

#WWDC18

# What's New in Testing

Session 403

Honza Dvorsky, Xcode Engineer  
Ethan Vaughan, Xcode Engineer

Code coverage

Test selection and ordering

Parallel testing

Code coverage

Test selection and ordering

Parallel testing

# Code Coverage

9.3

# Code Coverage

9.3

Xcode 9.3

# Code Coverage

9.3

Xcode 9.3

Performance and accuracy

# Code Coverage

9.3

Xcode 9.3

Performance and accuracy

Target selection

# Code Coverage

9.3

Xcode 9.3

Performance and accuracy

Target selection

xccov



# Code Coverage

9.3

Xcode 9.3

Performance and accuracy

Target selection

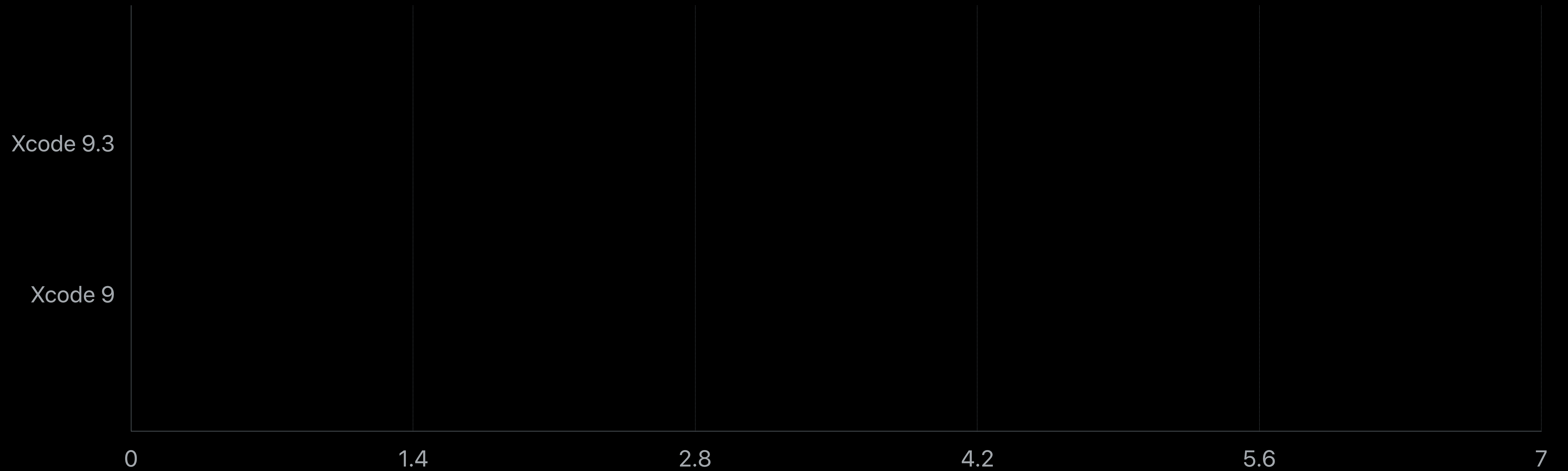
xccov

Source editor

# Performance and Accuracy

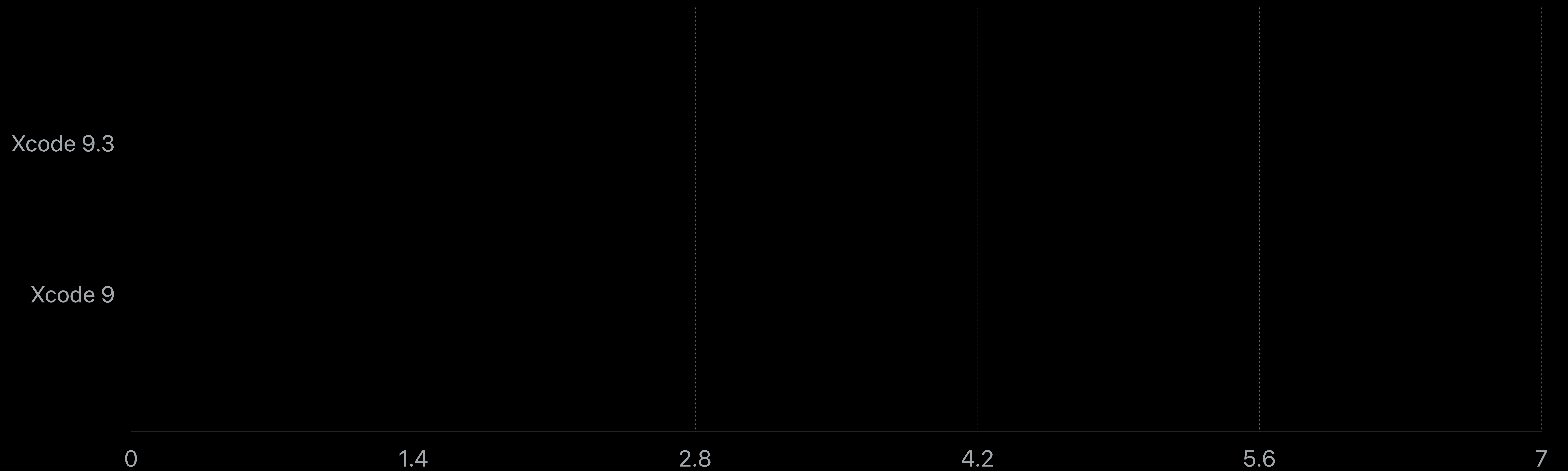
# Time to Load

In seconds



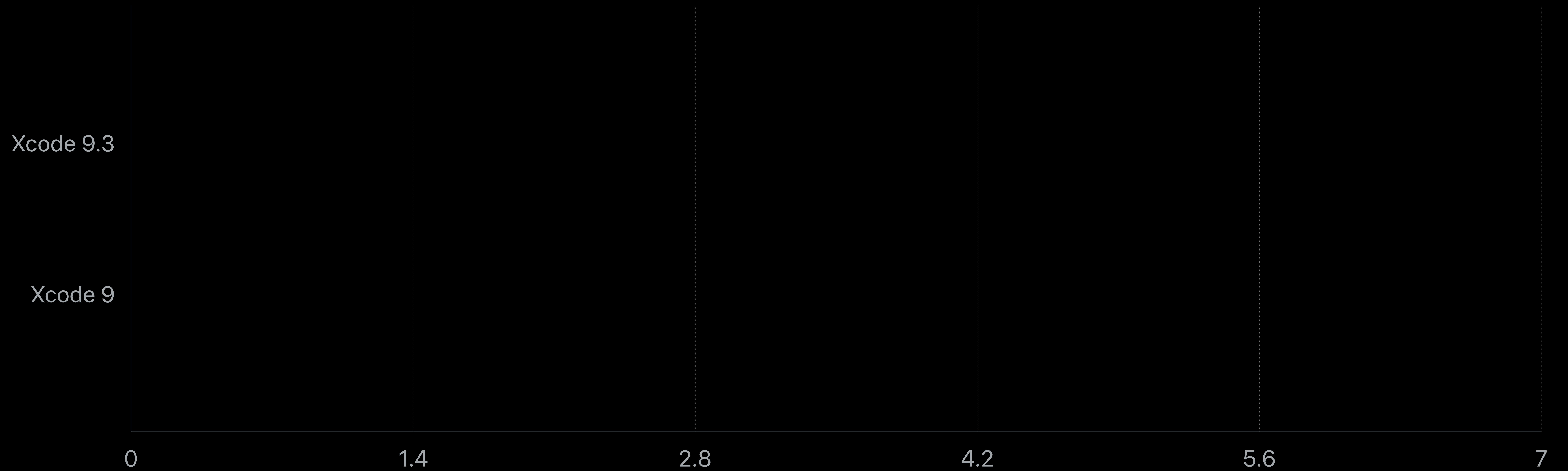
# Time to Load

In seconds



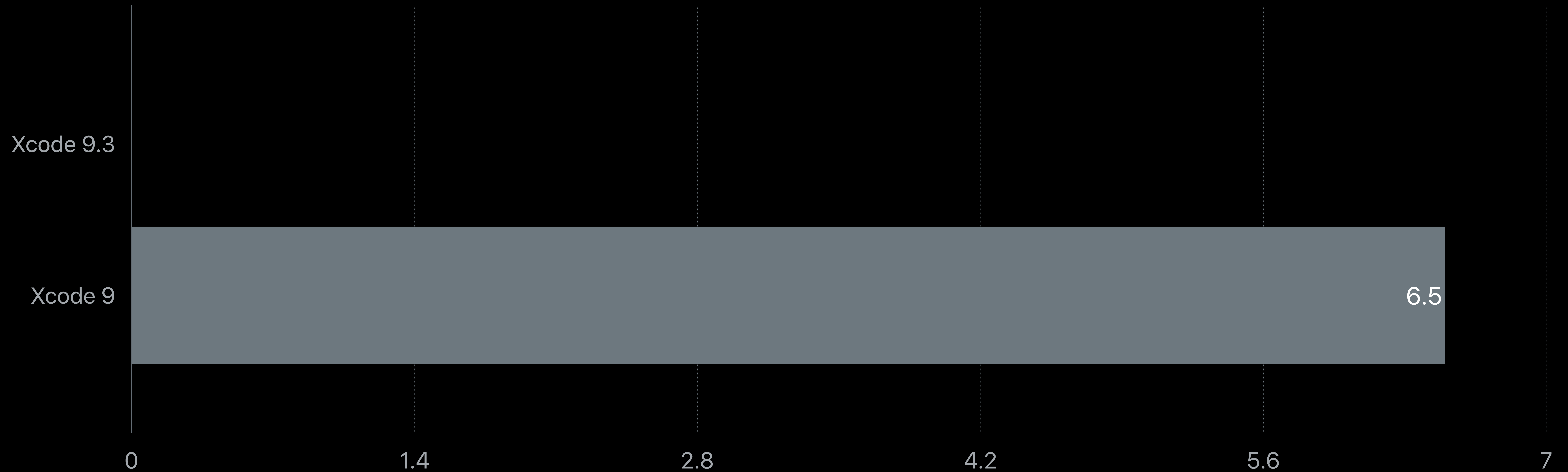
# Time to Load

In seconds



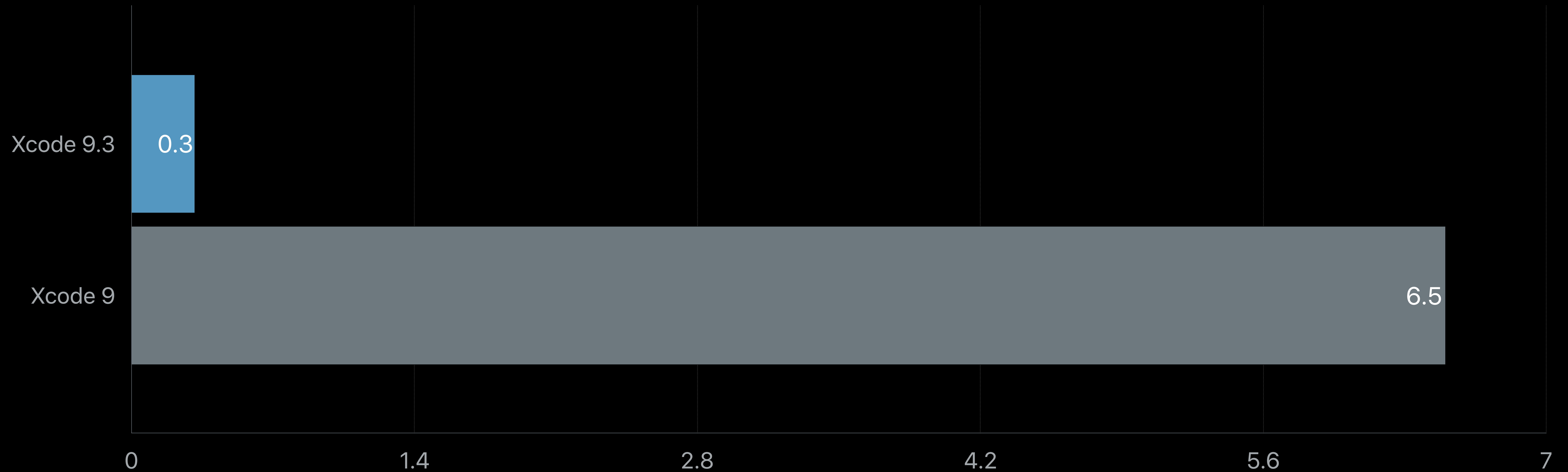
# Time to Load

In seconds



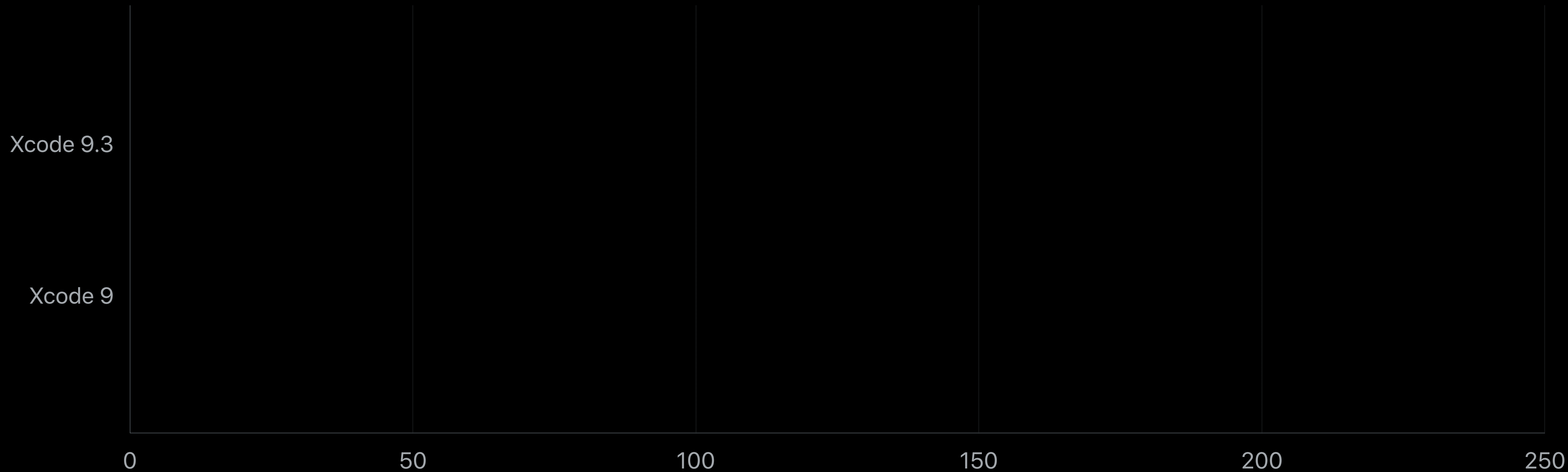
# Time to Load

In seconds



# File Size

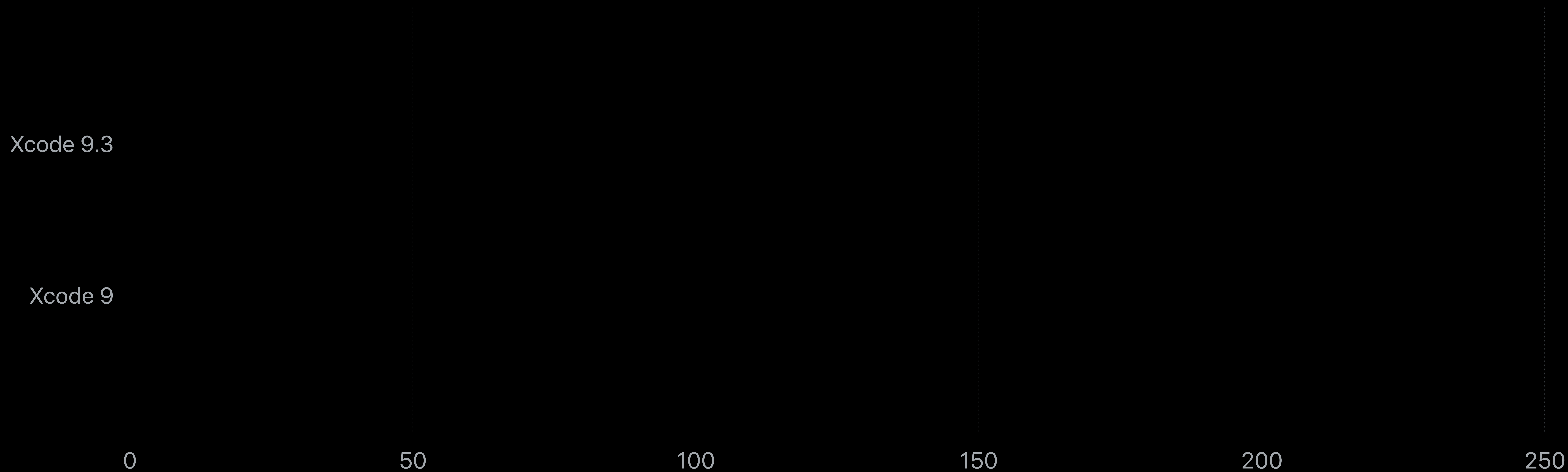
In MB





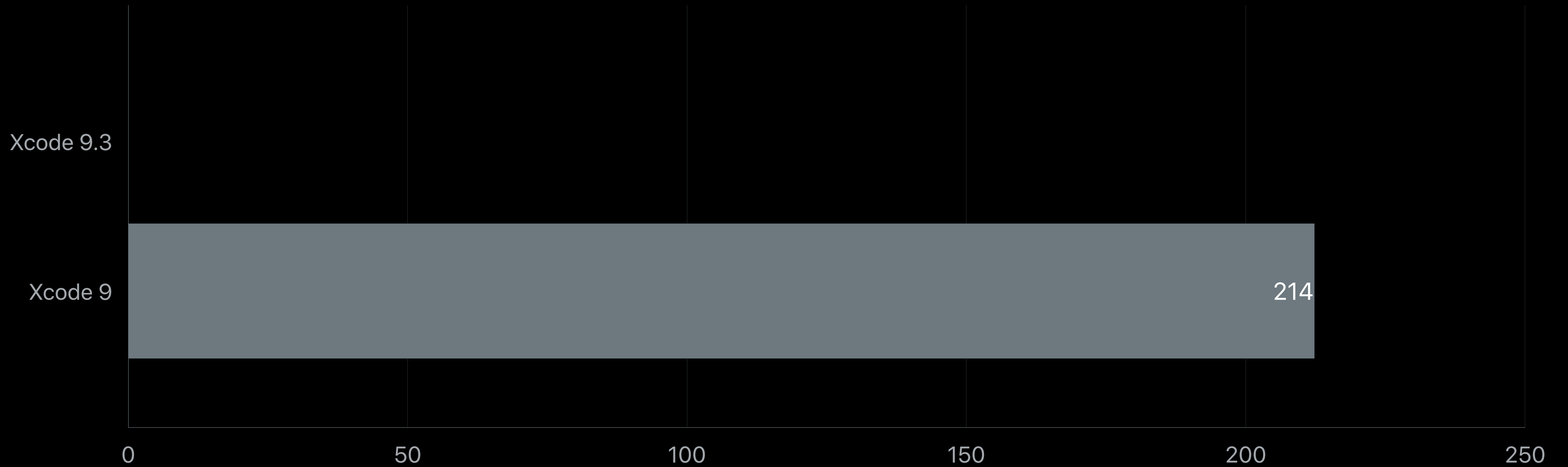
# File Size

In MB



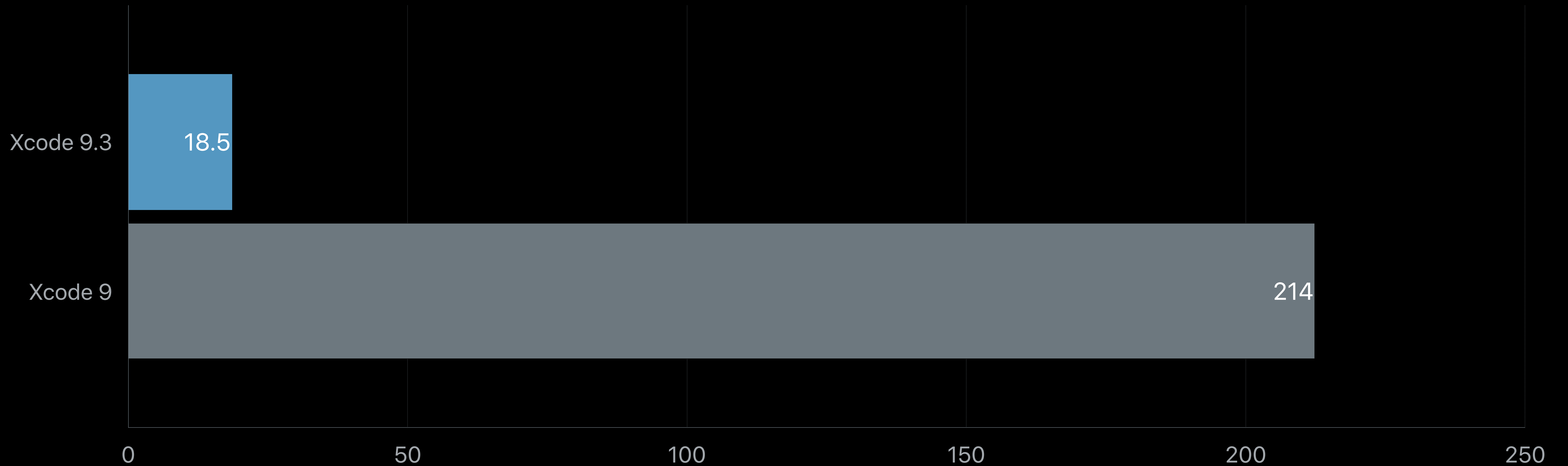
# File Size

In MB



# File Size

In MB



# Xcode 9

## C++ header

9.0

```
2  class Store {
3  private:
4      int count;
5
6  public:
7      void increment() {
8          count += 1;
9      }
10
11     int currentCount() {
12         return count;
13     }
14 };
```



