Building Confidence

Testing your Apps

Saul Mora - saul.mora@liulishuo.com
@casademora

liulishuo.com/joinus

Testing Sucks

Testing Sucks

单元测试弱爆了！

Big Time

真的弱爆了！

Slow

拖累开发进度！

Tedious

烦琐！

Boring

无聊！

QA

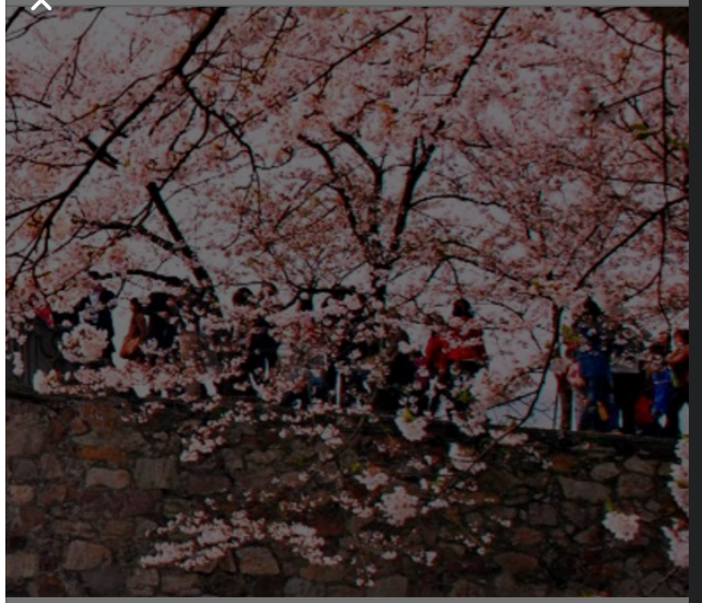
Indies

Twitter, Inc. twitter.com/PDChina/status/651418300665188352?ref_src=twsrc%

Home Moments

Search Twitter

Have an account? Log in



People's Daily,China

@PDChina

The largest newspaper group in China; Timely updates
youtube.com/c/peoplesdaily
fb.me/PeoplesDaily

Beijing, China

en.people.cn

Joined May 2011

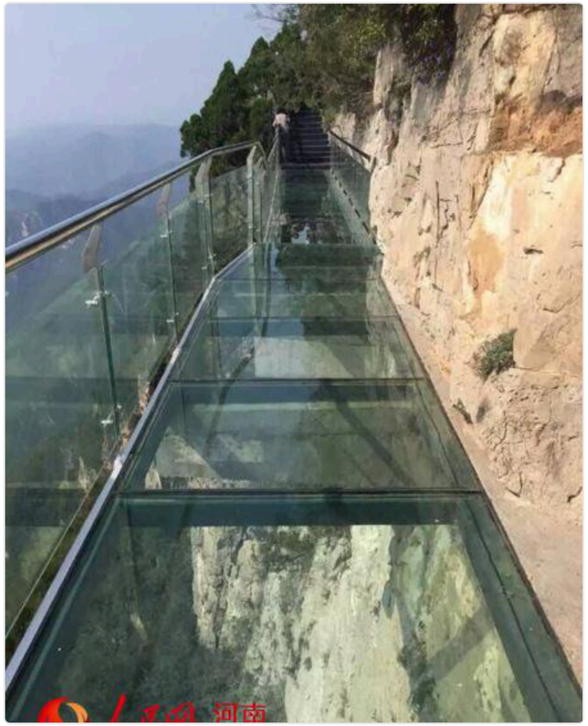
© 2017 Twitter About Help Center Terms
Privacy policy Cookies Ads info

People's Daily,China

@PDChina

Follow

Yuntaishan over-cliff glass bridge cracked in C China Oct. 5, causing panic among visitors en.people.cn/n/2015/1006/c9…



RETWEETS 103 LIKES 42

8:26 AM - 6 Oct 2015

15 103 42

Toby J @tobyj_official · 6 Oct 2015


Twitter, Inc. twitter.com/PDChina/status/651418300665188352?ref_src=twsrc%
Home
Moments
Search Twitter
Have an account? Log in
People's Daily,China
@PDChina
Follow
Yuntaishan over-cliff glass bridge cracked in C
China Oct. 5, causing panic among visitors
en.people.cn/n/2015/1006/c9…
People's Daily,China
@PDChina
Follow
Yuntaishan over-cliff glass bridge cracked in C
China Oct. 5, causing panic among visitors
en.people.cn/n/2015/1006/c9…
Timely updates
youtube.com/c/peoplesdaily
fb.me/PeoplesDaily
Beijing, China
en.people.cn
Joined May 2011
RETWEETS
103
LIKES
42
8:26 AM - 6 Oct 2015
15
103
42
Toby J @tobyj_official · 6 Oct 2015

