Before continuing...

- ✓ iOS app should compile and run.
- ✓ iMessage extension should compile and run.

Git tag: 0.5.0

#6
Creating
IssuesUI framework

Before continuing...

- ✓ iOS app should compile and run (with the IssueCell in the new IssueView)
- ✓ iMessage extension should compile and run.

Git tag: 0.6.0

#7
Creating
ShareIssue view controller

Before continuing...

- ✓ The iMessage extension should compile and run and you should be able to share issues from the list.

Git tag: 0.7.0

#8
From Rake to Sake

Thanks!

shopify

shopify