# Xcode 9.3

## C++ header

9.3

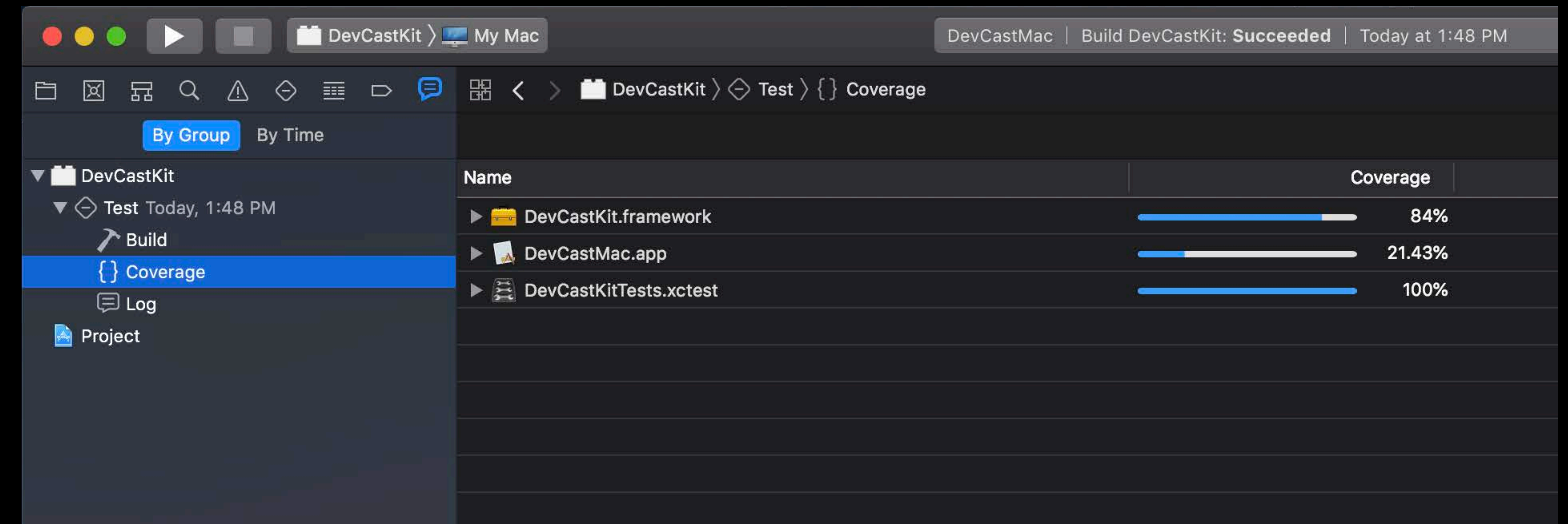
```
2  class Store {
3  private:
4      int count;
5
6  public:
7      void increment() {
8          count += 1;
9      }
10
11     int currentCount() {
12         return count;
13     }
14 };
```

# Target Selection

# Target Selection

## Code Coverage

- Enabled for all targets
- Enabled for selected targets
- Disabled





DevCastMac | Build DevCastKit: Succeeded | Today at 9:41 PM

DevCastMac > My Mac

DevCastMac

- DevCastMac
- DevCastPhone
- DevCastKit
  - Server.swift
  - Storage.swift
  - Message.swift
  - Thread.swift
  - User.swift
  - Resources
  - DevCastKitTests
  - Products
  - Frameworks

Name

- DevCastKit
- DevCastMac
- DevCastKitT

Build 1 target

- Run Debug
- Test Debug**
- Profile Release
- Analyze Debug
- Archive Release

Options

Application Language System Language

Application Region System Region

UI Testing
 

- Capture screenshots automatically
- Delete when each test succeeds

Attachments
 

- Delete when each test succeeds

Code Coverage
 

- Gather coverage for some targets

Targets

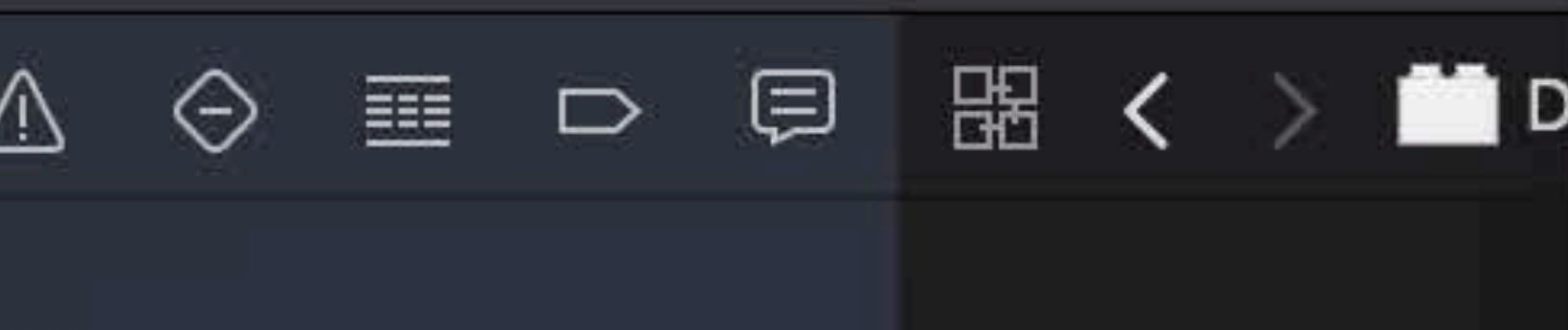
- DevCastKit

Filter

Duplicate Scheme Manage Schemes...  Shared Close Test

Filter





- Name
- ▶ DevCastKit.f
- ▶ DevCastMac
- ▶ DevCastKitT

DevCastKit > My Mac

Build 1 target

Run Debug

**Test** Debug

Profile Release

Analyze Debug

Archive Release

Info Arguments **Options** Diagnostics

Application Language System Language

Application Region System Region

UI Testing  Capture screenshots automatically  
 Delete when each test succeeds

Attachments  Delete when each test succeeds

Code Coverage  Gather coverage for some targets

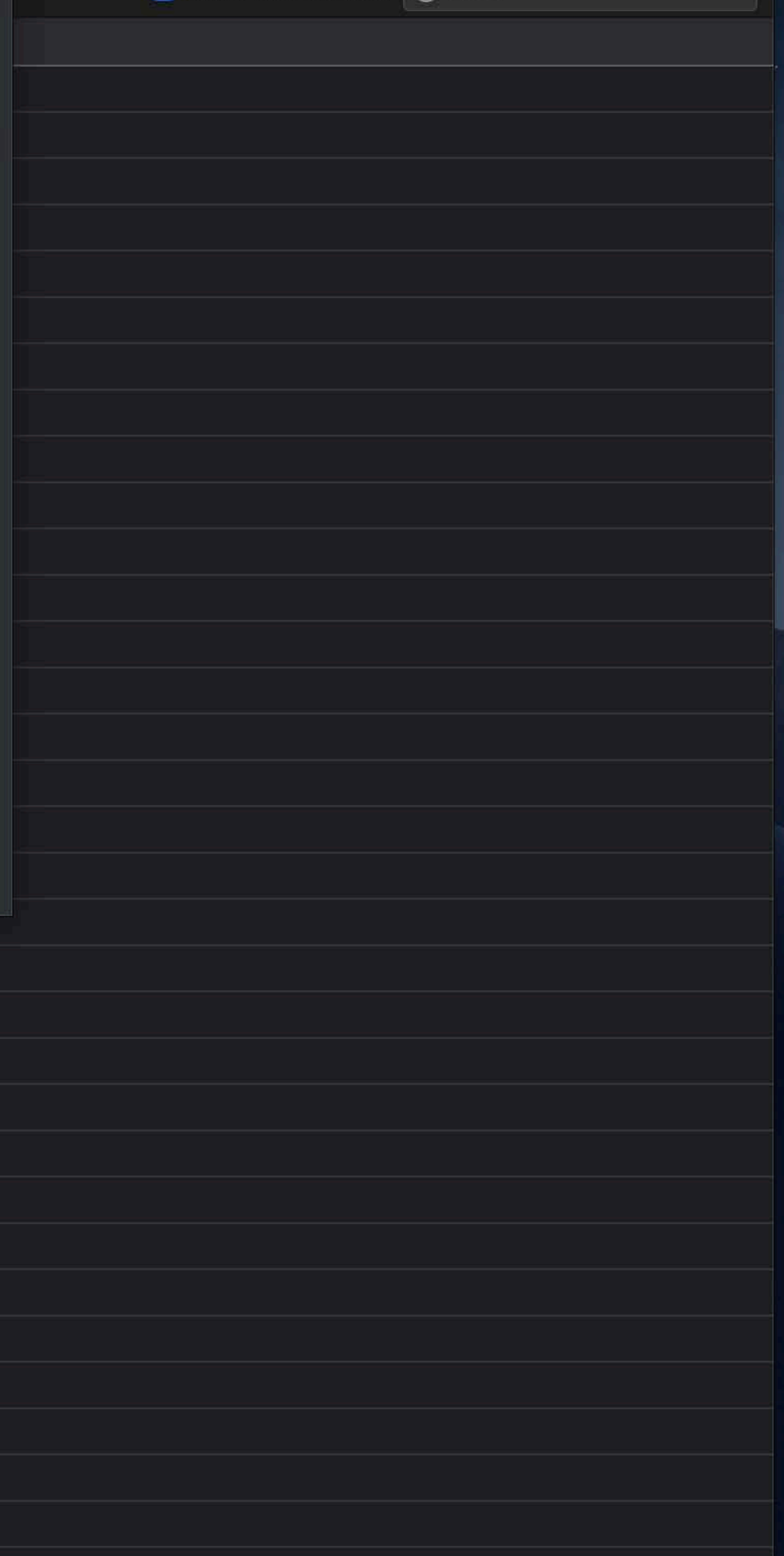
Targets

- DevCastKit

+ - Filter

Duplicate Scheme Manage Schemes...  Shared Close Test

Show Test Bundles Filter





- Name
- ▶ DevCastKit.f
- ▶ DevCastMac
- ▶ DevCastKitT

- ▶ Build  
1 target
- ▶ Run  
Debug
- ▶ **Test**  
Debug
- ▶ Profile  
Release
- ▶ Analyze  
Debug
- ▶ Archive  
Release

Info Arguments **Options** Diagnostics

Application Language System Language

Application Region System Region

UI Testing  Capture screenshots automatically  
 Delete when each test succeeds

Attachments  Delete when each test succeeds

Code Coverage  Gather coverage for some targets

Targets

- DevCastKit

+ - Filter

**XCCOV**

# XCCOV

Command line tool

Output formats

- Human-readable
- Machine-parseable (JSON)

View coverage data

# Coverage Data





# Coverage Data



Coverage report

Line coverage percentages

# Coverage Data



# Coverage Data



# Coverage Data

Derived data

# Coverage Data

Derived data

Result bundle

- `xcodebuild -resultBundlePath ...`

```
xcrun xccov view HelloWorld.xccovreport
```

Name	Coverage
-----	-----
HelloWorld.app	57.89% (11/19)
/tmp/HelloWorld/HelloWorld/AppDelegate.m	50.00% (3/6)
-[AppDelegate applicationWillTerminate:]	0.00% (0/3)
-[AppDelegate applicationDidFinishLaunching:]	100.00% (3/3)
/tmp/HelloWorld/HelloWorld/ViewController.m	50.00% (5/10)
-[ViewController setRepresentedObject:]	0.00% (0/5)
-[ViewController viewDidLoad]	100.00% (5/5)
/tmp/HelloWorld/HelloWorld/main.m	100.00% (3/3)
main	100.00% (3/3)
HelloWorldTests.xctest	100.00% (21/21)
/tmp/HelloWorld/HelloWorldTests/HelloWorldTests.m	100.00% (21/21)
-[HelloWorldTests testExample]	100.00% (4/4)

```
xcrun xccov view HelloWorld.xccovreport
```

Name	Coverage
-----	-----
HelloWorld.app	57.89% (11/19)
/tmp/HelloWorld/HelloWorld/AppDelegate.m	50.00% (3/6)
-[AppDelegate applicationWillTerminate:]	0.00% (0/3)
-[AppDelegate applicationDidFinishLaunching:]	100.00% (3/3)
/tmp/HelloWorld/HelloWorld/ViewController.m	50.00% (5/10)
-[ViewController setRepresentedObject:]	0.00% (0/5)
-[ViewController viewDidLoad]	100.00% (5/5)
/tmp/HelloWorld/HelloWorld/main.m	100.00% (3/3)
main	100.00% (3/3)
HelloWorldTests.xctest	100.00% (21/21)
/tmp/HelloWorld/HelloWorldTests/HelloWorldTests.m	100.00% (21/21)
-[HelloWorldTests testExample]	100.00% (4/4)



```
xcrun xccov view HelloWorld.xccovreport
```

Name	Coverage
-----	-----
HelloWorld.app	57.89% (11/19)
/tmp/HelloWorld/HelloWorld/AppDelegate.m	50.00% (3/6)
-[AppDelegate applicationWillTerminate:]	0.00% (0/3)
-[AppDelegate applicationDidFinishLaunching:]	100.00% (3/3)
/tmp/HelloWorld/HelloWorld/ViewController.m	50.00% (5/10)
-[ViewController setRepresentedObject:]	0.00% (0/5)
-[ViewController viewDidLoad]	100.00% (5/5)
/tmp/HelloWorld/HelloWorld/main.m	100.00% (3/3)
main	100.00% (3/3)
HelloWorldTests.xctest	100.00% (21/21)
/tmp/HelloWorld/HelloWorldTests/HelloWorldTests.m	100.00% (21/21)
-[HelloWorldTests testExample]	100.00% (4/4)

```
xcrun xccov view HelloWorld.xccovreport
```

Name	Coverage
-----	-----
HelloWorld.app	57.89% (11/19)
/tmp/HelloWorld/HelloWorld/AppDelegate.m	50.00% (3/6)
-[AppDelegate applicationWillTerminate:]	0.00% (0/3)
-[AppDelegate applicationDidFinishLaunching:]	100.00% (3/3)
/tmp/HelloWorld/HelloWorld/ViewController.m	50.00% (5/10)
-[ViewController setRepresentedObject:]	0.00% (0/5)
-[ViewController viewDidLoad]	100.00% (5/5)
/tmp/HelloWorld/HelloWorld/main.m	100.00% (3/3)
main	100.00% (3/3)
HelloWorldTests.xctest	100.00% (21/21)
/tmp/HelloWorld/HelloWorldTests/HelloWorldTests.m	100.00% (21/21)
-[HelloWorldTests testExample]	100.00% (4/4)

```
xcrun xccov view --json HelloWorld.xccovreport
```

```
{  
  "coveredLines": 32,  
  "lineCoverage": 0.8,  
  "targets": [  
    {  
      "coveredLines": 11,  
      "lineCoverage": 0.57894736842105265,  
      "files": [  
        {  
          "coveredLines": 3,  
          "lineCoverage": 0.5,  
          "path": "tmpHelloWorldHelloWorldAppDelegate.m",  
          "functions": [  
            {  
              "coveredLines": 0,  
              "lineCoverage": 0,  
              "path": "tmpHelloWorldHelloWorldAppDelegate.m",  
              "start": 1,  
              "end": 1,  
              "name": "main",  
              "signature": "int main(int argc, char * argv[])",  
              "type": "function"  
            }  
          ]  
        }  
      ]  
    }  
  ]  
}
```

```
xcrun xccov view --json HelloWorld.xccovreport
```

```
{
  "coveredLines": 32,
  "lineCoverage": 0.8,
  "targets": [
    {
      "coveredLines": 11,
      "lineCoverage": 0.57894736842105265,
      "files": [
        {
          "coveredLines": 3,
          "lineCoverage": 0.5,
          "path": "tmpHelloWorldHelloWorldAppDelegate.m",
          "functions": [
            {
              "coveredLines": 0,
```



```
xcrun xccov view --json HelloWorld.xccovreport
```

```
{
  "coveredLines": 32,
  "lineCoverage": 0.8,
  "targets": [
    {
      "coveredLines": 11,
      "lineCoverage": 0.57894736842105265,
      "files": [
        {
          "coveredLines": 3,
          "lineCoverage": 0.5,
          "path": "tmpHelloWorldHelloWorldAppDelegate.m",
          "functions": [
            {
              "coveredLines": 0,
```

man xccov

## NAME

`xccov` – view Xcode coverage data in human-readable or machine-parseable format.

## SYNOPSIS

```
xccov view [--only-targets | --files-for-target target_name | --functions-for-file
name_or_path] [--json] report.xccovreport
```

```
xccov view [--file-list | --file path] archive.xccovarchive
```

## DESCRIPTION

When tests are run with code coverage enabled, Xcode generates two files that contain its representation of coverage data. These are the coverage report, with extension `xccovreport`, and the coverage archive, with extension `xccovarchive`. The coverage report contains line coverage percentages for each target, source file, and function/method that has coverage information. The coverage archive contains the raw

# Source Editor



- Show Completions <sup>^Space</sup>
- Edit All in Scope <sup>^⌘E</sup>
- Refactor ▶
- Fix All Issues <sup>^⌘⇧F</sup>
- Show Issue
- Issues ▶
- Select Structure <sup>⇧⌘A</sup>
- Structure ▶
- Code Folding ▶
- Syntax Coloring ▶
- Font Size ▶
- Theme ▶
- Show Invisibles
- Show Last Change For Line
- Show Code Coverage
- Create Code Snippet

DevCastMac

- DevCastMac
- DevCastPhone
- DevCastKit
  - Server.swift
  - Storage.swift
  - Message.swift
  - Thread.swift
  - User.swift
  - Resources
  - DevCastKitTests
  - Products
  - Frameworks

Filter

```

Server.swift
DevCastKit

Copyright © 2018 Apple Inc. All rights reserved.

Swift Foundation
9
10 protocol Server {
11     func sendMessage(_ message: Message) throws -> MessageReceipt
12     func getMessages() throws -> [Message]
13 }
14
15 extension Server {
16     func getRecentMessages() throws -> [Message] {
17         let messages = try getMessages()
18         return Array(messages.dropFirst(max(messages.count / 2, 0)))
19     }
20 }
21
22 class LocalServer: Server {
23
24     private var storage: [Message] = []
25
26     init() { }
27
28     func sendMessage(_ message: Message) throws -> MessageReceipt {
29         storage.append(message)
30         return MessageReceipt(delivered: true, deliveryDate: Date())
31     }
32
33     func getMessages() throws -> [Message] {
34         return storage
35     }

```



DevCastMac

- DevCastMac
- DevCastPhone
- DevCastKit
  - Server.swift
  - Storage.swift
  - Message.swift
  - Thread.swift
  - User.swift
  - Resources
  - DevCastKitTests
  - Products
  - Frameworks

- Show Completions <sup>⌘</sup> Space
- Edit All in Scope <sup>⌘</sup> ⌘E
- Refactor ▶
- Fix All Issues <sup>⌘</sup> ⌘F
- Show Issue
- Issues ▶
- Select Structure <sup>⌘</sup> ⌘A
- Structure ▶
- Code Folding ▶
- Syntax Coloring ▶
- Font Size ▶
- Theme ▶
- Show Invisibles
- Show Last Change For Line
- Show Code Coverage
- Create Code Snippet