Twitter, Inc. twitter.com/inhabitat/status/651747112795947008?ref_src=twsrc%

Home Moments

Have an account? Log in

inhabitat
@inhabitat

Follow

Oh hell no. Crazy glass bridge 3,543 feet above ground cracks in #China. With people on it. bit.ly/1PieNQX

RETWEETS 12 LIKES 7

6:13 AM - 7 Oct 2015

3 12 7

cyn @Wary12 · 7 Oct 2015
Replying to @inhabitat
@inhabitat Duh! that was waiting to happen!
1 1

Akaa @Grabben__ · 7 Oct 2015
Replying to @inhabitat
@inhabitat lol, made in china!
2

The Fabric House @FabricHouseCT · 7 Oct 2015
Replying to @inhabitat
@inhabitat 😢

inhabitat
@inhabitat
good design will save the world!
everywhere
inhabitat.com
Joined March 2008

inhabitat
@inhabitat

Follow

Oh hell no. Crazy glass bridge 3,543 feet above ground cracks in #China. With people on it. bit.ly/1PieNQX

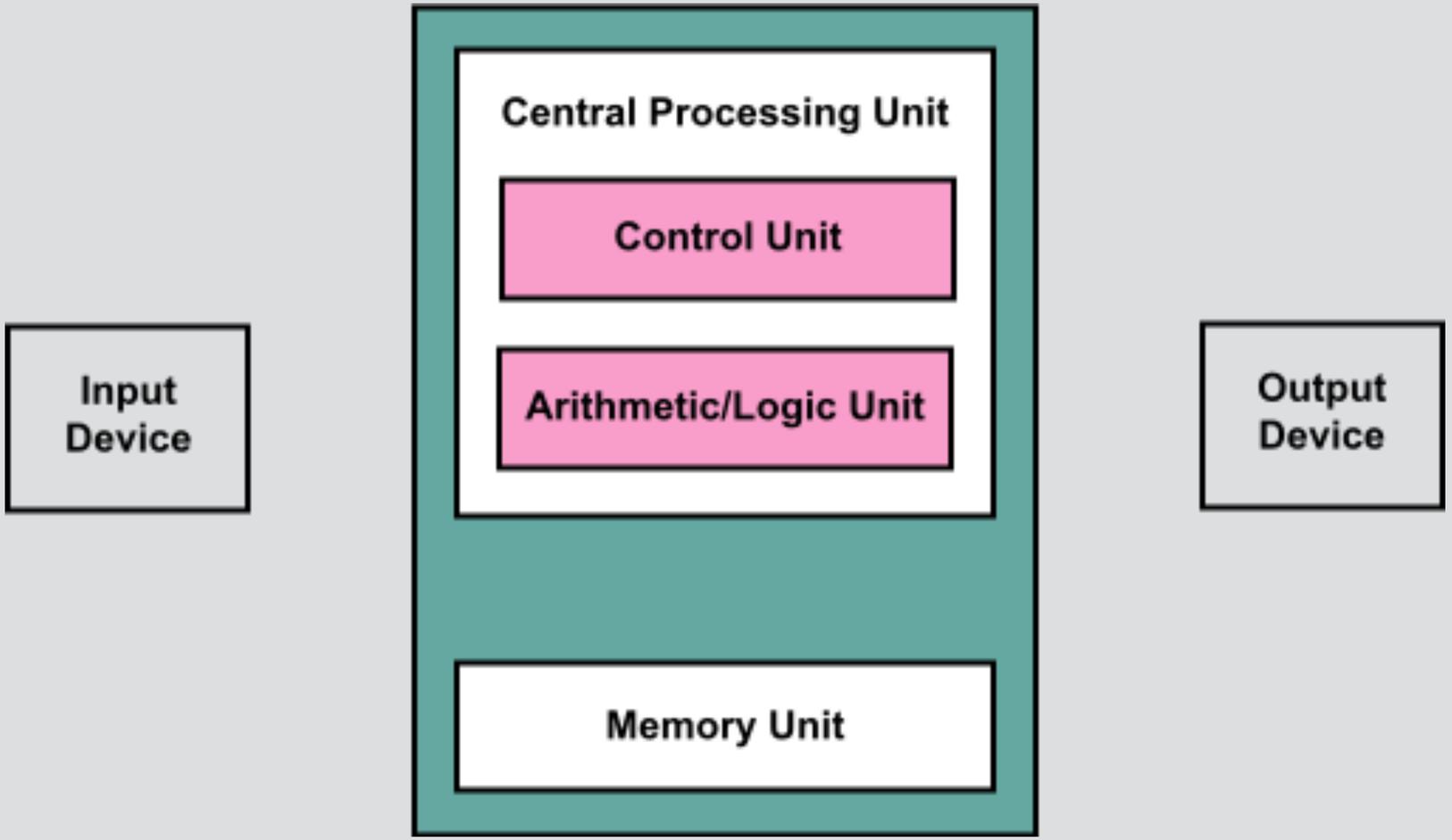
If tests are so important...

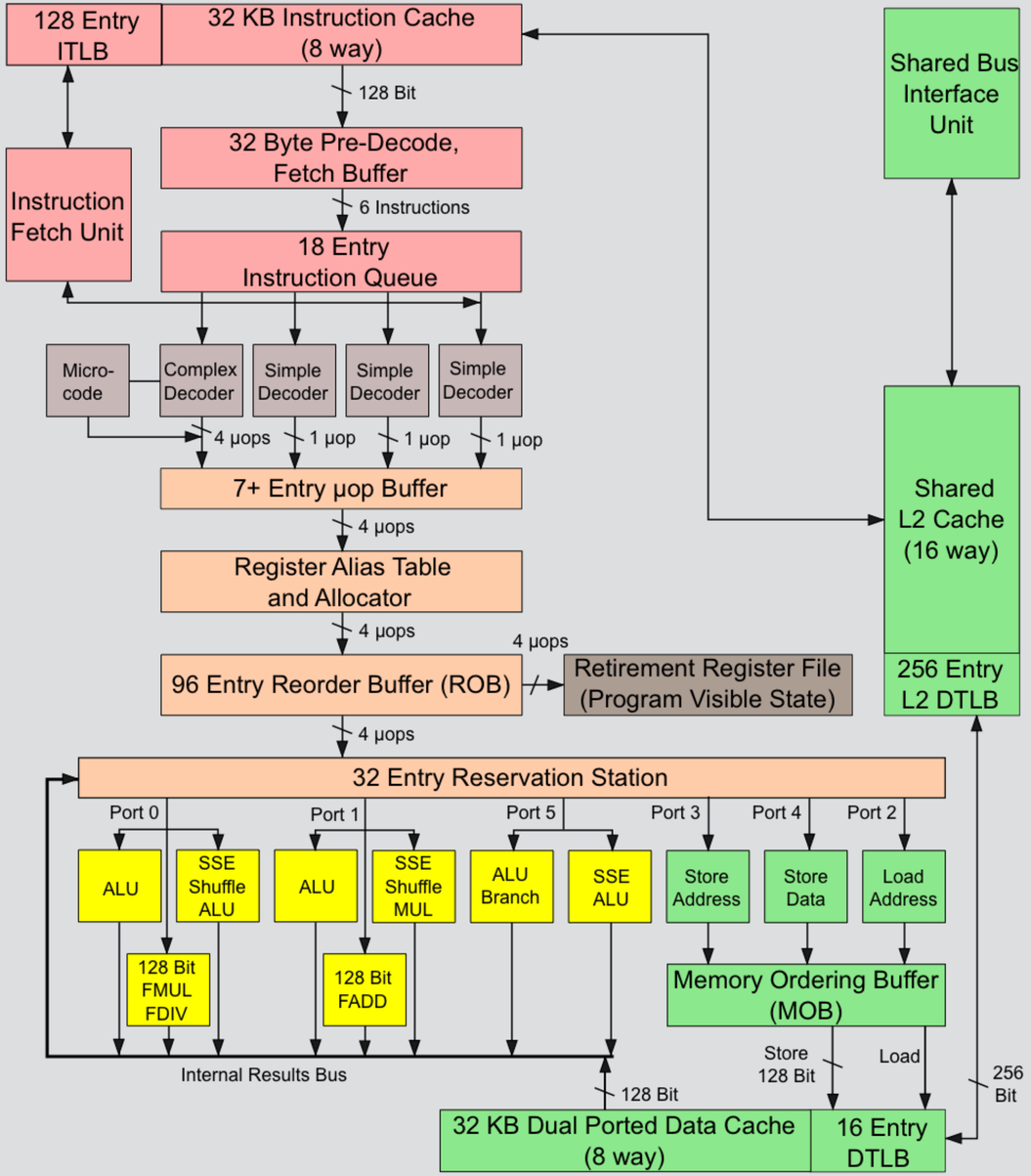
Why does testing suck?