DevCastMac | Build DevCastKit: Succeeded | Today at 9:41 PM

DevCastMac > DevCastKit > Server.swift > No Selection

```

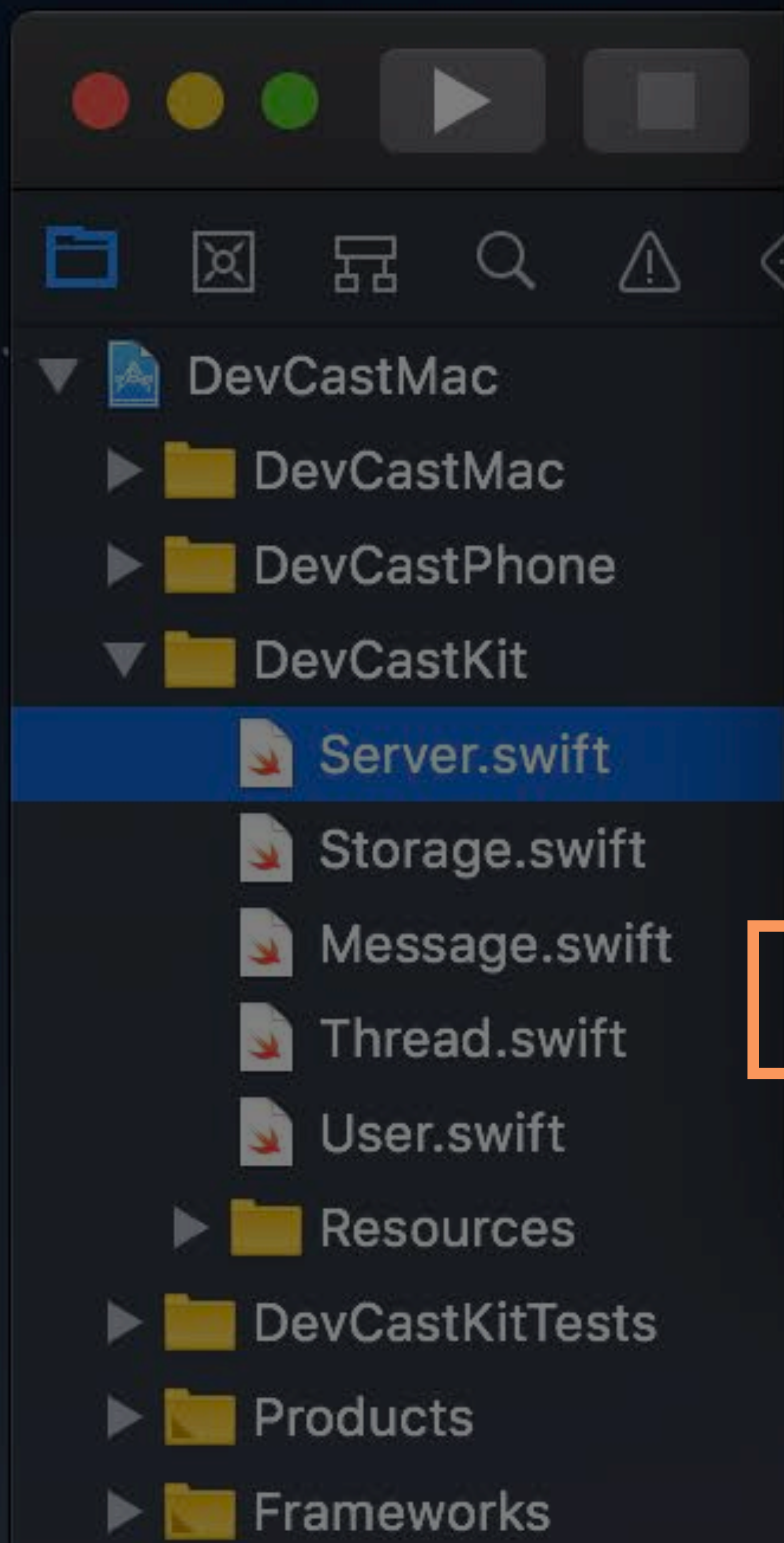
Server.swift
DevCastKit

Copyright © 2018 Apple Inc. All rights reserved.

Foundation
9
10 protocol Server {
11     func sendMessage(_ message: Message) throws -> Message
12     func getMessages() throws -> [Message]
13 }
14
15 extension Server {
16     func getRecentMessages() throws -> [Message] {
17         let messages = try getMessages()
18         return Array(messages.dropFirst(max(messages.count, 1)))
19     }

```





- Show Completions <sup>^</sup>Space
- Edit All in Scope <sup>^</sup>⌘E
- Refactor ▶
- Fix All Issues <sup>^</sup>⌘⇧F
- Show Issue
- Issues ▶
- Select Structure <sup>⌘</sup>⌘A
- Structure ▶
- Code Folding ▶
- Syntax Coloring ▶
- Font Size ▶
- Theme ▶
- Show Invisibles
- Show Last Change For Line
- Show Code Coverage
- Create Code Snippet

DevCastMac | Build DevCastKit: Succeeded | Today at 9:41 PM

DevCastMac > DevCastKit > Server.swift > No Selection

```
Server.swift
DevCastKit
Copyright © 2018 Apple Inc. All rights reserved.
Foundation
9
10 protocol Server {
11     func sendMessage(_ message: Message) throws -> Message
12     func getMessages() throws -> [Message]
13 }
14
15 extension Server {
16     func getRecentMessages() throws -> [Message] {
17         let messages = try getMessages()
18         return Array(messages.dropFirst(max(messages.count, 1)))
19     }

```



```
DevCastMac | Build DevCastKit: Succeeded | Today at 9:41 PM  
DevCastMac > DevCastKit > Server.swift > No Selection  
DevCastMac  
├── DevCastMac  
├── DevCastPhone  
└── DevCastKit  
    ├── Server.swift  
    ├── Storage.swift  
    ├── Message.swift  
    ├── Thread.swift  
    ├── User.swift  
    └── Resources  
DevCastKitTests  
Products  
Frameworks  
1 //  
2 // Server.swift  
3 // DevCastKit  
4 //  
5 // Copyright © 2018 Apple Inc. All rights reserved.  
6 //  
7  
8 import Foundation  
9  
10 protocol Server {  
11     func sendMessage(_ message: Message) throws -> MessageReceipt  
12     func getMessages() throws -> [Message]  
13 }  
14  
15 extension Server {  
16     func getRecentMessages() throws -> [Message] {  
17         let messages = try getMessages()  
18         return Array(messages.dropFirst(max(messages.count / 2, 0)))  
19     }  
20 }  
21  
22 class LocalServer: Server {  
23  
24     private var storage: [Message] = []  
25  
26     init() { }  
27  
28     func sendMessage(_ message: Message) throws -> MessageReceipt {  
29         storage.append(message)  
30         return MessageReceipt(delivered: true, deliveryDate: Date())  
31     }  
32  
33     func getMessages() throws -> [Message] {  
34         return storage  
35     }  
0  
1  
1  
4
```



***Demo***



Code coverage

Test selection and ordering

Parallel testing

Code coverage

**Test selection and ordering**

Parallel testing

# Test Selection

Not all tests are equal

- 1,000 unit tests
- 10 UI tests

Different tests in different situations

- Pre-commit
- Nightly

# Test Selection with Schemes



# Test Selection with Schemes

“Tests to skip”

- Automatically includes new tests

# Test Selection with Schemes

NEW

“Tests to skip”

- Automatically includes new tests

“Tests to run”

- Explicit opt-in for tests



DevCastKit > My Mac | Build DevCastKit: Succeeded | Today at 9:41 PM

DevCastMac

- DevCastMac
- DevCastPhone
- DevCastKit
  - Server.swift
  - Storage.swift
  - Message.swift
  - Thread.swift
  - User.swift
  - Resources
  - DevCastKitTests
  - Products
  - Frameworks

1 //  
2 //  
3 //  
4 //  
5 //  
6 //  
7 //  
8 imp  
9  
10 pro  
11  
12  
13 }  
14  
15 ext  
16  
17  
18  
19  
20 }  
21  
22 class LocalServer: Server {  
23  
24 private var storage: [Message] = []  
25  
26 init() { }  
27  
28 func sendMessage(\_ message: Message) throws -> MessageReceipt {  
29 storage.append(message)  
30 return MessageReceipt(delivered: true, deliveryDate: Date())  
31 }  
32  
33 func getMessages() throws -> [Message] {  
34 return storage  
35 }

Build 1 target  
Run Debug Build Configuration: Debug  
Test Debug Debugger: Debug executable  
Profile Release Debug Process As: Me (honza)  
Analyze Debug root  
Archive Release

Tests Enabled  
DevCastKitTests Options... [checked]

Execute in parallel [unchecked]  
Randomize execution order [unchecked]  
Automatically include new tests [checked]  
Location: None

Duplicate Scheme Manage Schemes... [checked] Shared Close



Build 1 target  
Run Debug  
Test Debug  
Profile Release  
Analyze Debug  
Archive Release

Info Arguments Options Diagnostics

Build Configuration: Debug

Debugger:  Debug executable

Debug Process As:  Me (honza)  root

Tests: Enabled

DevCastKitTests Options...

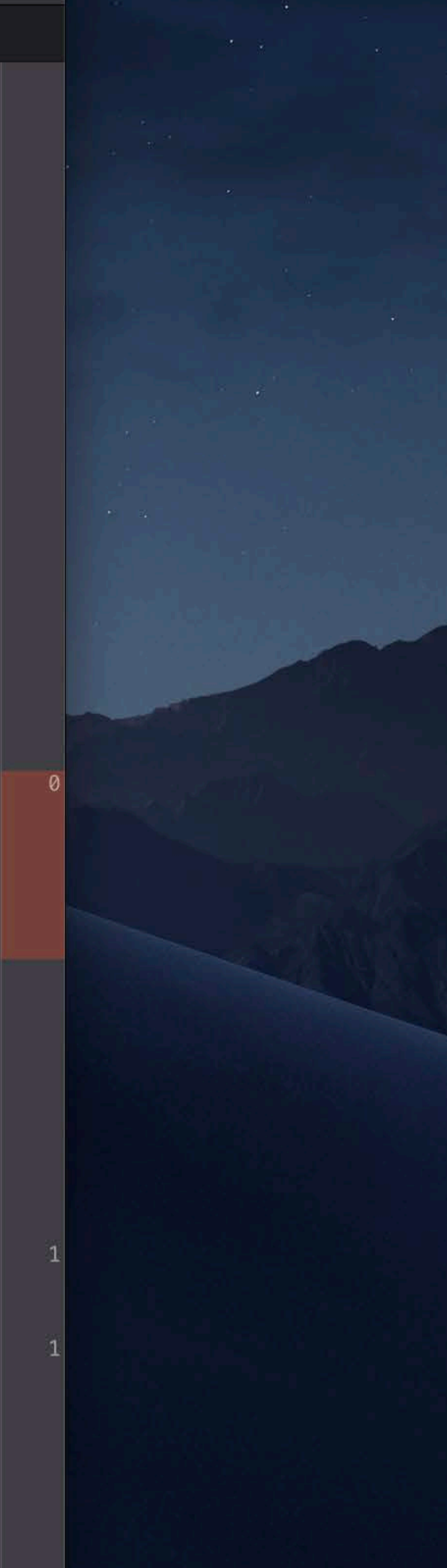
+ -

Duplicate Scheme Manage Schemes...  Shared Close

Execute in parallel  
 Randomize execution order  
 Automatically include new tests

Location: None

```
localServer: Server {  
private var storage: [Message] = []  
  
t() { }  
  
sendMessage(_ message: Message) throws -> MessageReceipt {  
storage.append(message)  
return MessageReceipt(delivered: true, deliveryDate: Date())  
}
```





DevCastKit > My Mac

Build  
1 target

Run  
Debug

Test  
Debug

Profile  
Release

Analyze  
Debug

Archive  
Release

Info Arguments Options Diagnostics

Build Configuration: Debug

Debugger:  Debug executable

Debug Process As:  Me (honza)  root

Tests: Enabled

DevCastKitTests Options...

Execute in parallel

Randomize execution order

**Automatically include new tests**

Location: None

Duplicate Scheme Manage Schemes...  Shared Close

```
localServer: Server {  
private var storage: [Message] = []  
  
t() { }  
  
sendMessage(_ message: Message) throws -> MessageReceipt {  
storage.append(message)  
return MessageReceipt(delivered: true, deliveryDate: Date())  
}
```



# Test Ordering



# Test Ordering

## Alphabetical

- Default
- Deterministic
- Can hide implicit dependencies



# Implicit Dependencies Between Tests



# Implicit Dependencies Between Tests

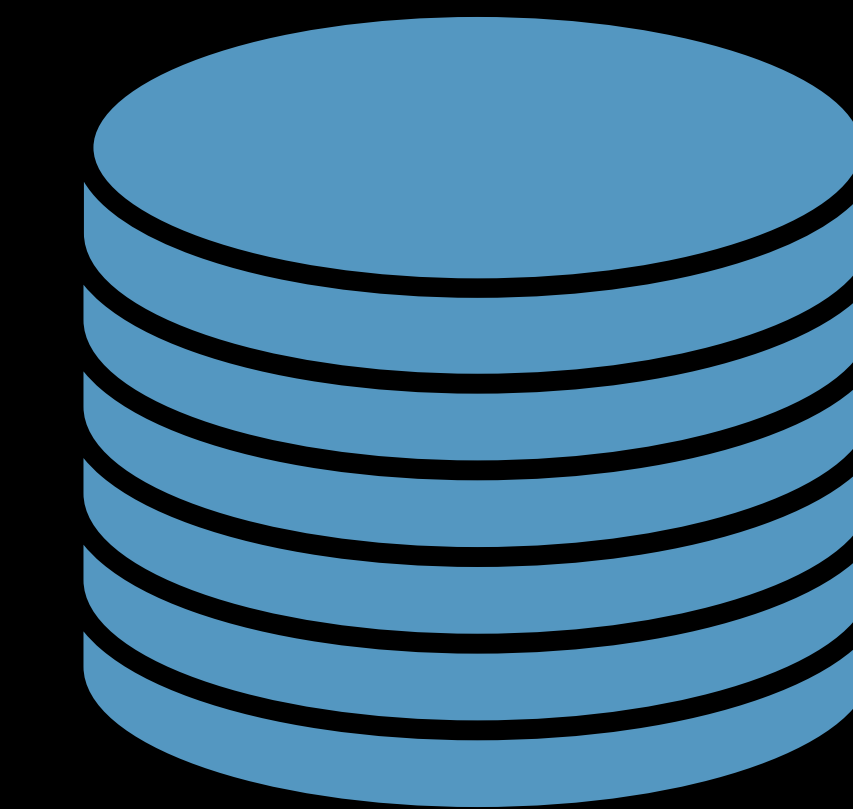
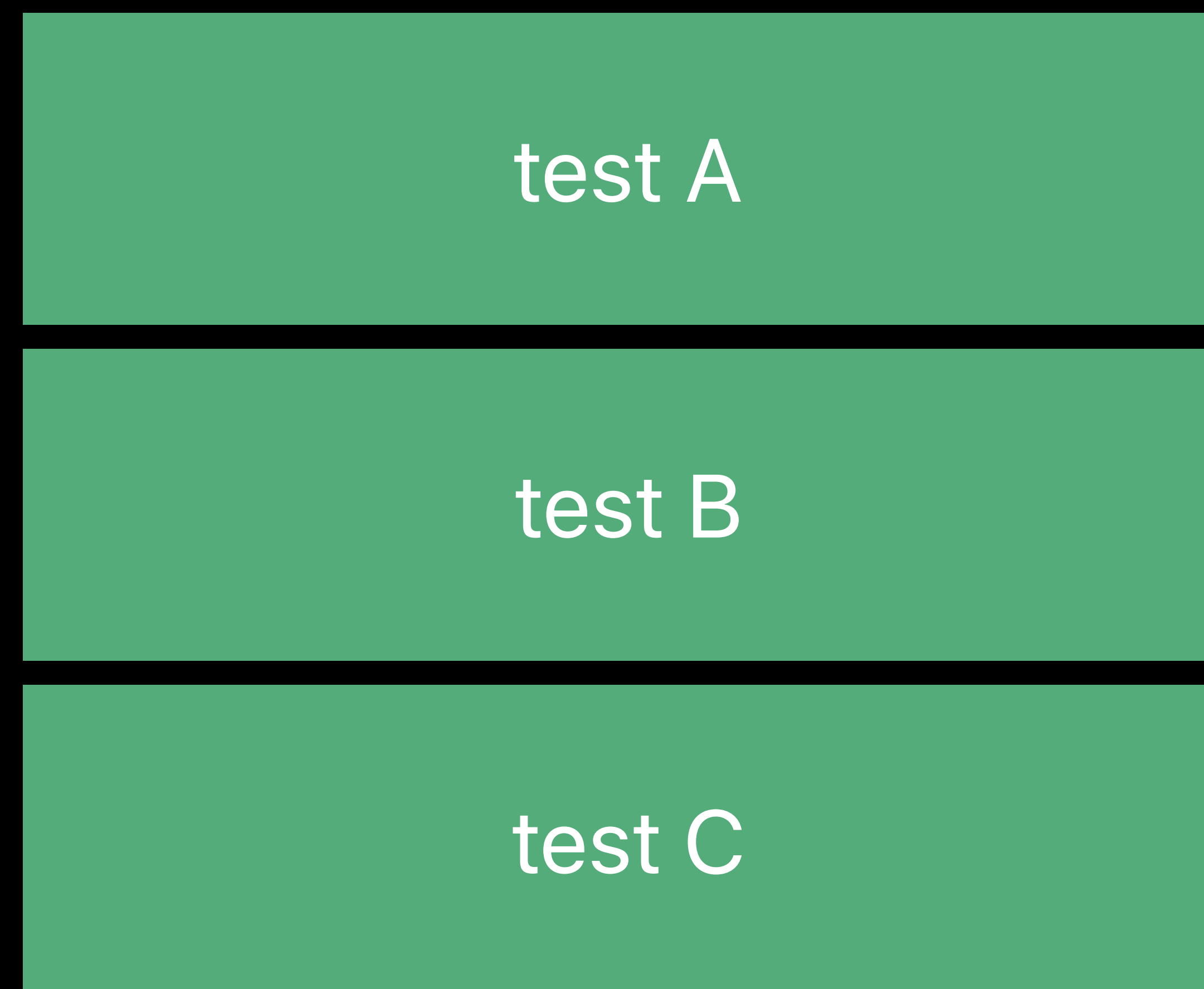