Writing tests is hard

Think Different

About Testing

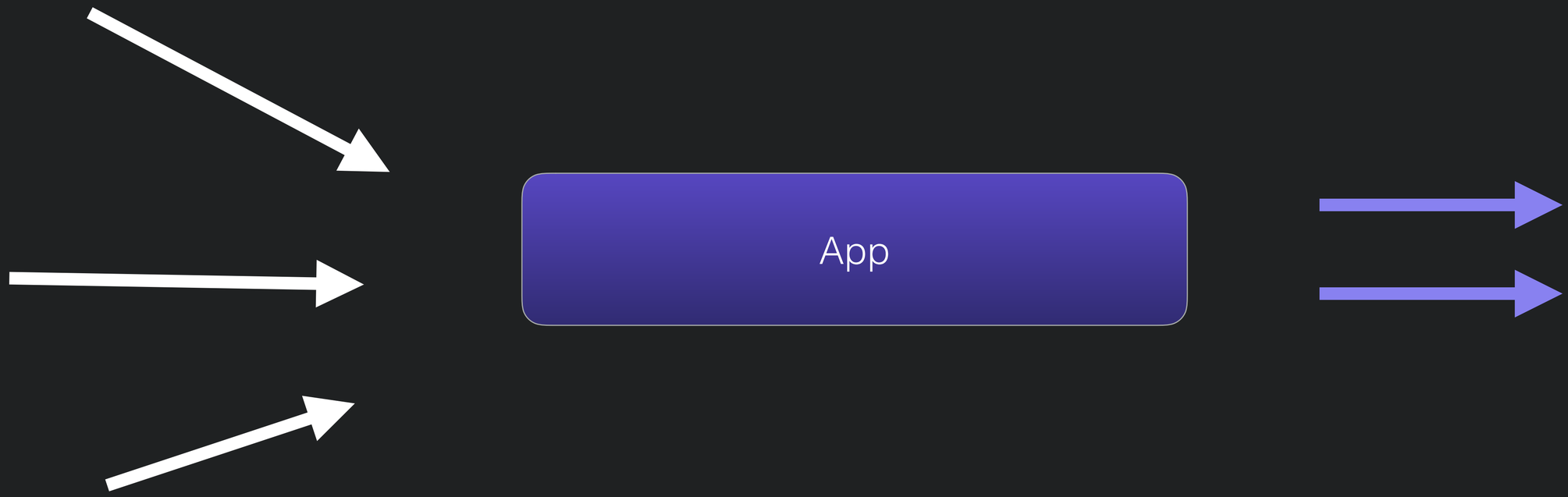

Central Processing Unit
Control Unit
Arithmetic/Logic Unit
Input Device
Output Device
Memory Unit



Intel Core 2 Architecture

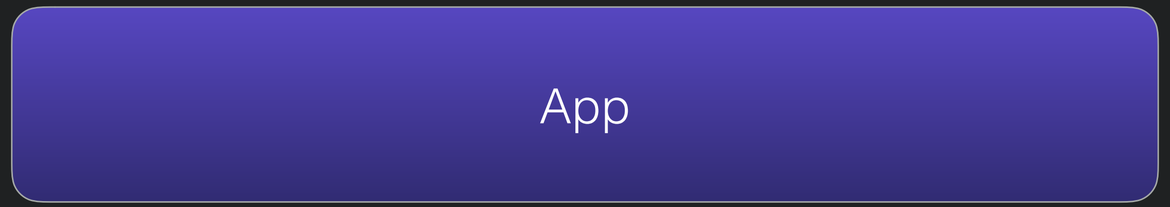
Inputs

Outputs



Taps, Notifications, Data

Animations, Sound, Data


App

A single function is easy to test!

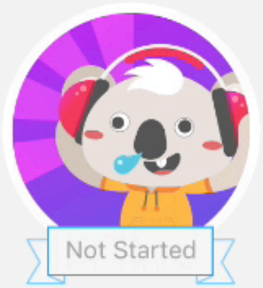
- Me just now

- 我刚说了

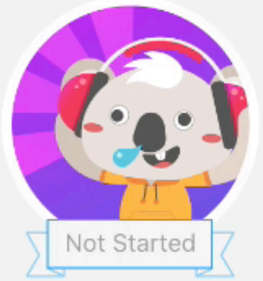
Carrier 6:24 PM

Practice

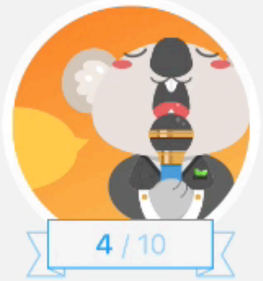
听小对话回答问题



听较长对话回答问题

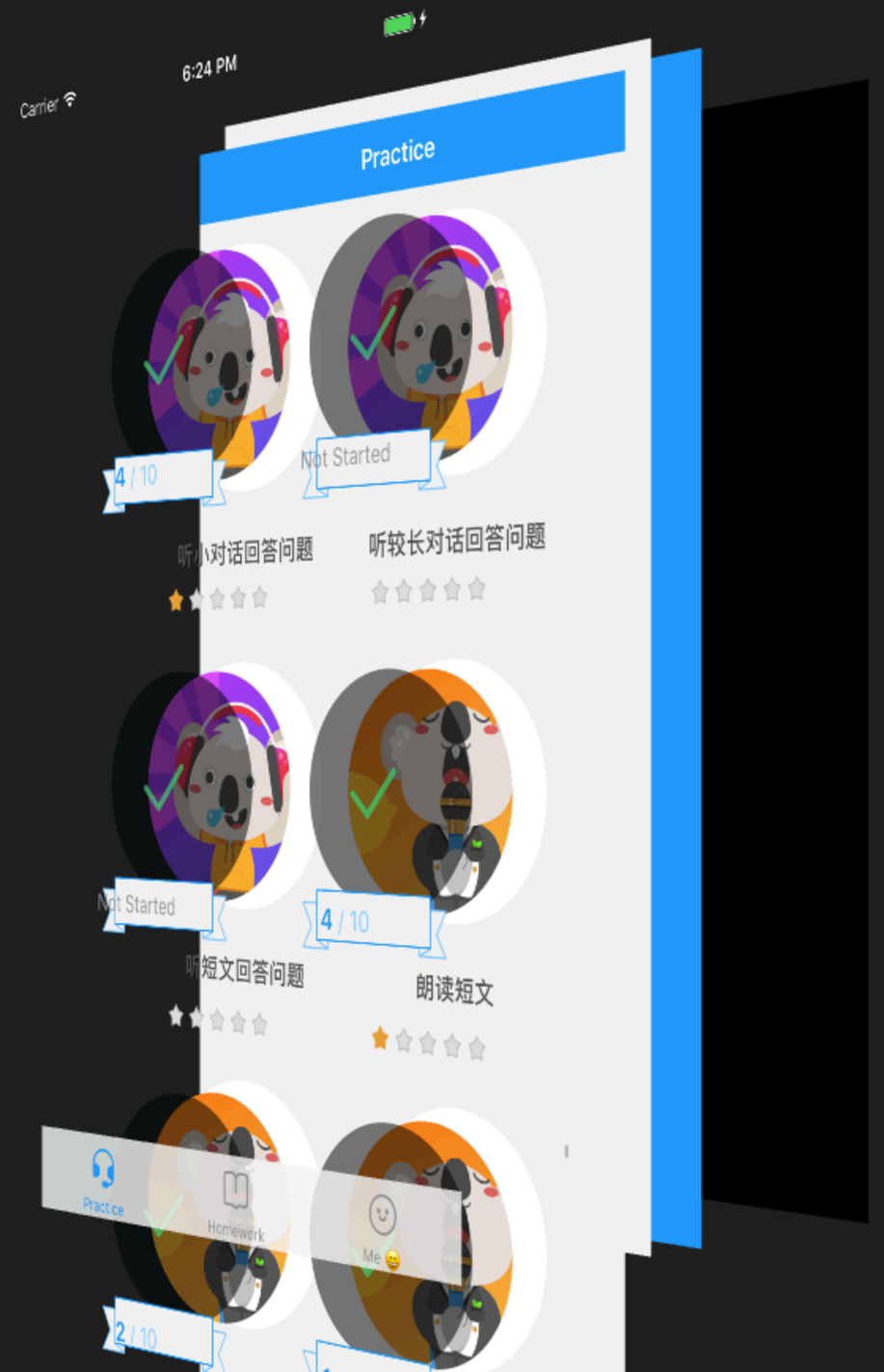


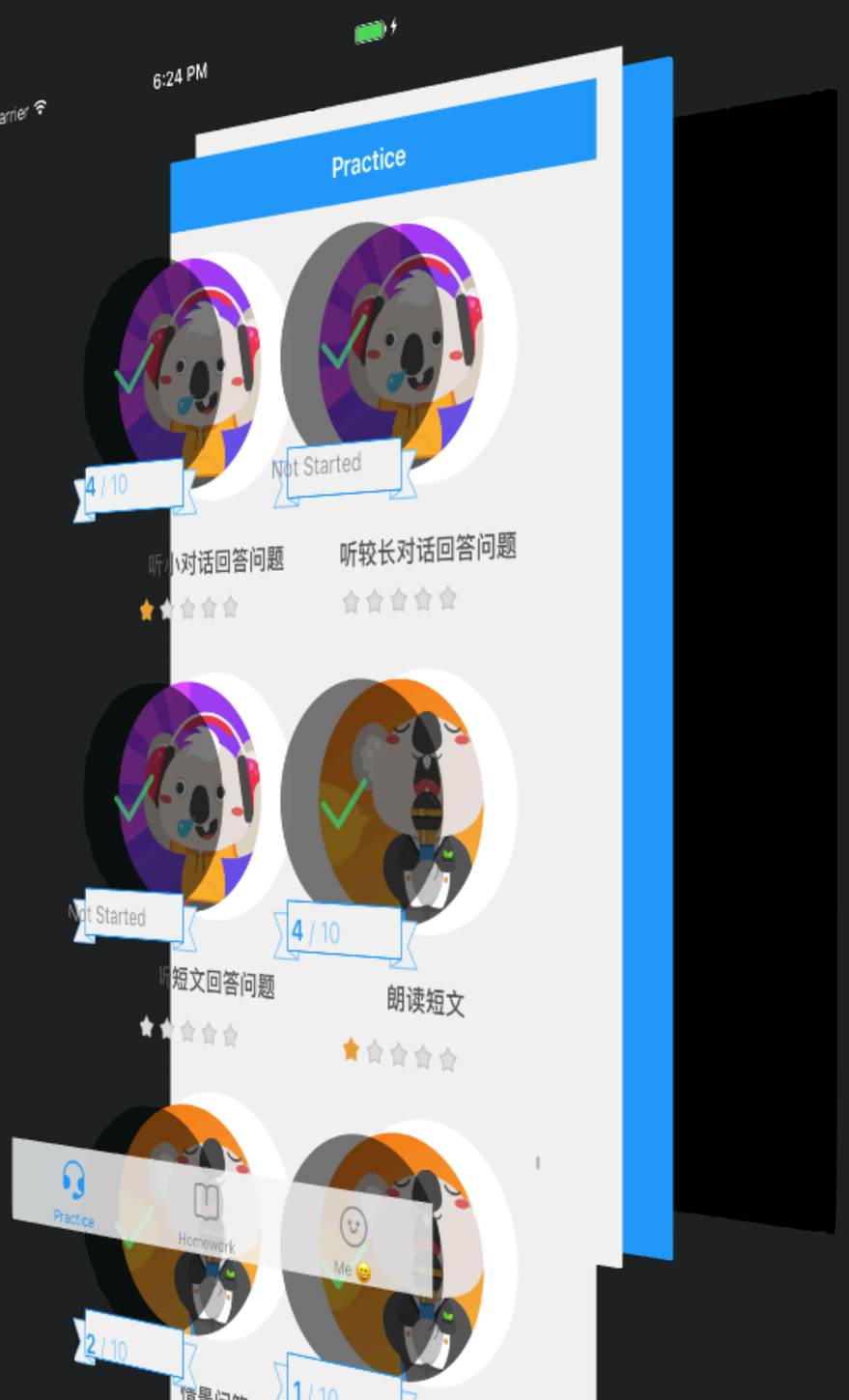
听短文回答问题

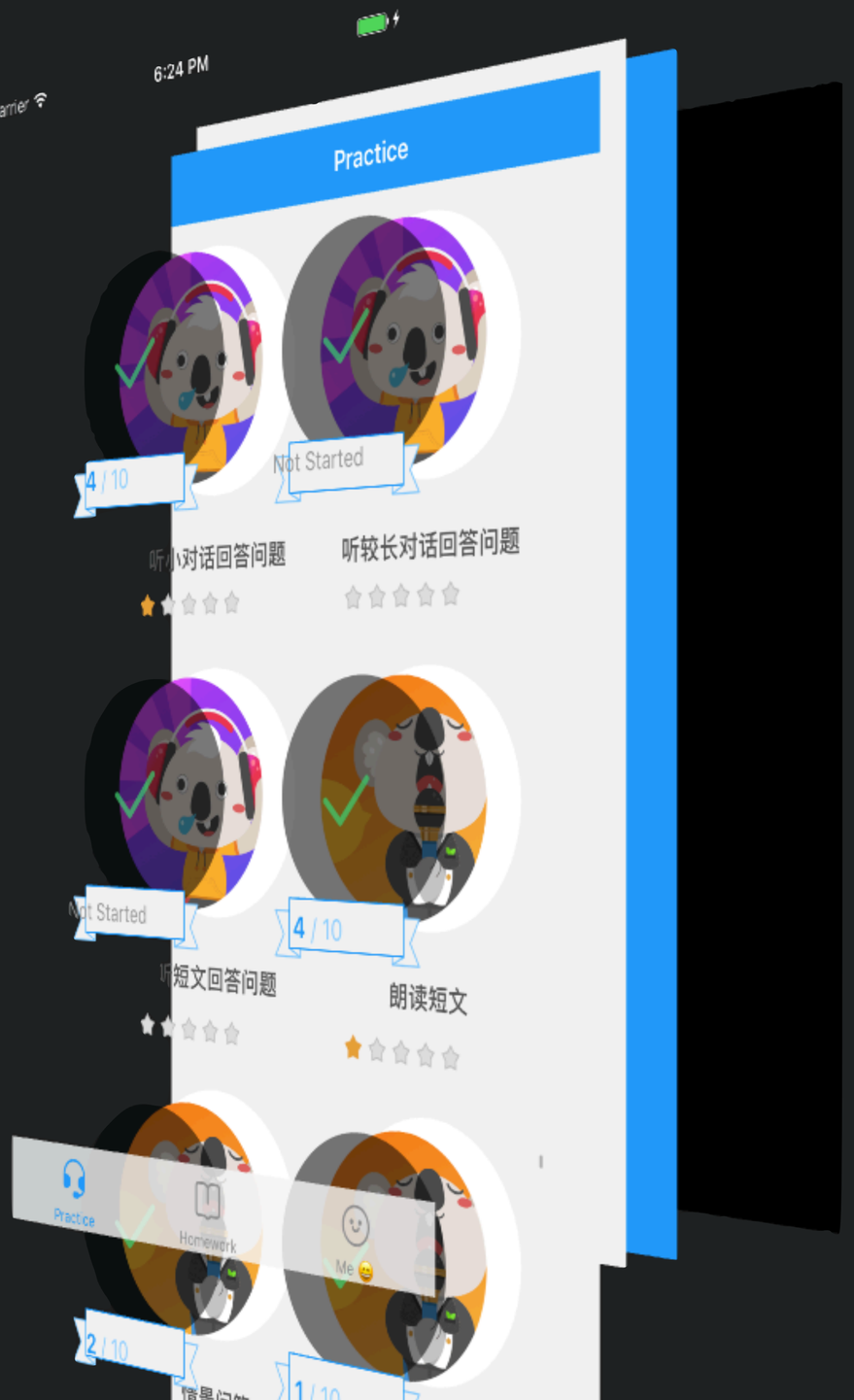


朗读短文

Practice Homework Me


Carrier
6:24 PM
Practice
4 / 10
Not Started
听较长对话回答问题
Not Started
4 / 10
朗读短文
Practice
Homework
Me
2 / 10