test A

test B

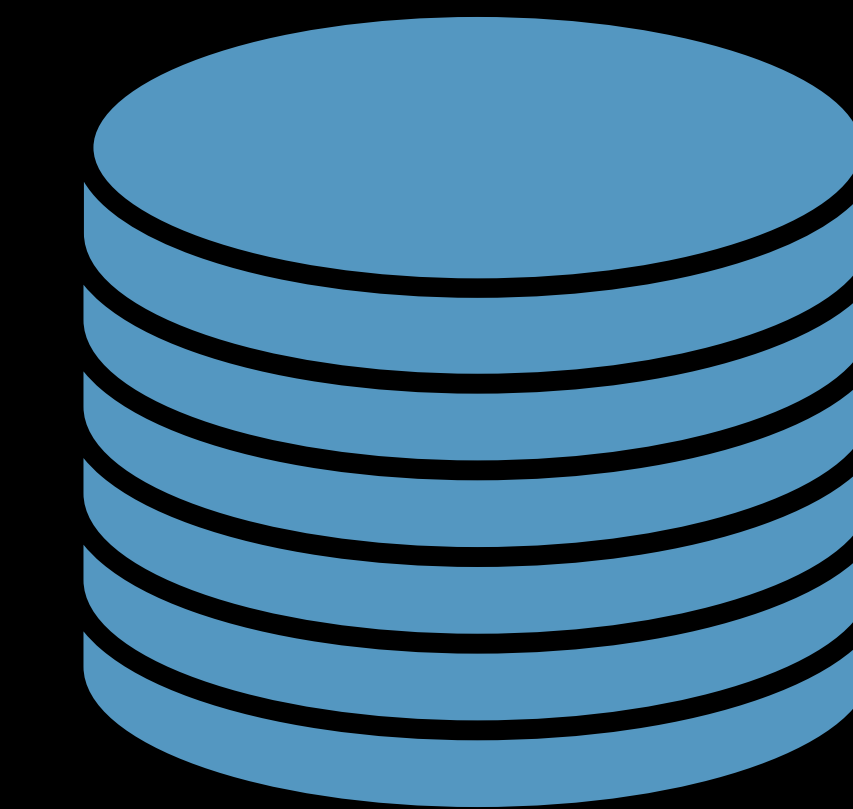
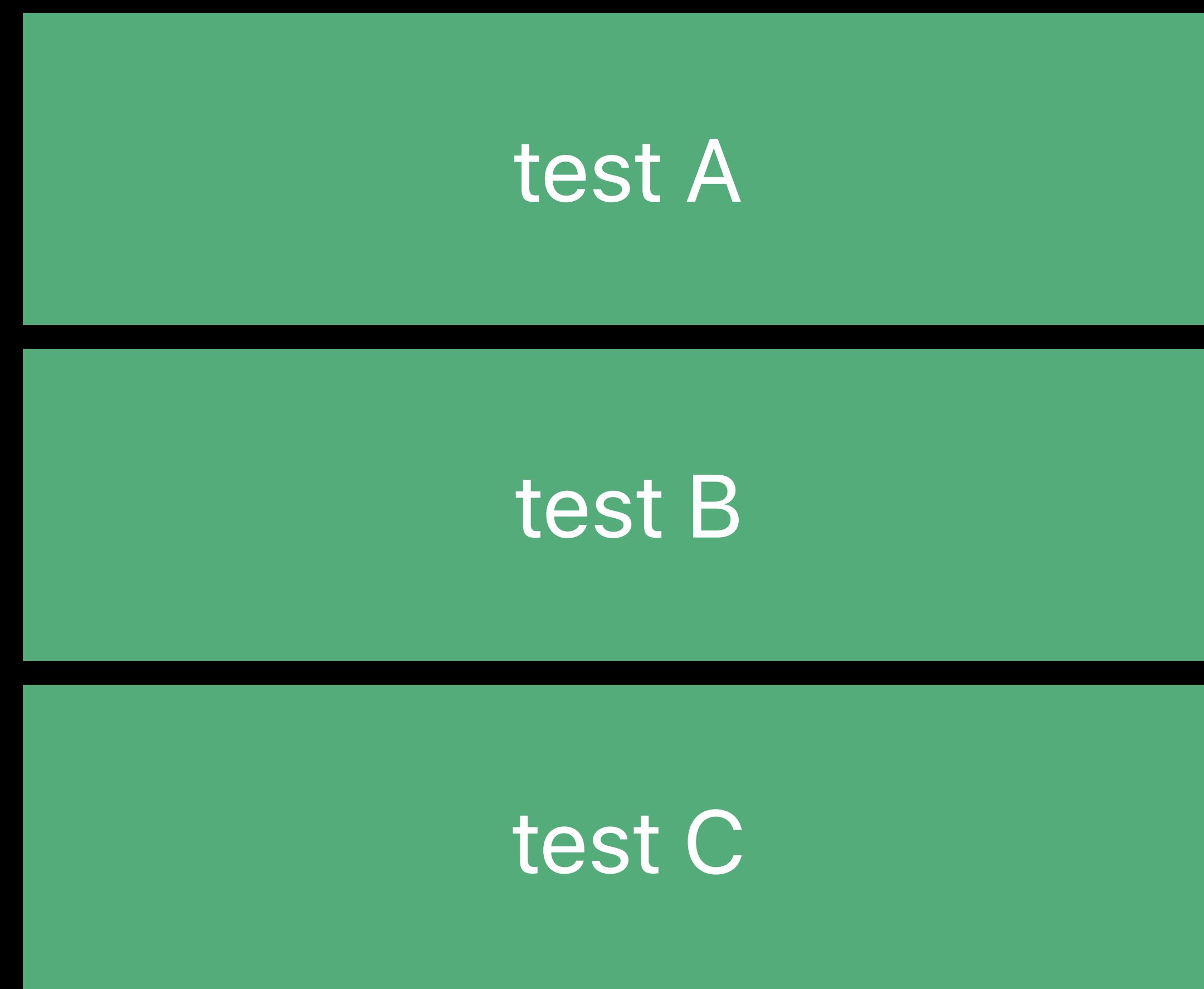
test C

# Implicit Dependencies Between Tests





# Implicit Dependencies Between Tests



# Implicit Dependencies Between Tests



test A

test B

test C



# Implicit Dependencies Between Tests



test A

test B

test C

# Implicit Dependencies Between Tests

test B

test A

test C



# Implicit Dependencies Between Tests

test B

test A

test C

# Test Ordering

## Alphabetical

- Default
- Deterministic
- Can hide implicit dependencies





# Test Ordering

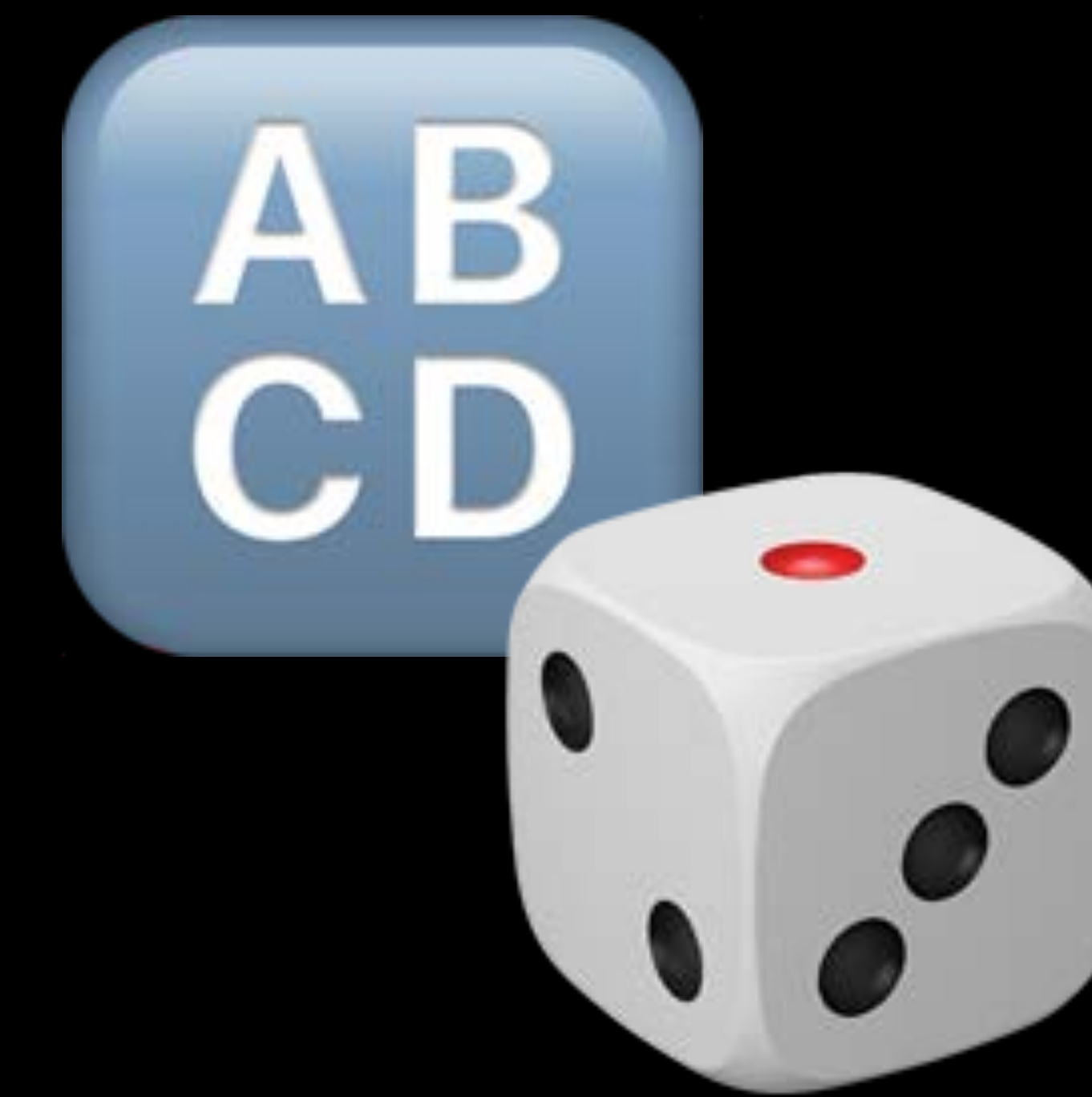
NEW

## Alphabetical

- Default
- Deterministic
- Can hide implicit dependencies

## Randomized

- Nondeterministic
- Can uncover implicit dependencies





DevCastKit > My Mac | Build DevCastKit: Succeeded | Today at 9:41 PM

DevCastKit > My Mac

- DevCastMac
  - DevCastMac
  - DevCastPhone
  - DevCastKit
    - Server.swift
    - Storage.swift
    - Message.swift
    - Thread.swift
    - User.swift
    - Resources
    - DevCastKitTests
    - Products
    - Frameworks

```

1 //
2 //
3 //
4 //
5 //
6 //
7 //
8 imp
9
10 pro
11
12
13 }
14
15 ext
16
17
18
19
20 }
21
22 class LocalServer: Server {
23
24     private var storage: [Message] = []
25
26     init() { }
27
28     func sendMessage(_ message: Message) throws -> MessageReceipt {
29         storage.append(message)
30         return MessageReceipt(delivered: true, deliveryDate: Date())
31     }
32
33     func getMessages() throws -> [Message] {
34         return storage
35     }

```

DevCastKit > My Mac

Build 1 target

Run Debug

Test Debug

Profile Release

Analyze Debug

Archive Release

Build Configuration: Debug

Debugger:  Debug executable

Debug Process As: Me (honza)

root

Tests: Enabled

DevCastKitTests

Options...

Execute in parallel

Randomize execution order

Automatically include new tests

Location: None

Duplicate Scheme | Manage Schemes... |  Shared | Close



DevCastKit > My Mac

Build 1 target

Run Debug

Test Debug

Profile Release

Analyze Debug

Archive Release

Info Arguments Options Diagnostics

Build Configuration: Debug

Debugger:  Debug executable

Debug Process As:  Me (honza)  root

Tests

DevCastKitTests Options...

Execute in parallel

Randomize execution order

Automatically include new tests

Location: None

Duplicate Scheme Manage Schemes...  Shared Close

```
class LocalServer: Server {  
  
    private var storage: [Message] = []  
  
    init() { }  
  
    func sendMessage(_ message: Message) throws -> MessageReceipt {  
        storage.append(message)  
        return MessageReceipt(delivered: true, deliveryDate: Date())  
    }  
  
    func getMessages() throws -> [Message] {
```



DevCastKit > My Mac

Build 1 target

Run Debug

Test Debug

Profile Release

Analyze Debug

Archive Release

Info Arguments Options Diagnostics

Build Configuration: Debug

Debugger:  Debug executable

Debug Process As:  Me (honza)  root

Tests

DevCastKitTests

Options...

Execute in parallel

Randomize execution order

Automatically include new tests

Location: None

Duplicate Scheme Manage Schemes...  Shared Close

```
class LocalServer: Server {  
  
    private var storage: [Message] = []  
  
    init() { }  
  
    func sendMessage(_ message: Message) throws -> MessageReceipt {  
        storage.append(message)  
        return MessageReceipt(delivered: true, deliveryDate: Date())  
    }  
  
    func getMessages() throws -> [Message] {
```



Code coverage

Test selection and ordering

Parallel testing

Code coverage

Test selection and ordering

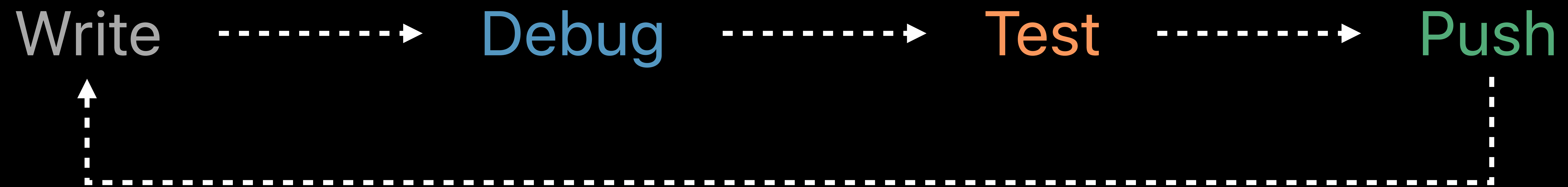
Parallel testing



# Parallel Testing

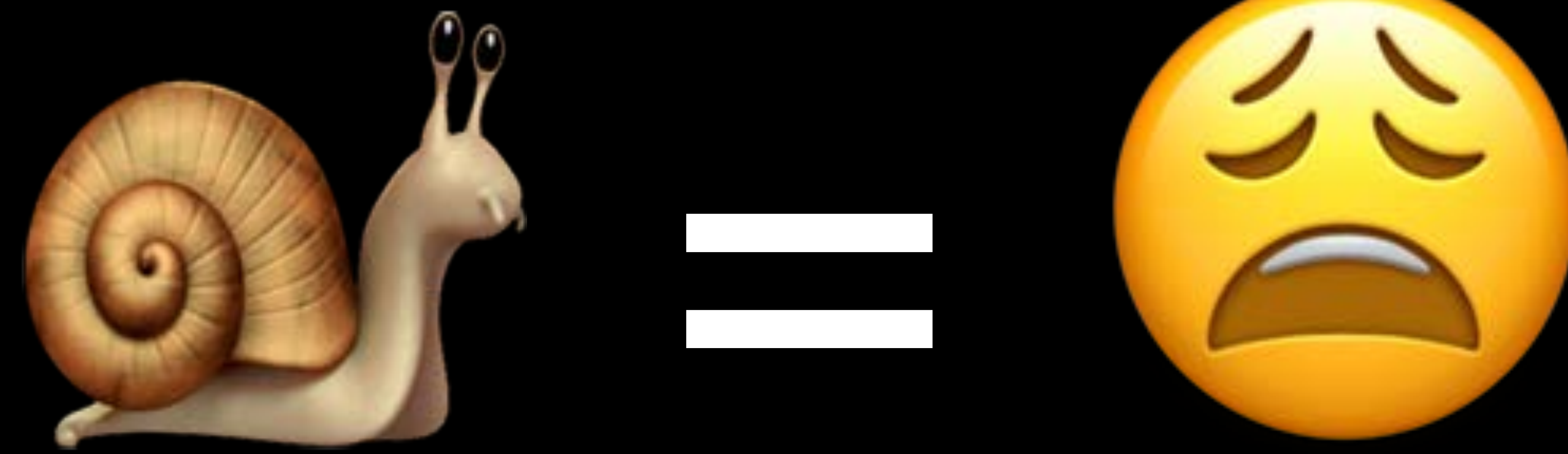
Ethan Vaughan, Xcode Engineer

# Development Cycle

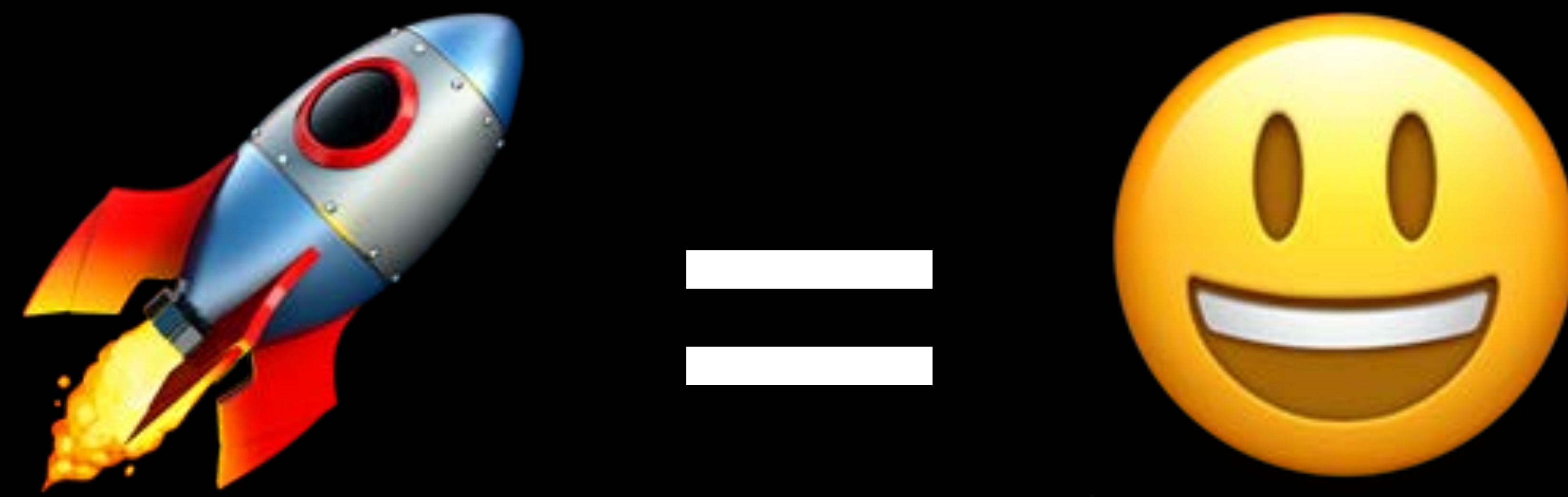




# Waiting for Tests to Finish



# Waiting for Tests to Finish





# Parallel Destination Testing

9.0

Ability to run all of your tests on multiple destinations simultaneously

```
xcodebuild test
  -destination 'platform=iOS,name=iPhone X'
  -destination 'platform=iOS,name=iPad'
```

# Parallel Destination Testing

8.0



iPhone X



iPad



Time



# Parallel Destination Testing

9.0



iPhone X



iPad



# Parallel Destination Testing

Limitations



# Parallel Destination Testing

## Limitations

Only beneficial if testing on multiple destinations

# Parallel Destination Testing

## Limitations

Only beneficial if testing on multiple destinations

Only available from xcodebuild



NEW

# Parallel Distributed Testing

# Parallel Distributed Testing

NEW

Execute tests in parallel on a single destination

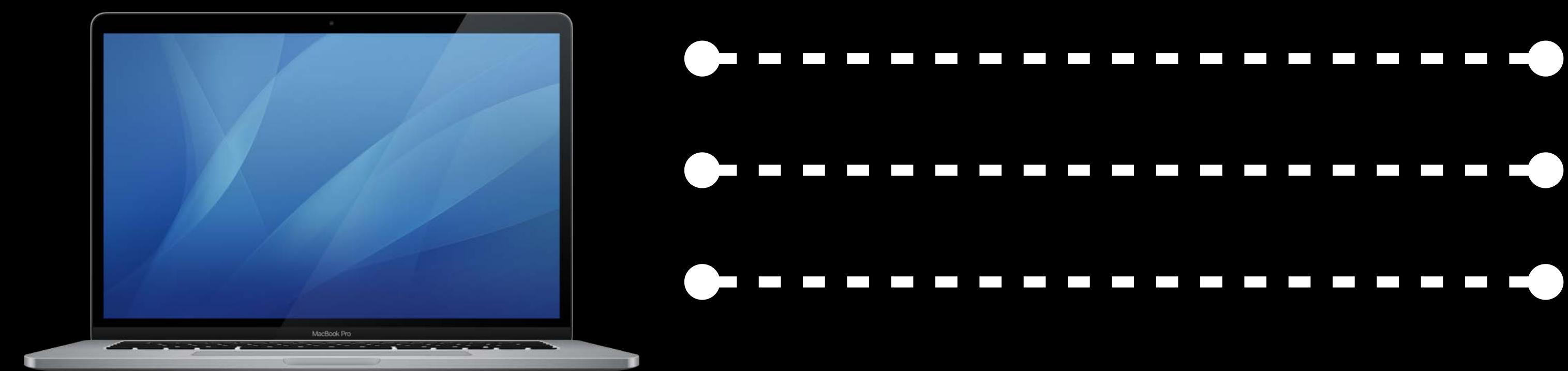




# Parallel Distributed Testing

NEW

Execute tests in parallel on a single destination



Time

# Testing Architecture



# Testing Architecture

Unit tests

```
func testFoo() { ... }
```

# Testing Architecture

Unit tests

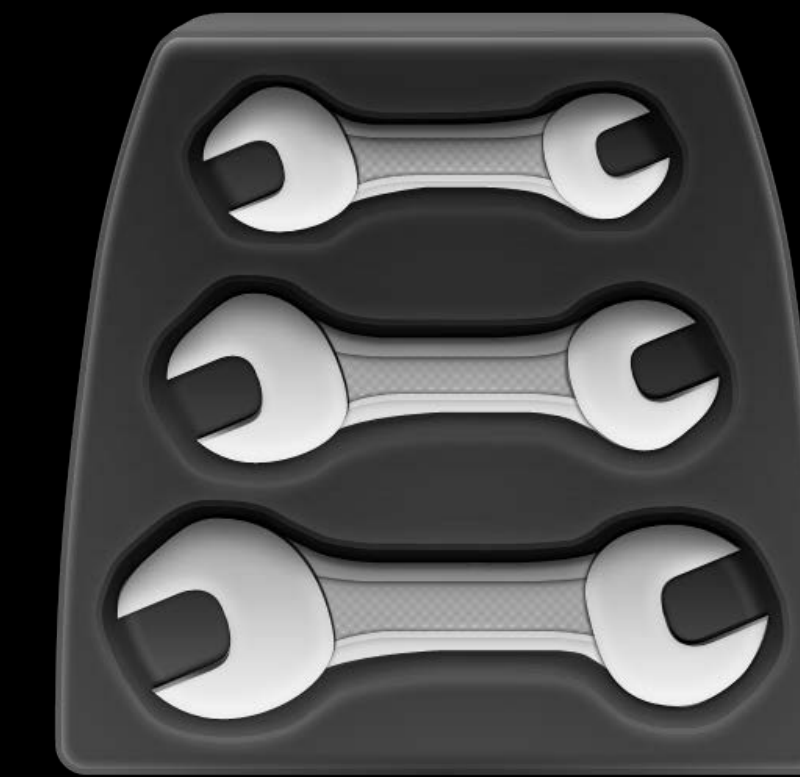
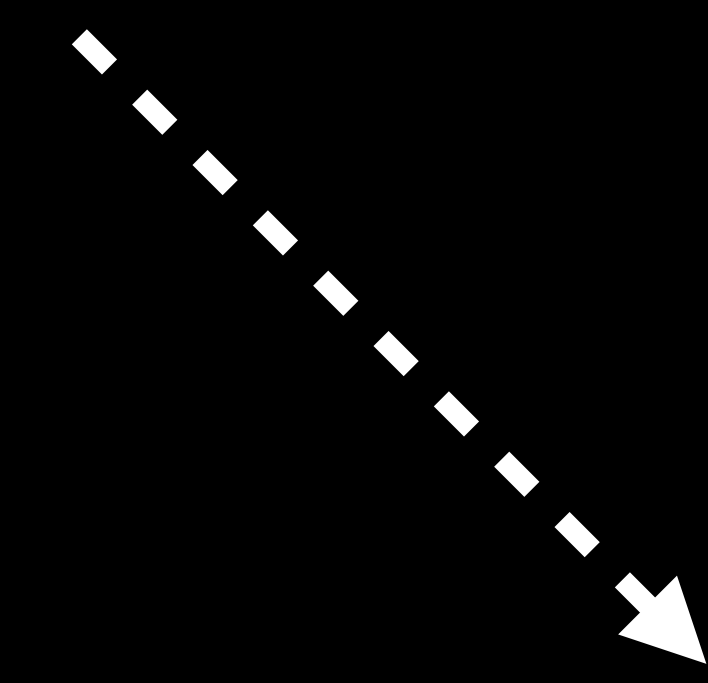
```
func testFoo() { ... }
```



# Testing Architecture

## Unit tests

```
func testFoo() { ... }
```

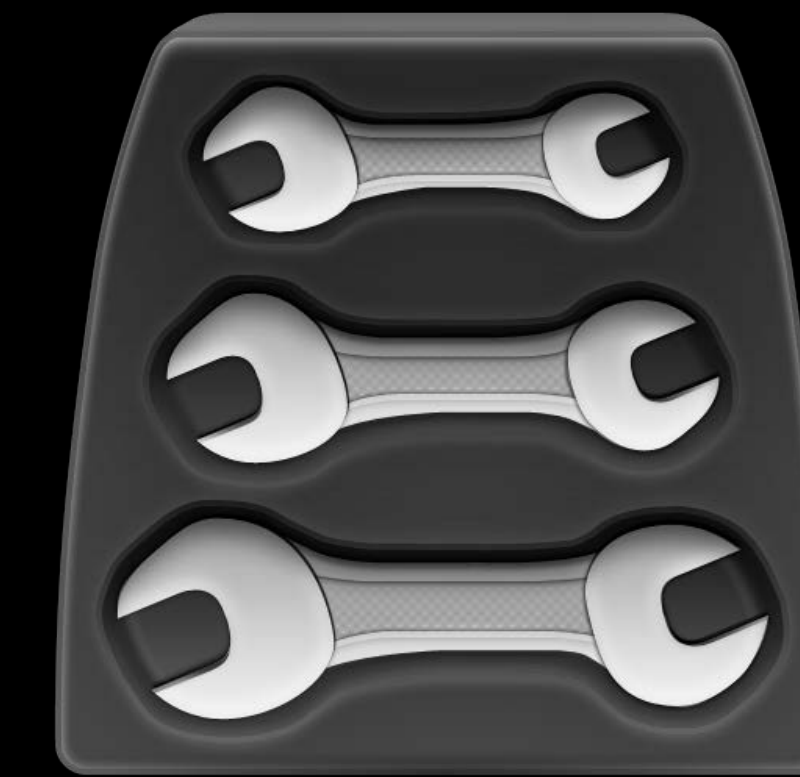
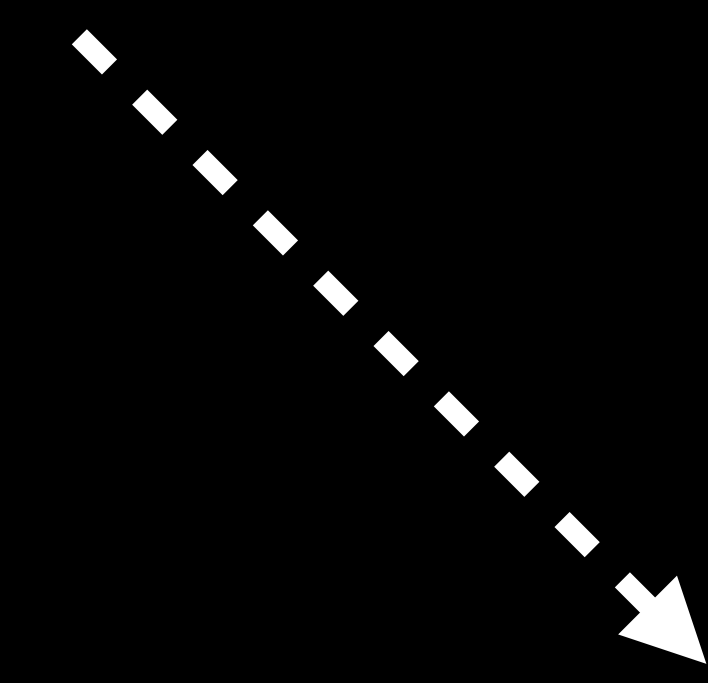


Test Bundle

# Testing Architecture

## Unit tests

```
func testFoo() { ... }
```



Test Bundle

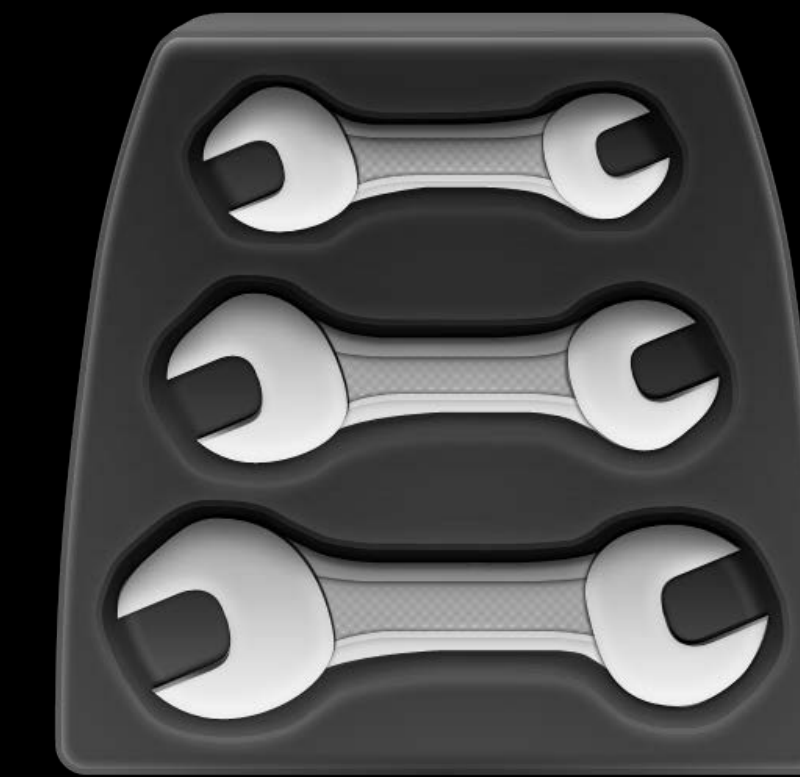


# Testing Architecture

Unit tests



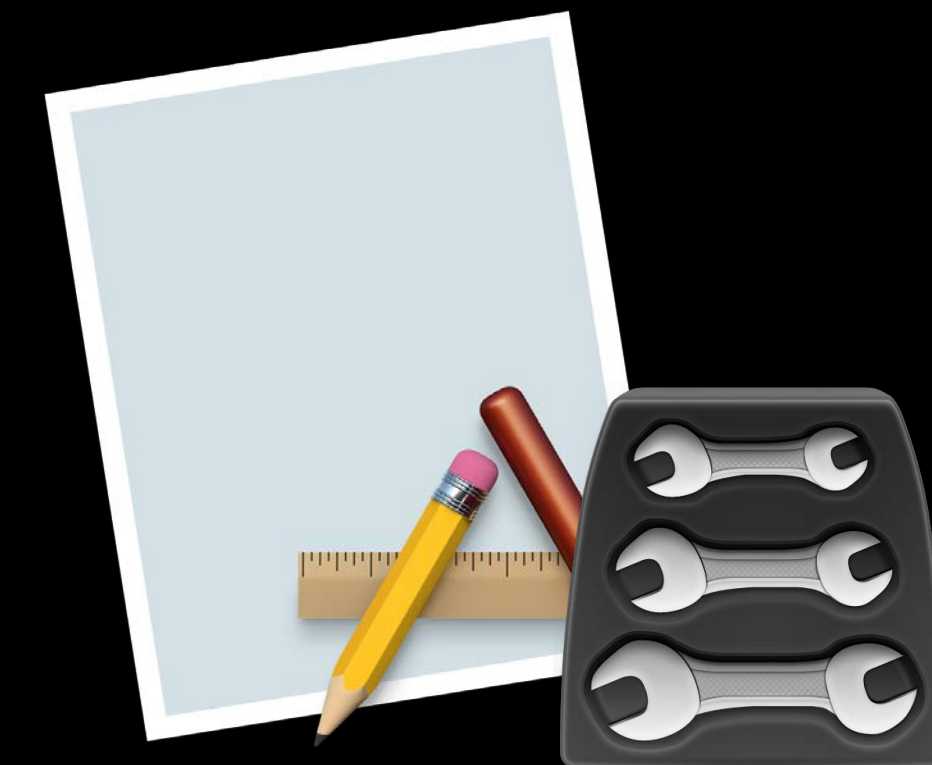
Your App



Test Bundle

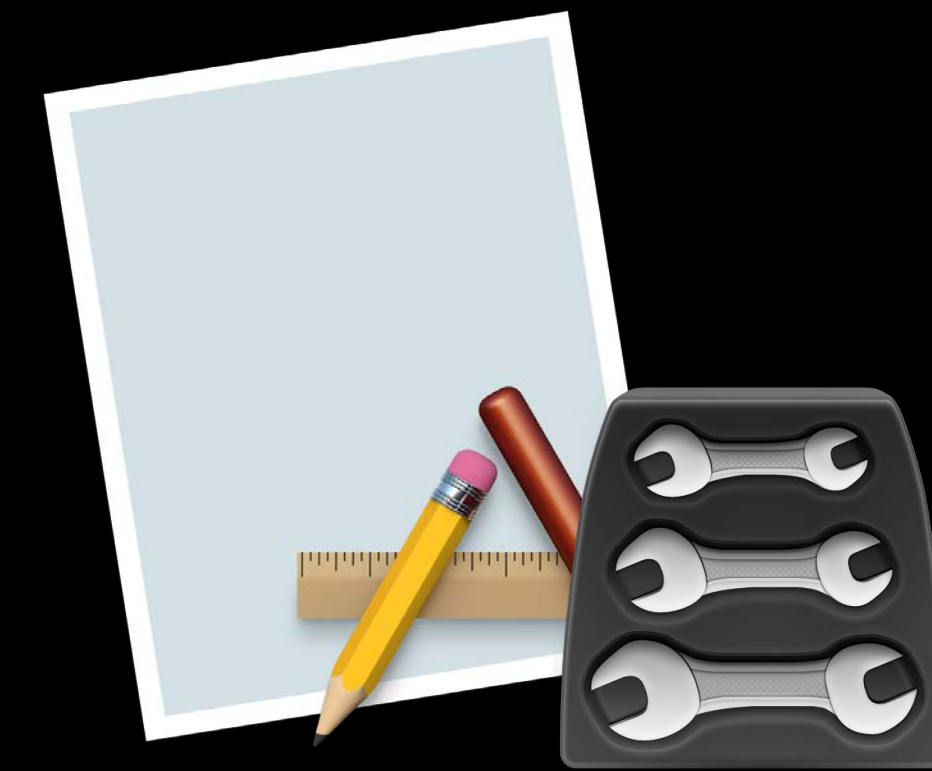
# Testing Architecture

Unit tests



# Testing Architecture

UI tests

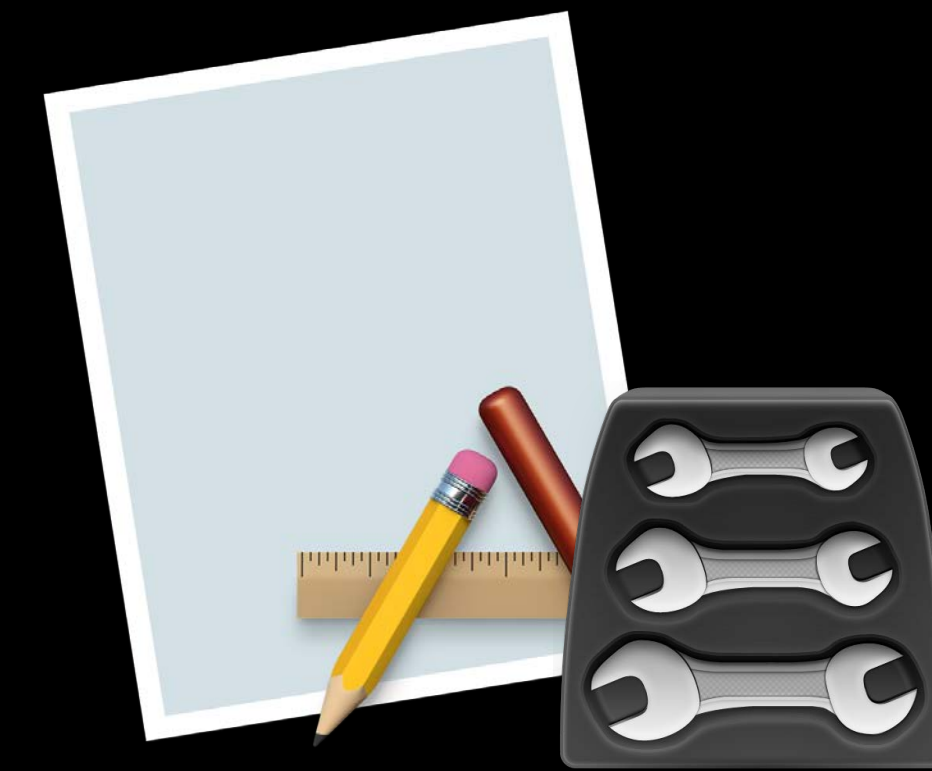


UI Test Runner



# Testing Architecture

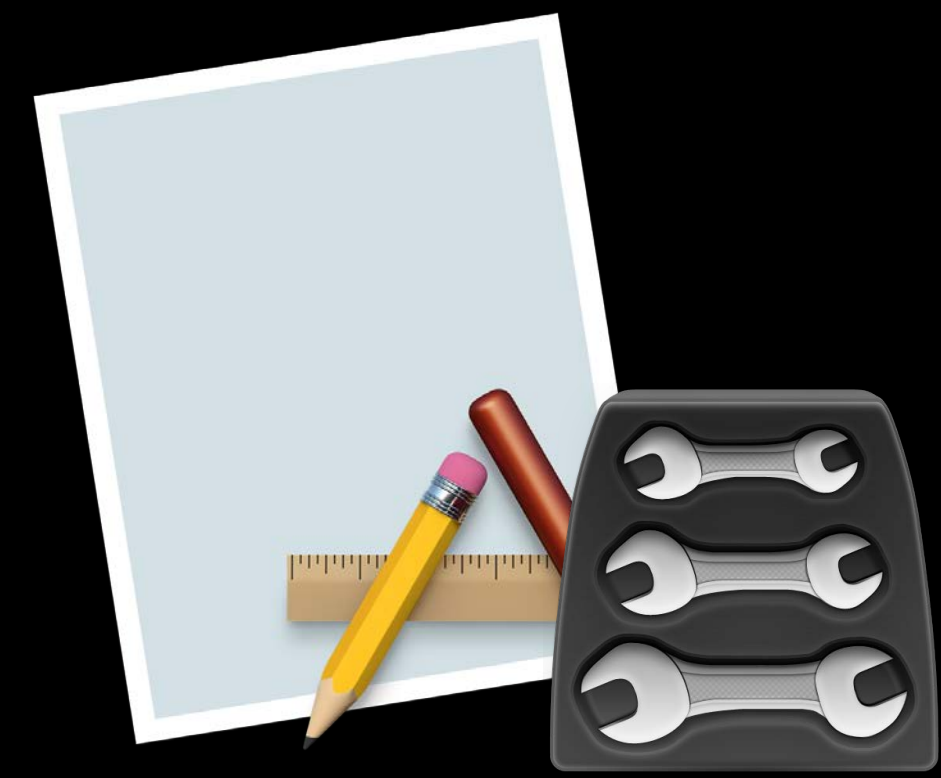
UI tests



UI Test Runner

# Testing Architecture

UI tests



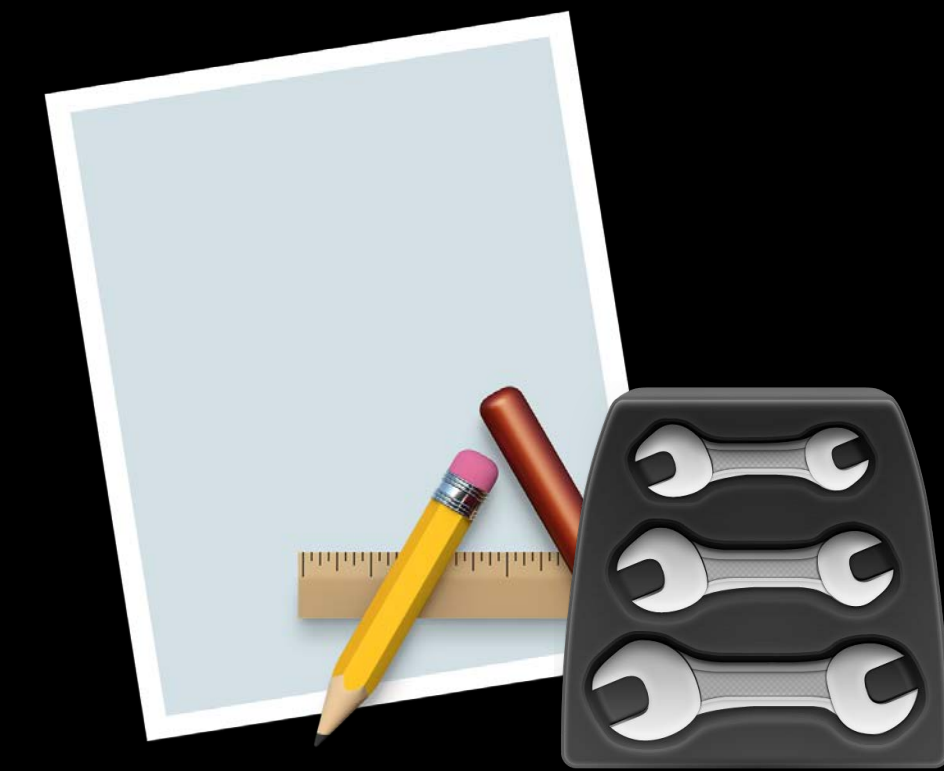
UI Test Runner



Your App

# Testing Architecture

UI tests



UI Test Runner

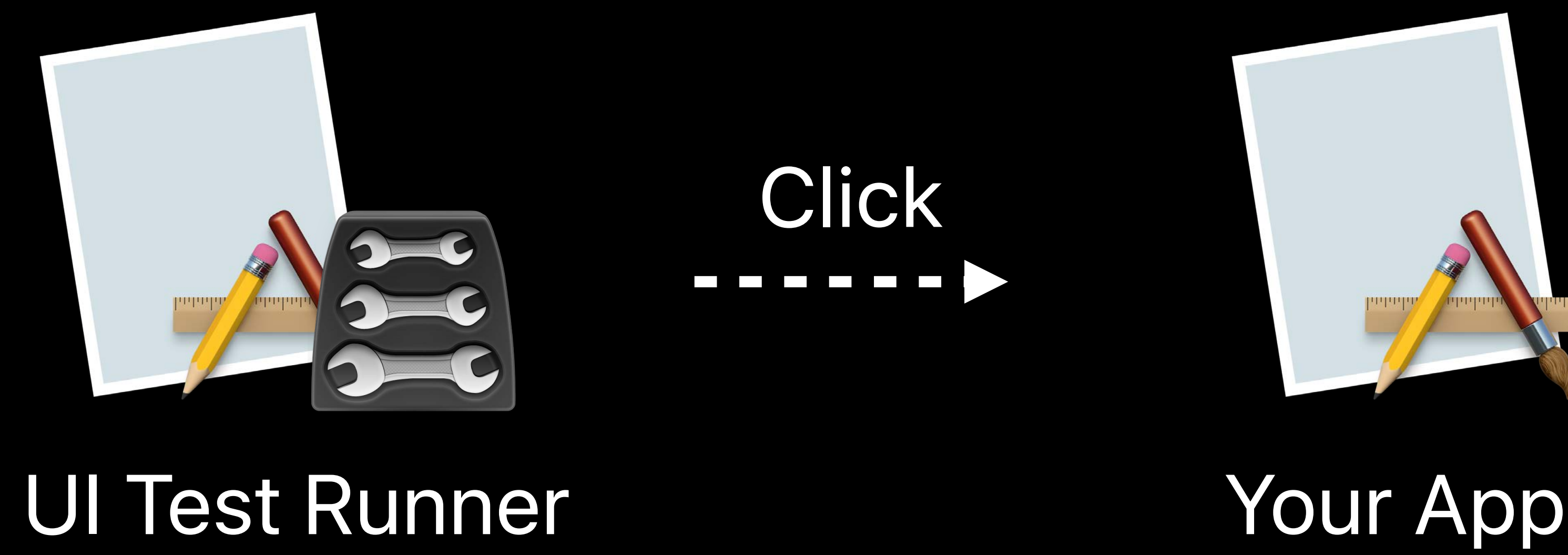


Your App



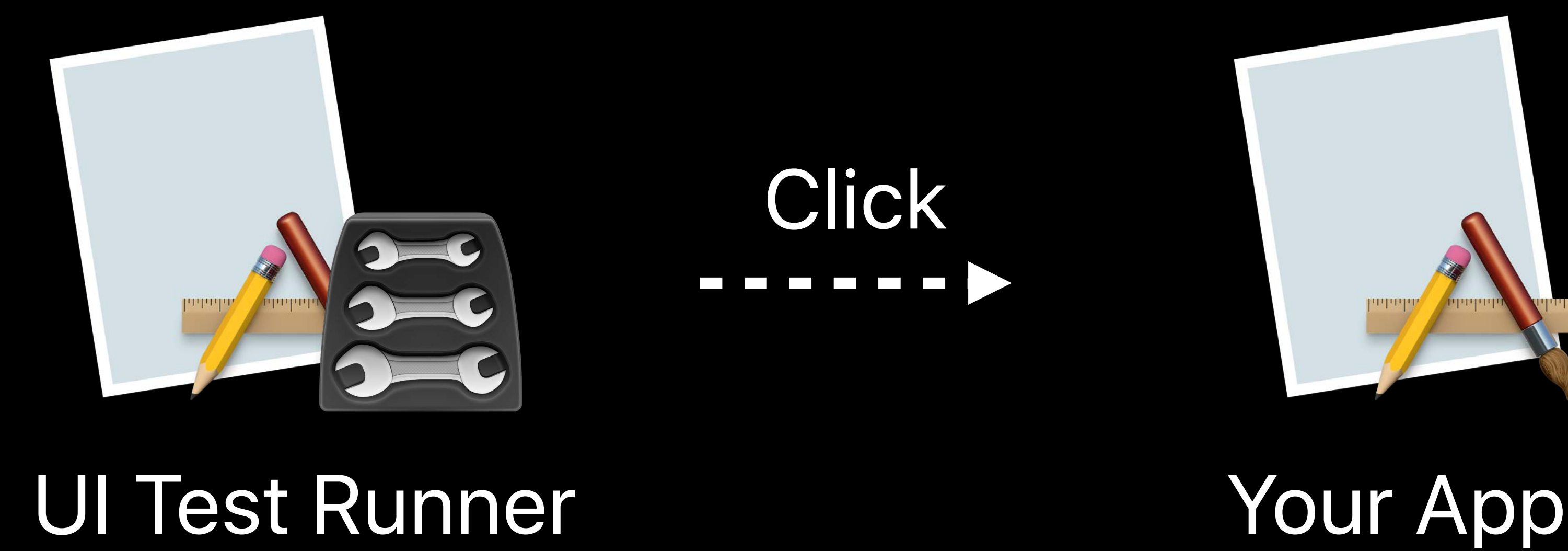
# Testing Architecture

UI tests



# Testing Architecture

## UI tests



# Overview

## Parallel Distributed Testing



Xcode



# Overview

## Parallel Distributed Testing



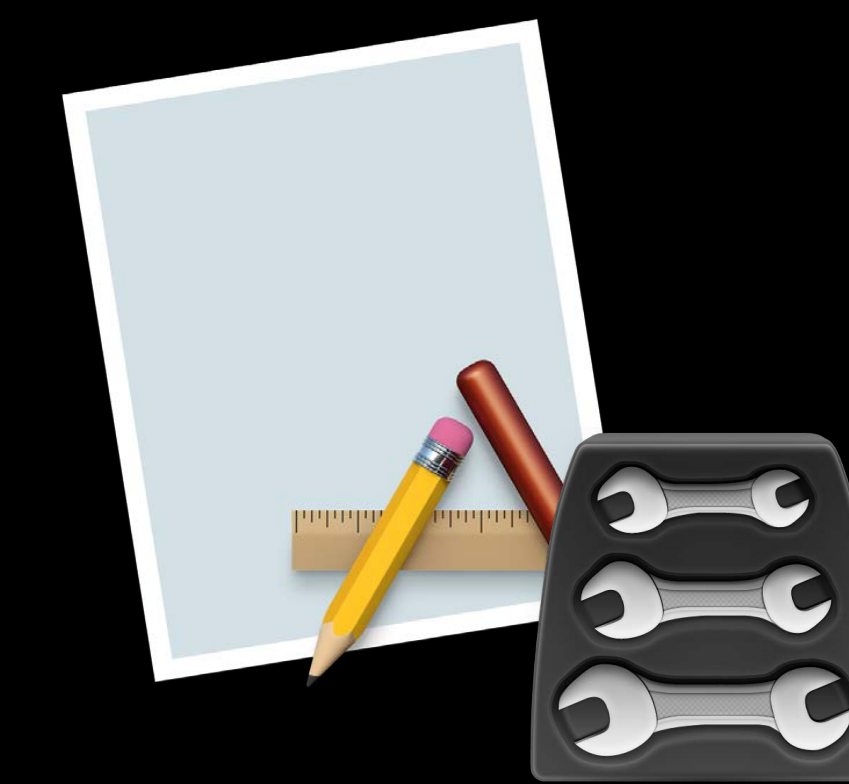
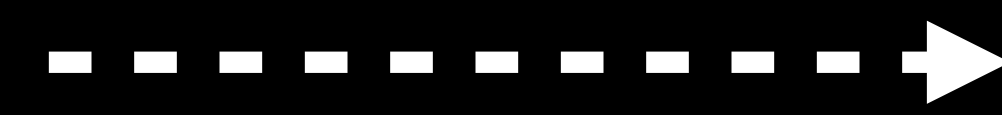
Xcode

# Overview

## Parallel Distributed Testing



Xcode



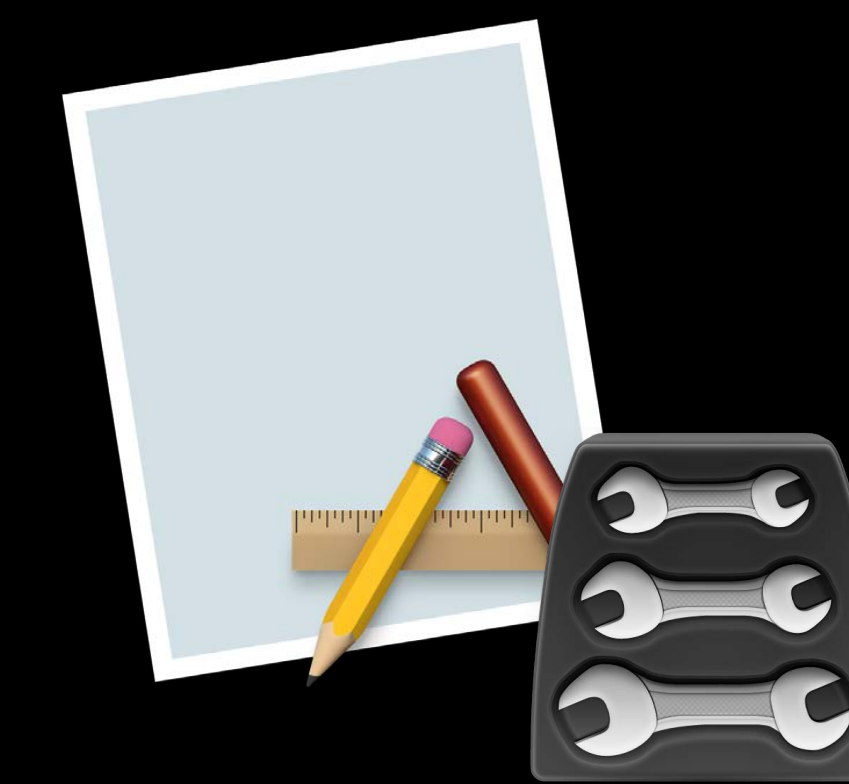
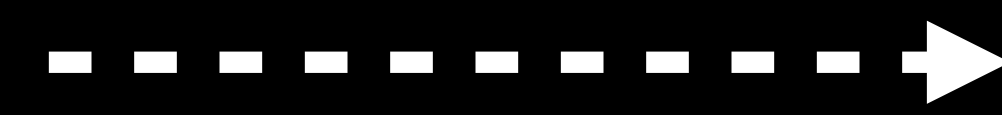
Runner

# Overview

## Parallel Distributed Testing



Xcode



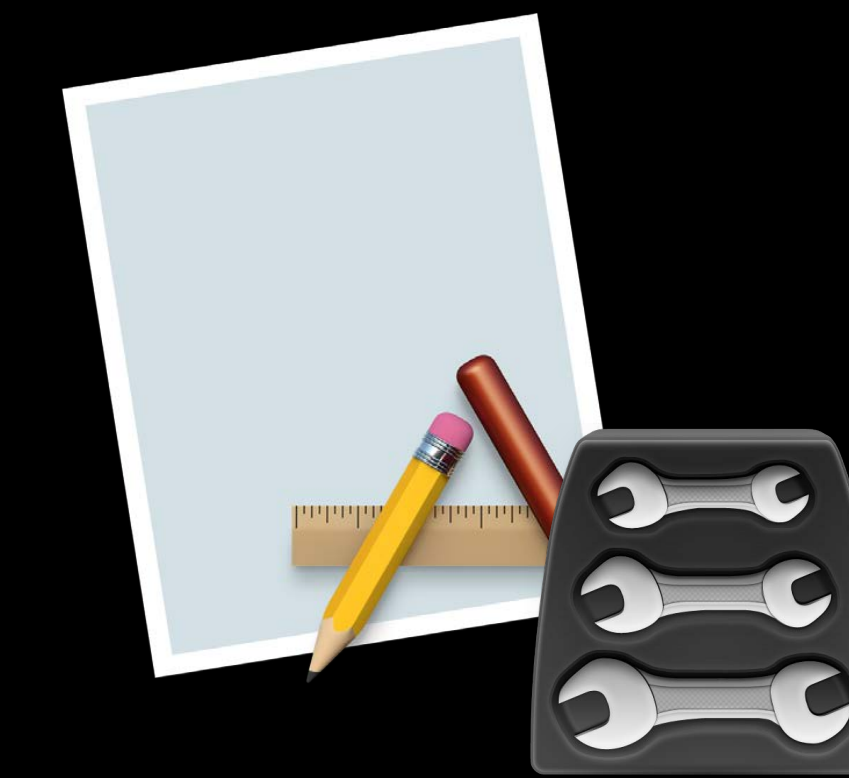
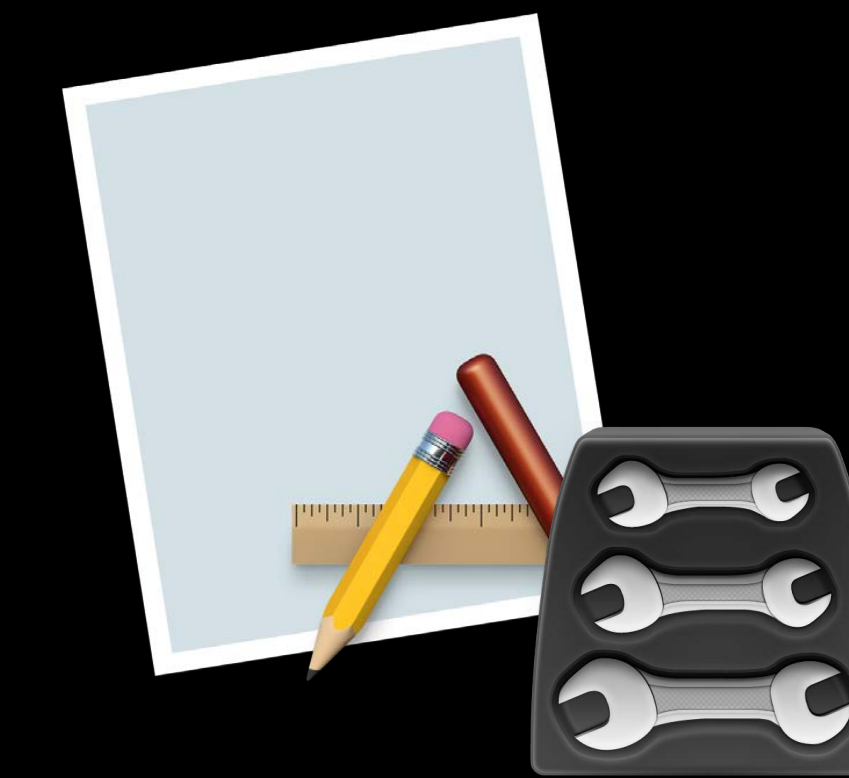
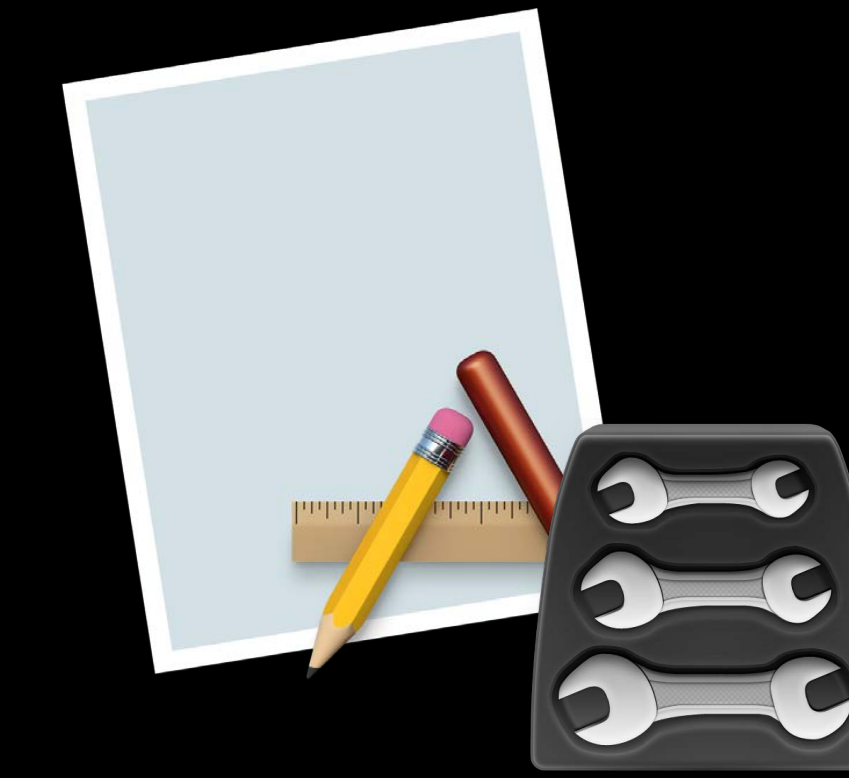
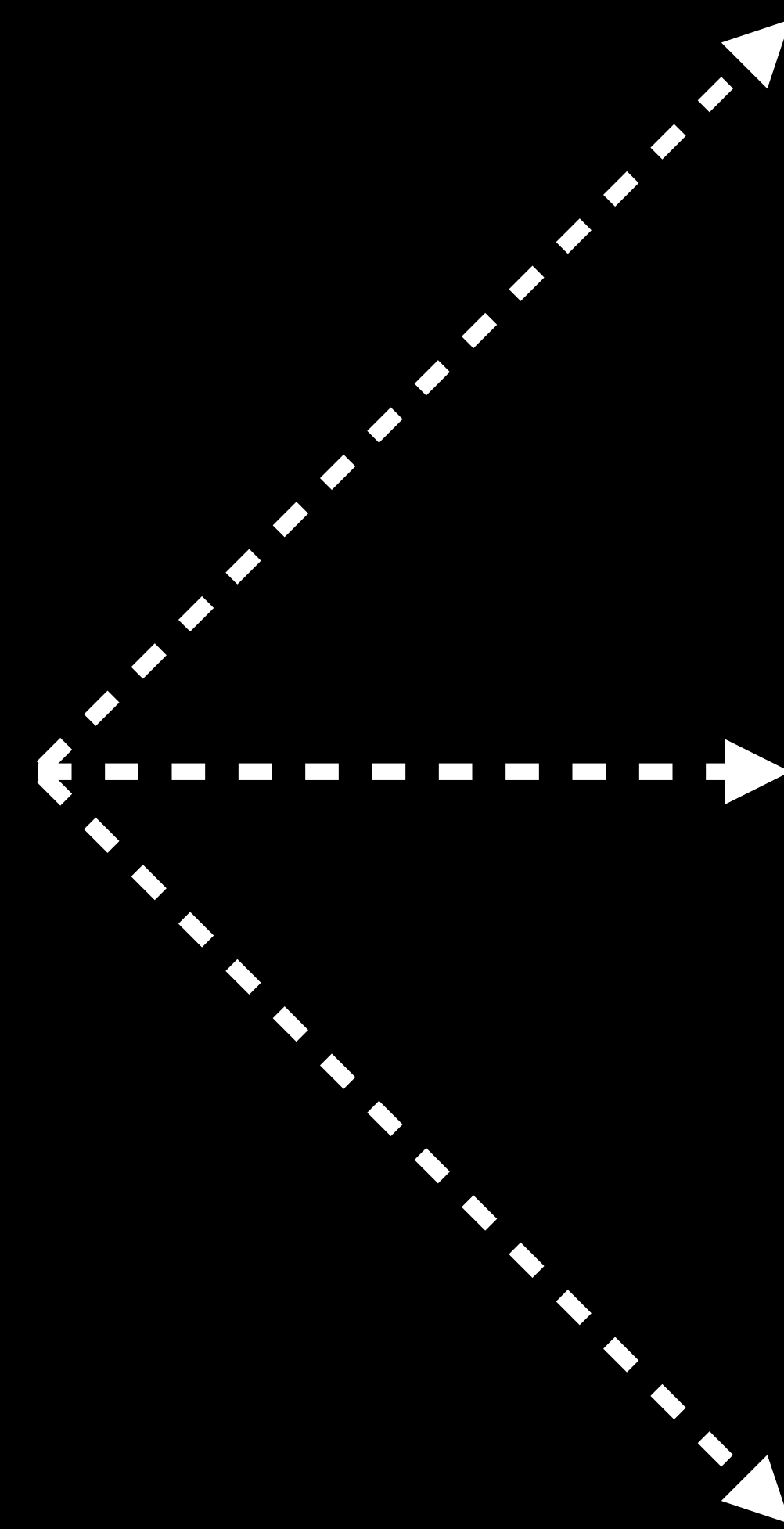


# Overview

## Parallel Distributed Testing

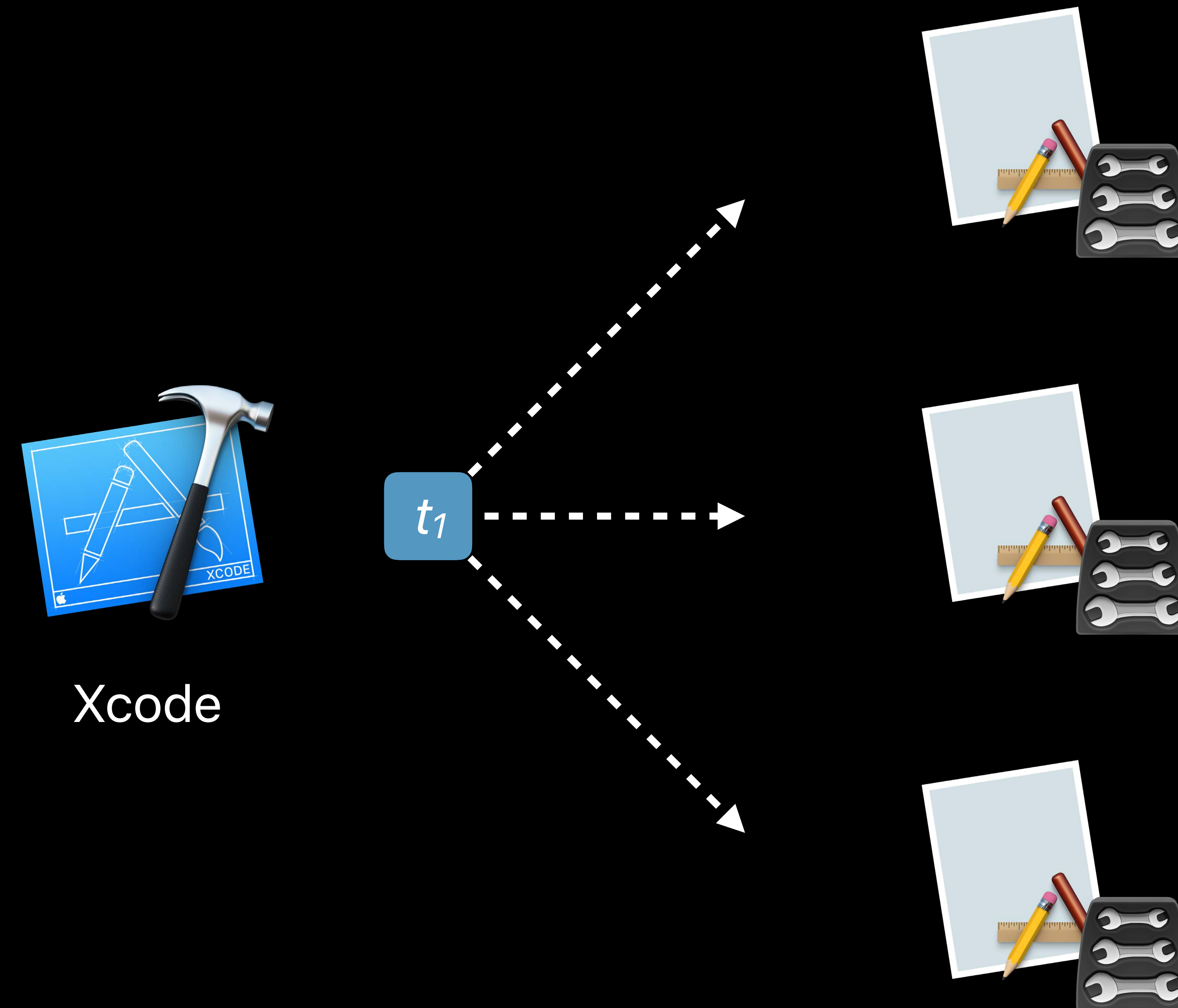


Xcode



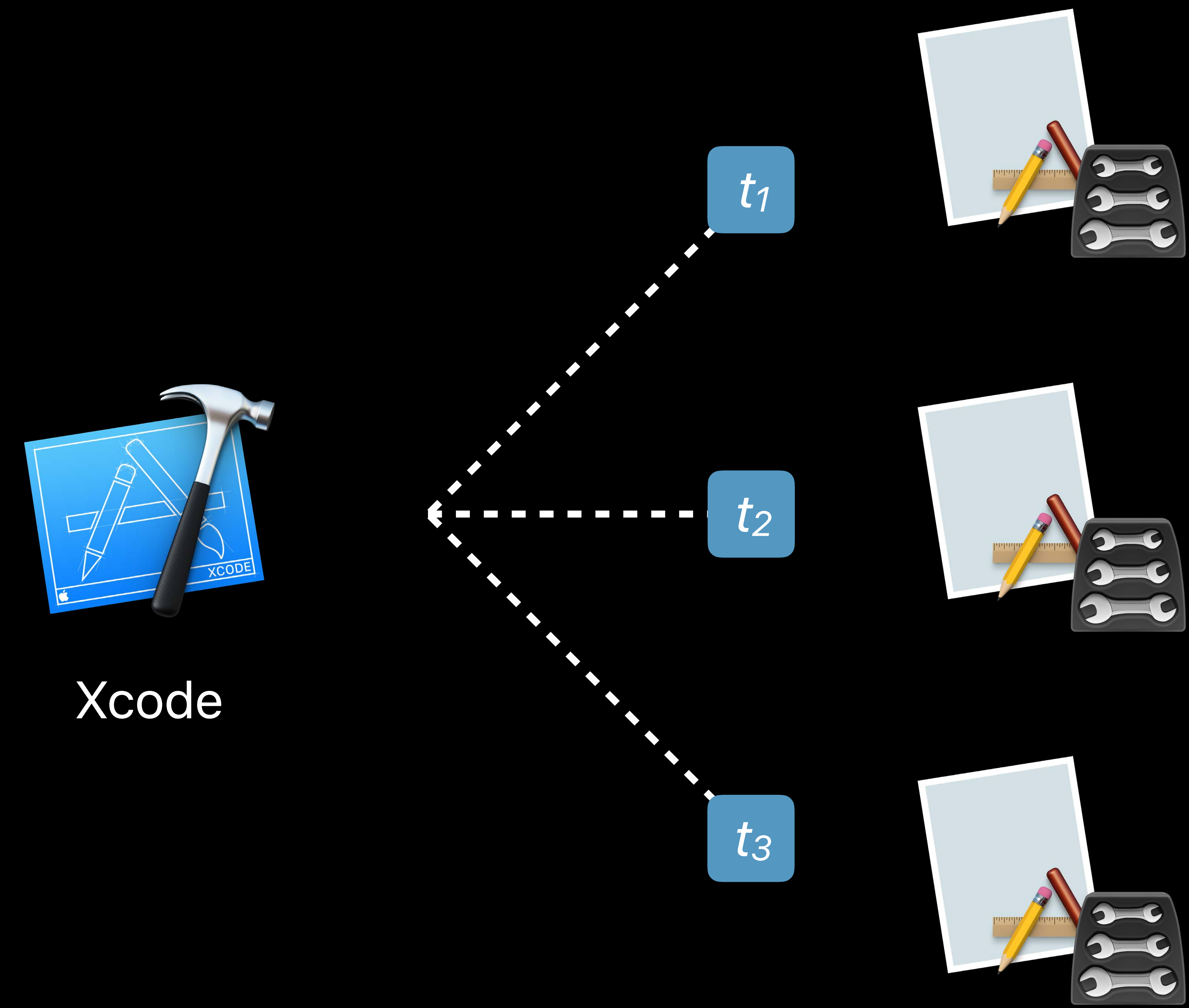
# Overview

## Parallel Distributed Testing



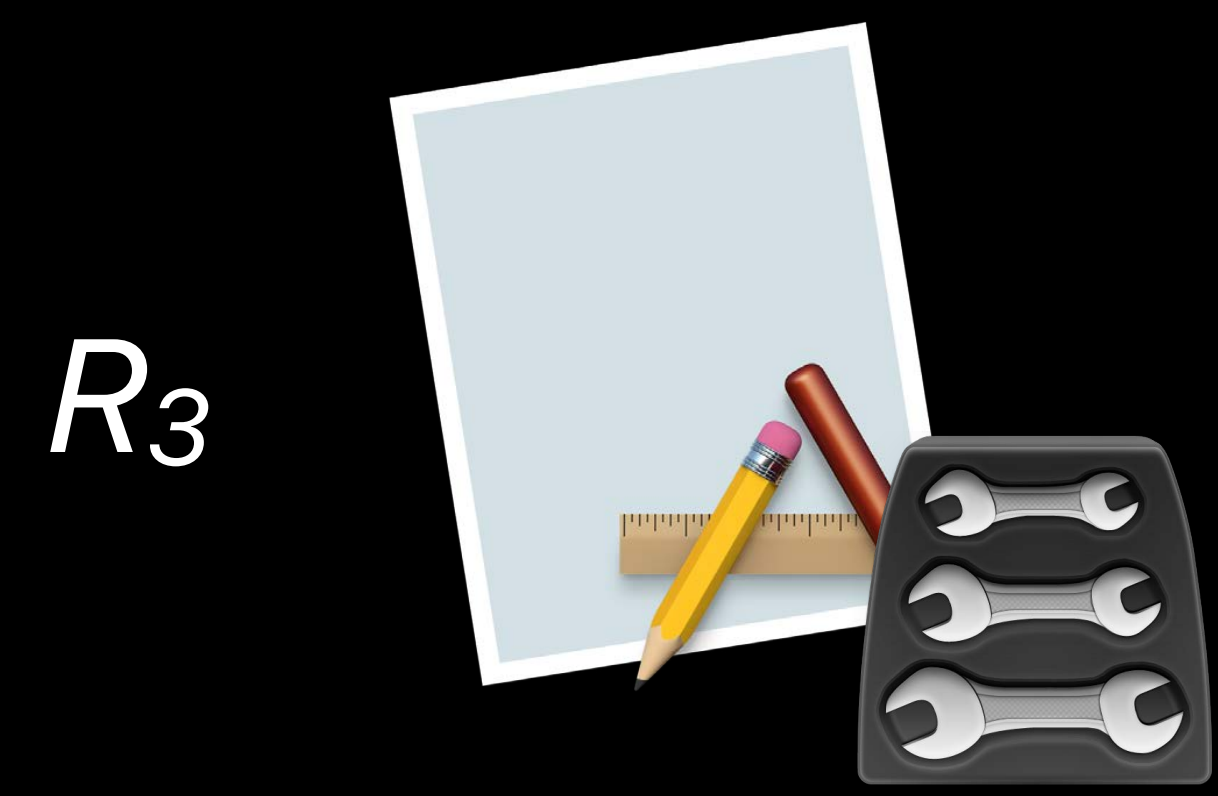
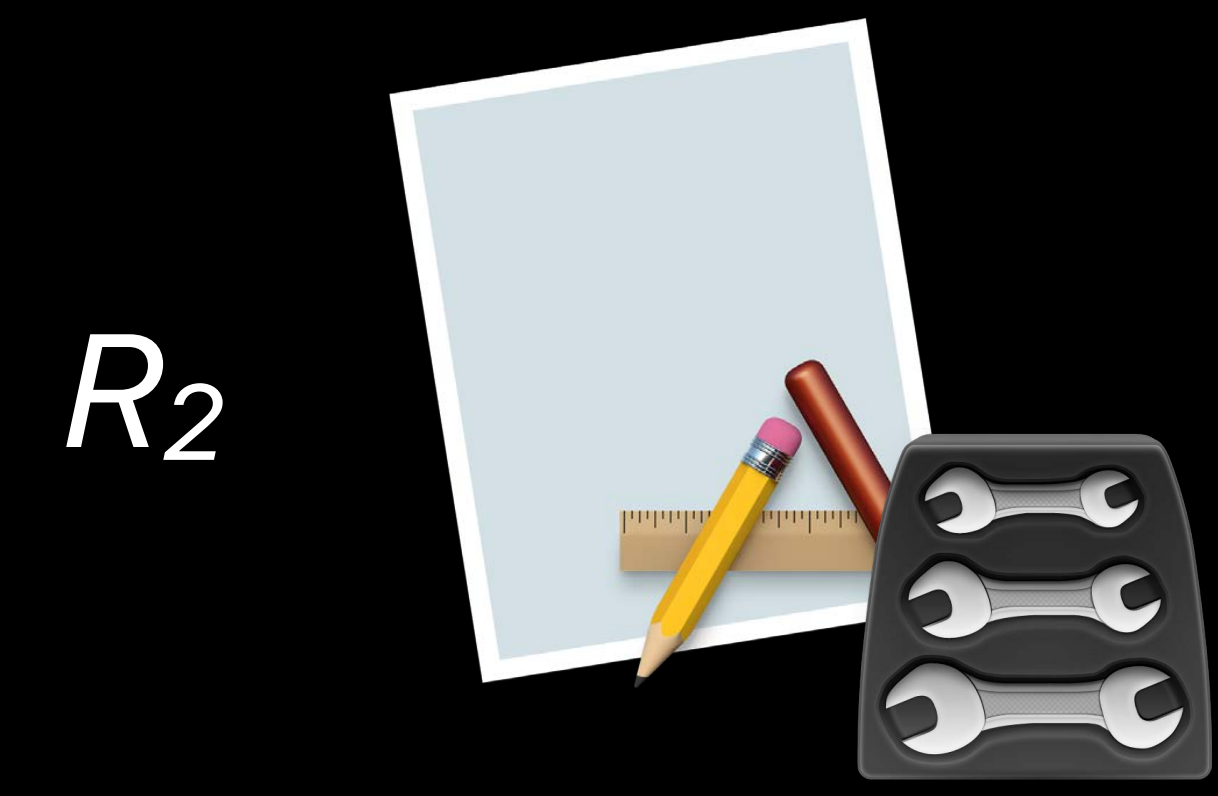
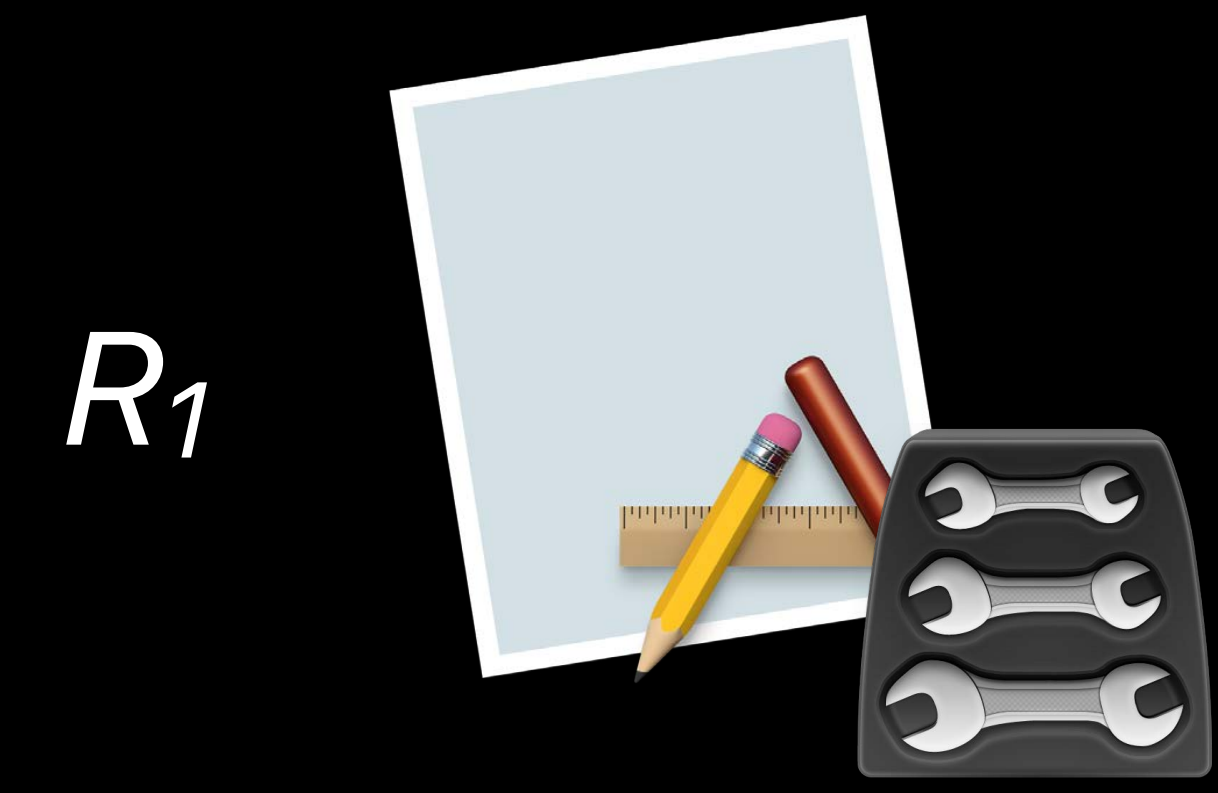
# Overview

## Parallel Distributed Testing



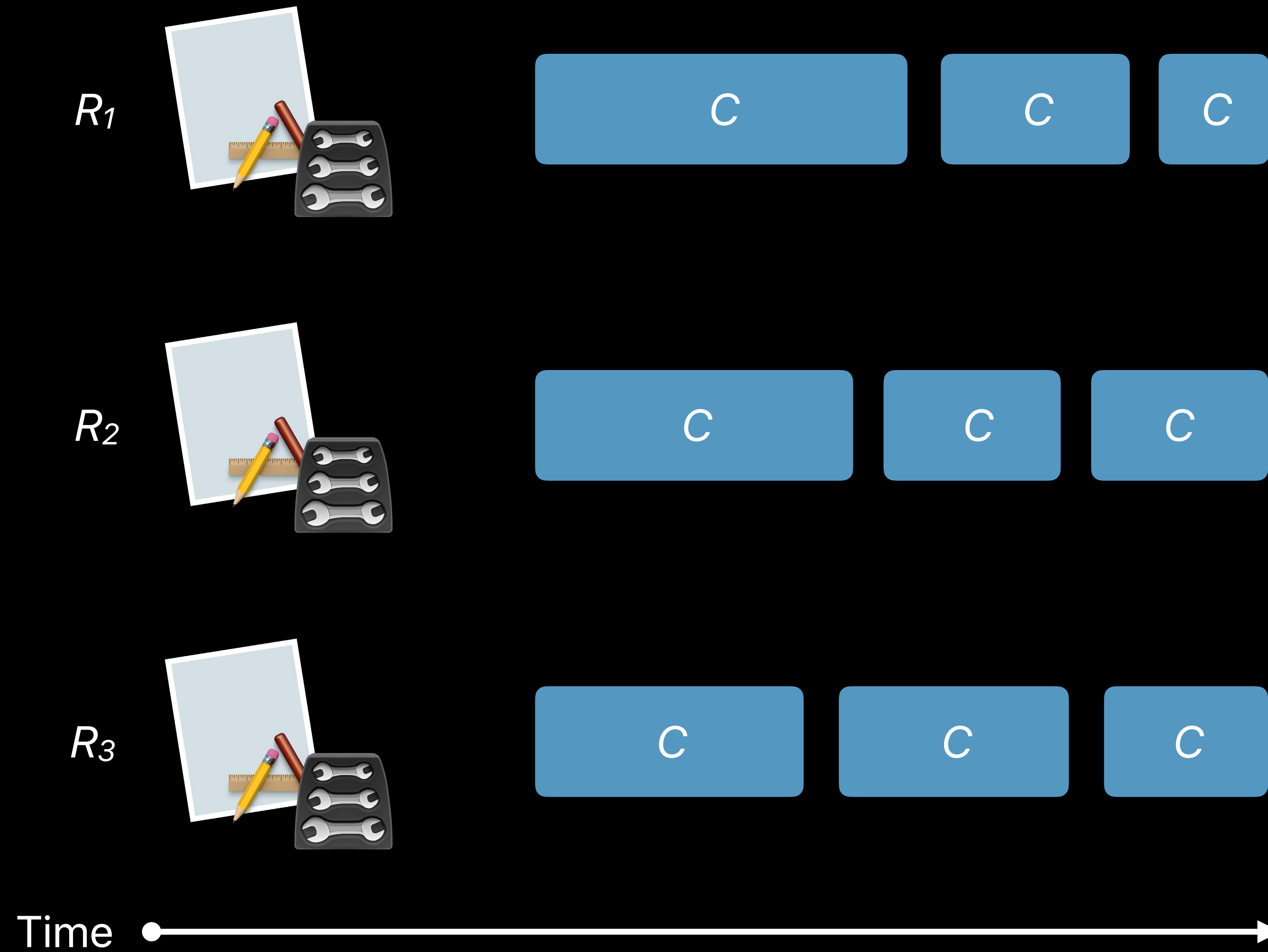


# Classes Execute in Parallel



Time 

# Classes Execute in Parallel



# Classes Execute in Parallel

Motivation



# Classes Execute in Parallel

Motivation

Hidden dependencies between tests in a class

# Classes Execute in Parallel

## Motivation

Hidden dependencies between tests in a class

Avoid unnecessary `+setUp` and `+tearDown` computation

# Parallel Testing on Simulator



Original  
Simulator

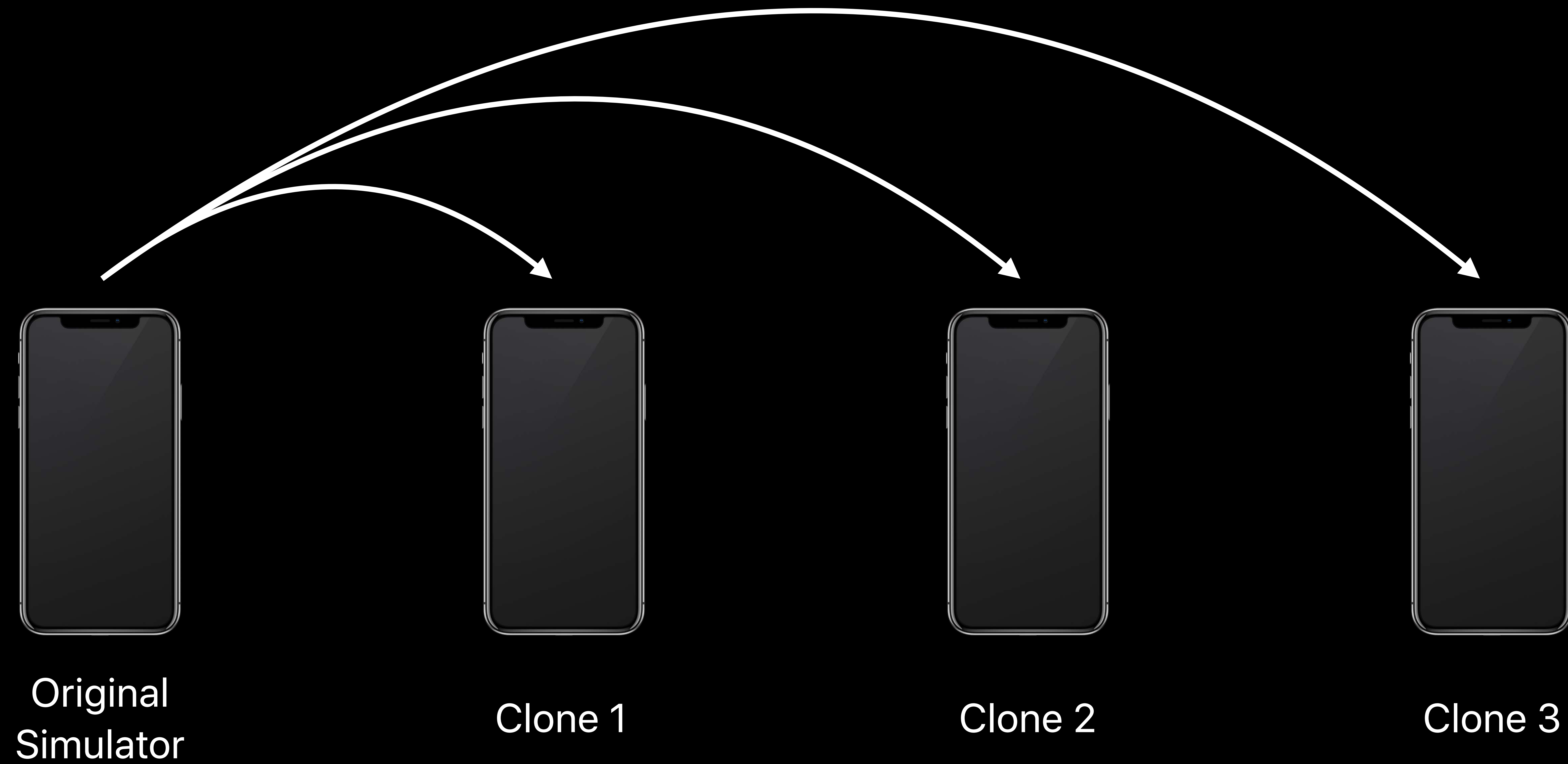


# Parallel Testing on Simulator

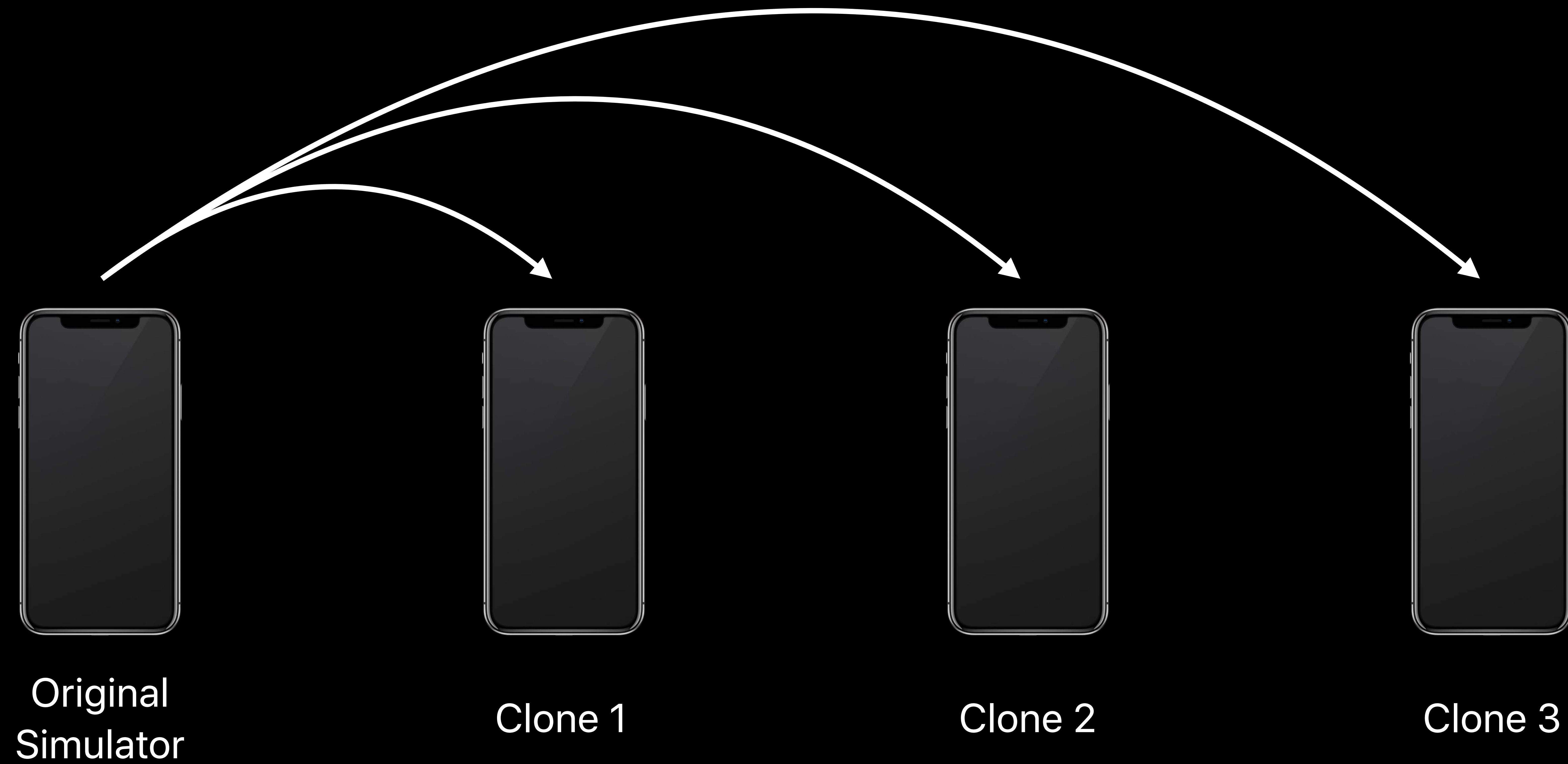


Original  
Simulator

# Parallel Testing on Simulator

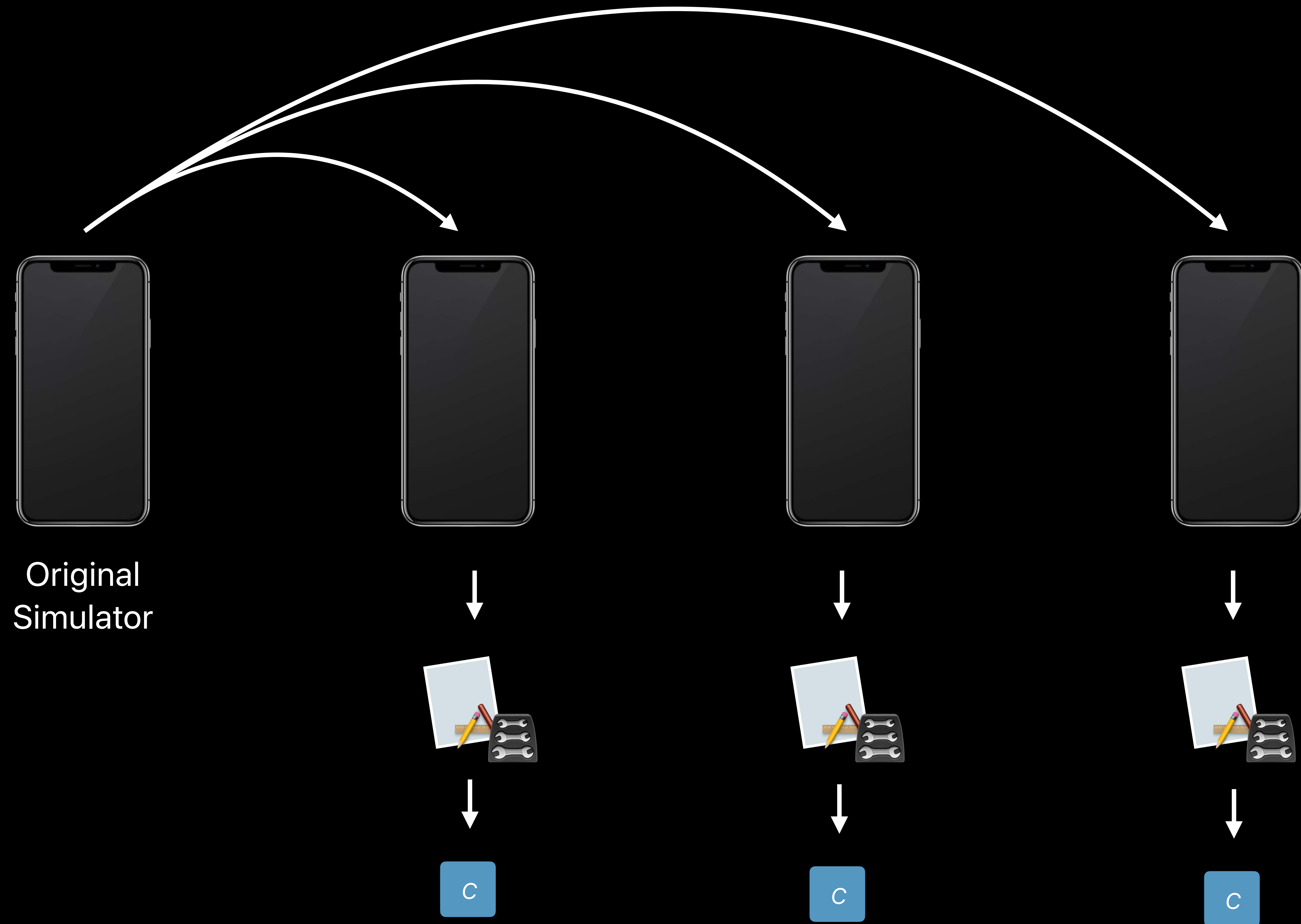


# Parallel Testing on Simulator





# Parallel Testing on Simulator



# Parallel Testing on Simulator

Implications

# Parallel Testing on Simulator

Implications

Original simulator is not used during testing



# Parallel Testing on Simulator

## Implications

Original simulator is not used during testing

Separate data containers

# Supported Configurations

	Unit Tests	UI Tests
macOS	✓	
Simulator (iOS and tvOS)	✓	✓

***Demo***



# Demo Recap

# Demo Recap

Enabling parallelization

# Demo Recap

Enabling parallelization

Viewing results in the test log and test report



# Demo Recap

Enabling parallelization

Viewing results in the test log and test report

Multiple instances of a Mac app running unit tests

# Demo Recap

Enabling parallelization

Viewing results in the test log and test report

Multiple instances of a Mac app running unit tests

Multiple simulator clones running UI tests

**xcodebuild**



# xcodebuild

Override the number of workers

```
-parallel-testing-worker-count n
```

# xcodebuild

Override the number of workers

```
-parallel-testing-worker-count n
```

Force parallel testing on or off

```
-parallel-testing-enabled YES | NO
```

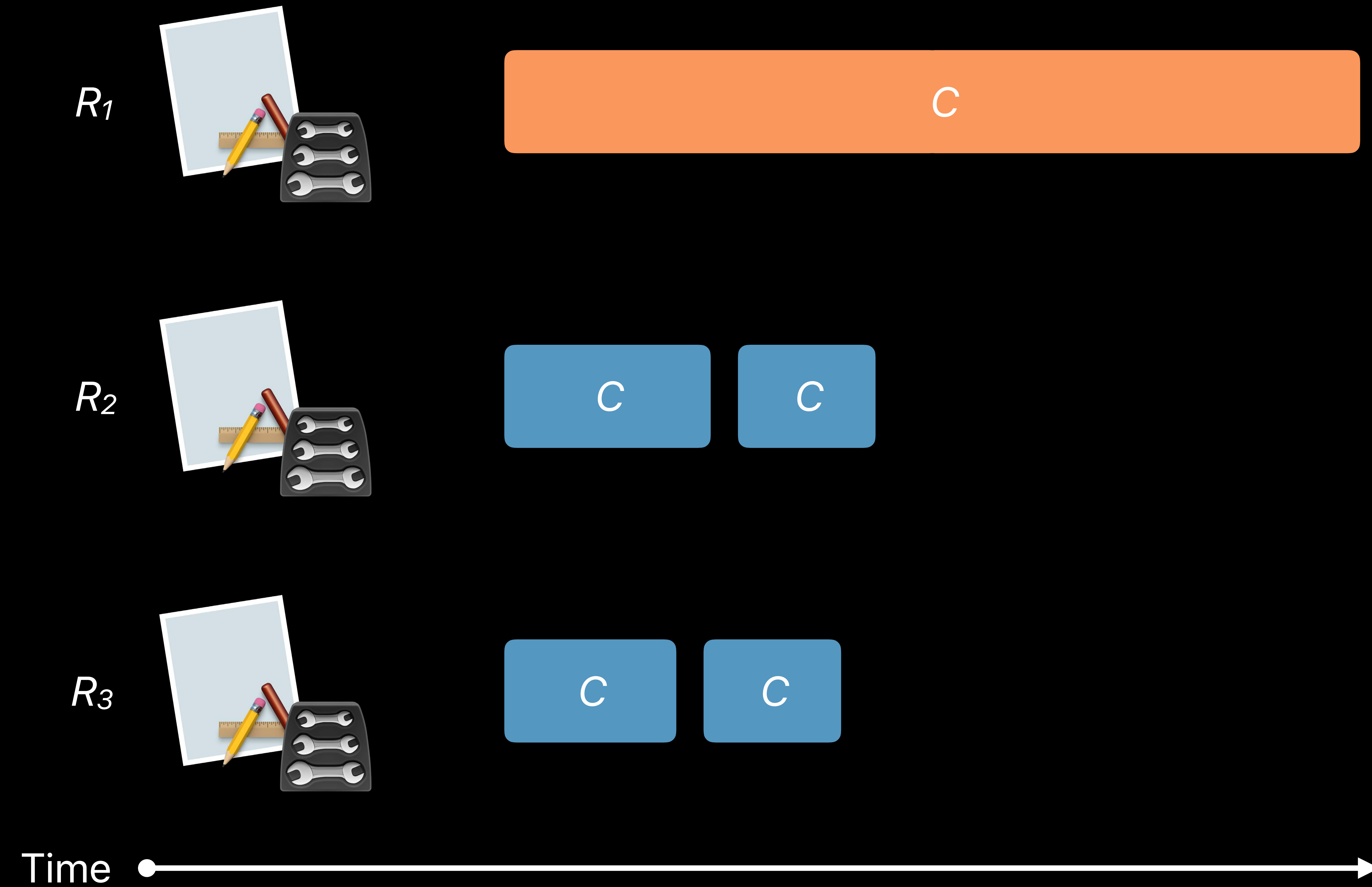
# Tips and Tricks



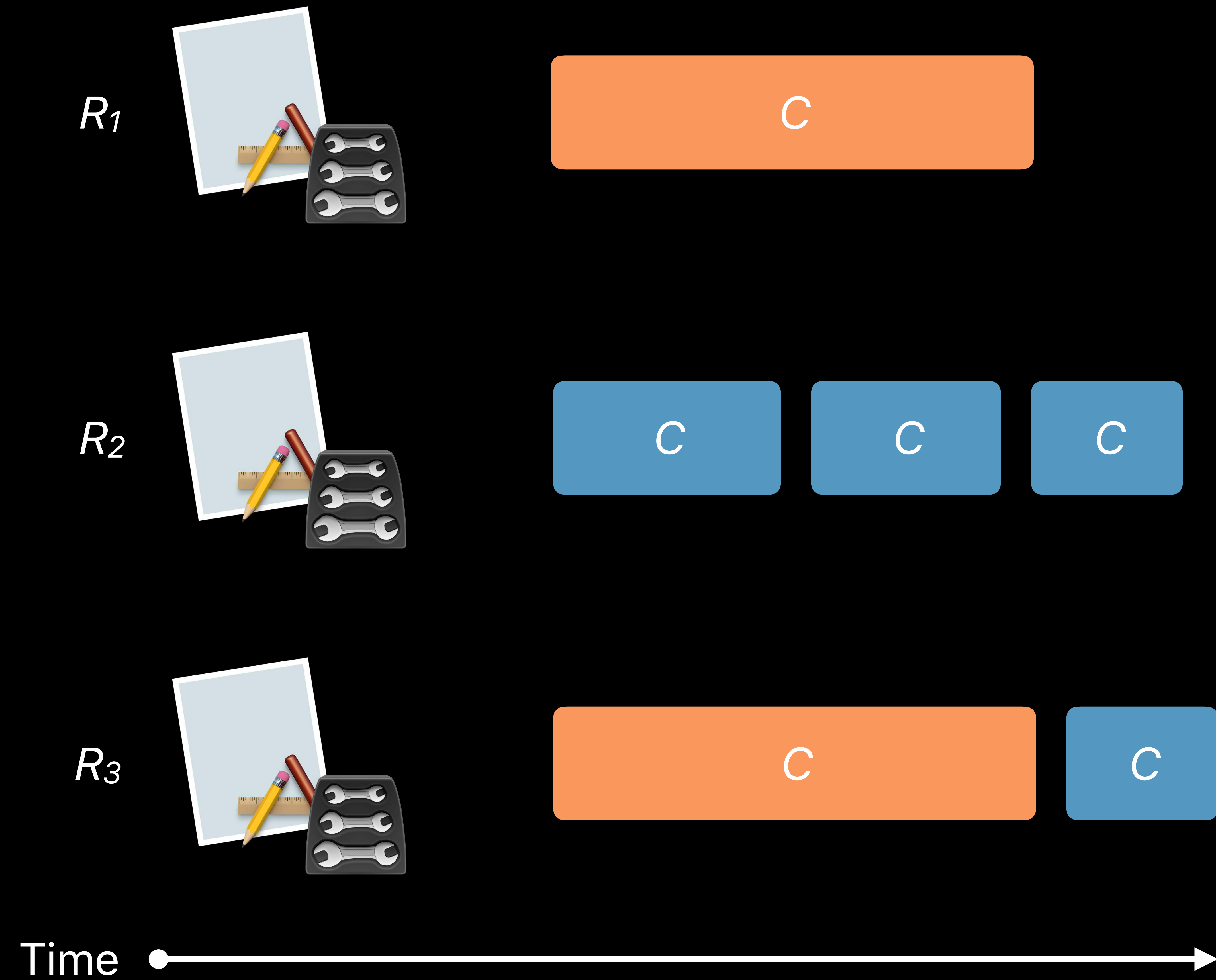
# Tips and Tricks

Consider splitting a long running class into two classes

# Classes Execute in Parallel



# Classes Execute in Parallel





# Tips and Tricks

Consider splitting a long running class into two classes

# Tips and Tricks

Consider splitting a long running class into two classes

Put performance tests into their own bundle, with parallelization disabled

# Tips and Tricks

Consider splitting a long running class into two classes

Put performance tests into their own bundle, with parallelization disabled

Understand which tests are not safe for parallelization



# Tips and Tricks

Consider splitting a long running class into two classes

Put performance tests into their own bundle, with parallelization disabled

Understand which tests are not safe for parallelization

# Summary

# Summary

Code coverage



# Summary

Code coverage

Test selection and ordering

# Summary

Code coverage

Test selection and ordering

Parallel testing

# More Information

<https://developer.apple.com/wwdc18/403>

---

Simulator, Testing, and Continuous Integration Lab

Technology Lab 8

Wednesday 12:00PM

---

Simulator, Testing, and Continuous Integration Lab

Technology Lab 8

Friday 9:00AM

---



 **WWDC18**