Carrier
6:24 PM
Practice
4 / 10
Not Started
听较长对话回答问题
Not Started
4 / 10
朗读短文
Practice
Homework
Me
2 / 10
1 / 10



UIView

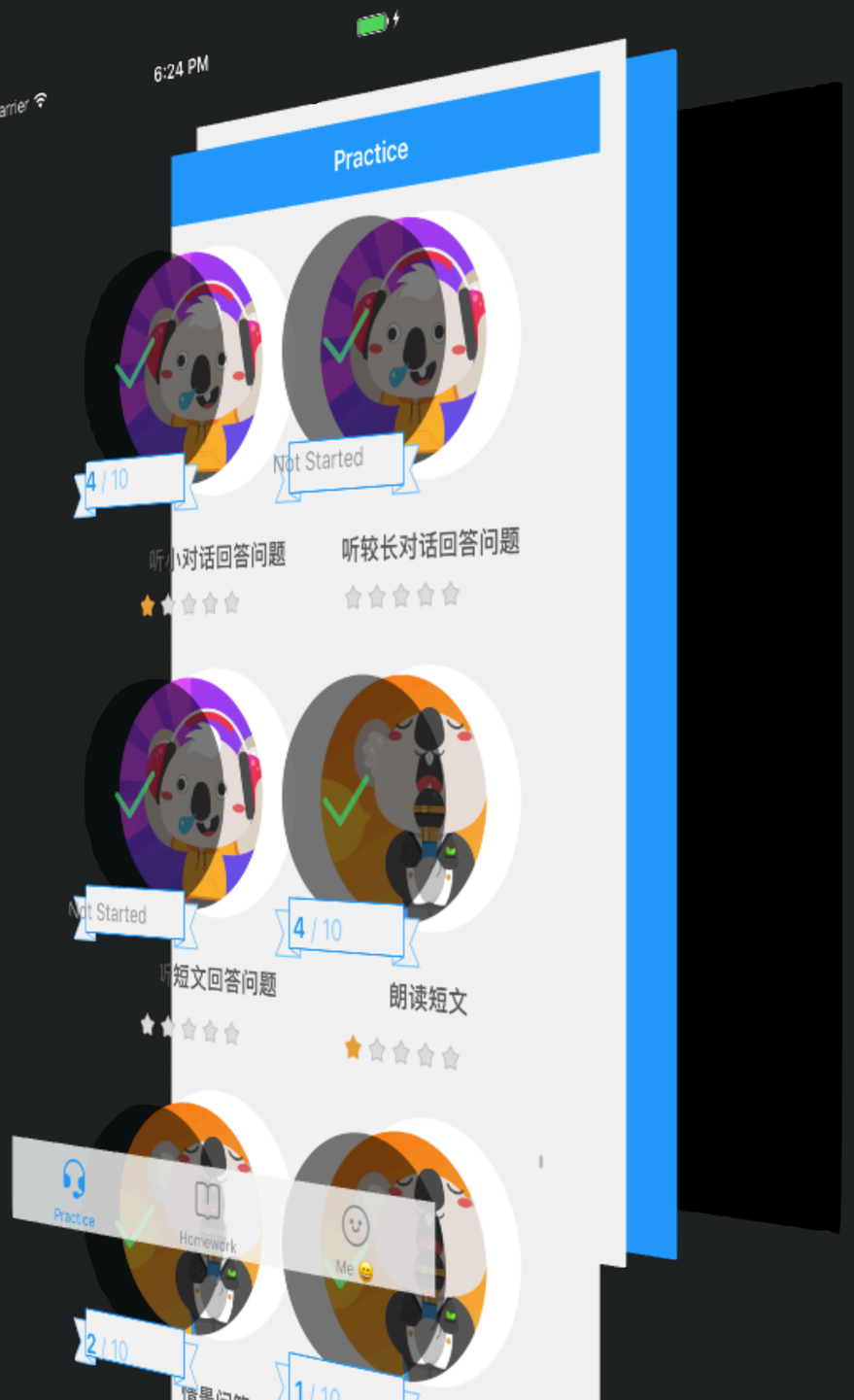
UIViewController

UICollectionViewCell

UICollectionViewDataSource

Disk Cache

Network



UIView

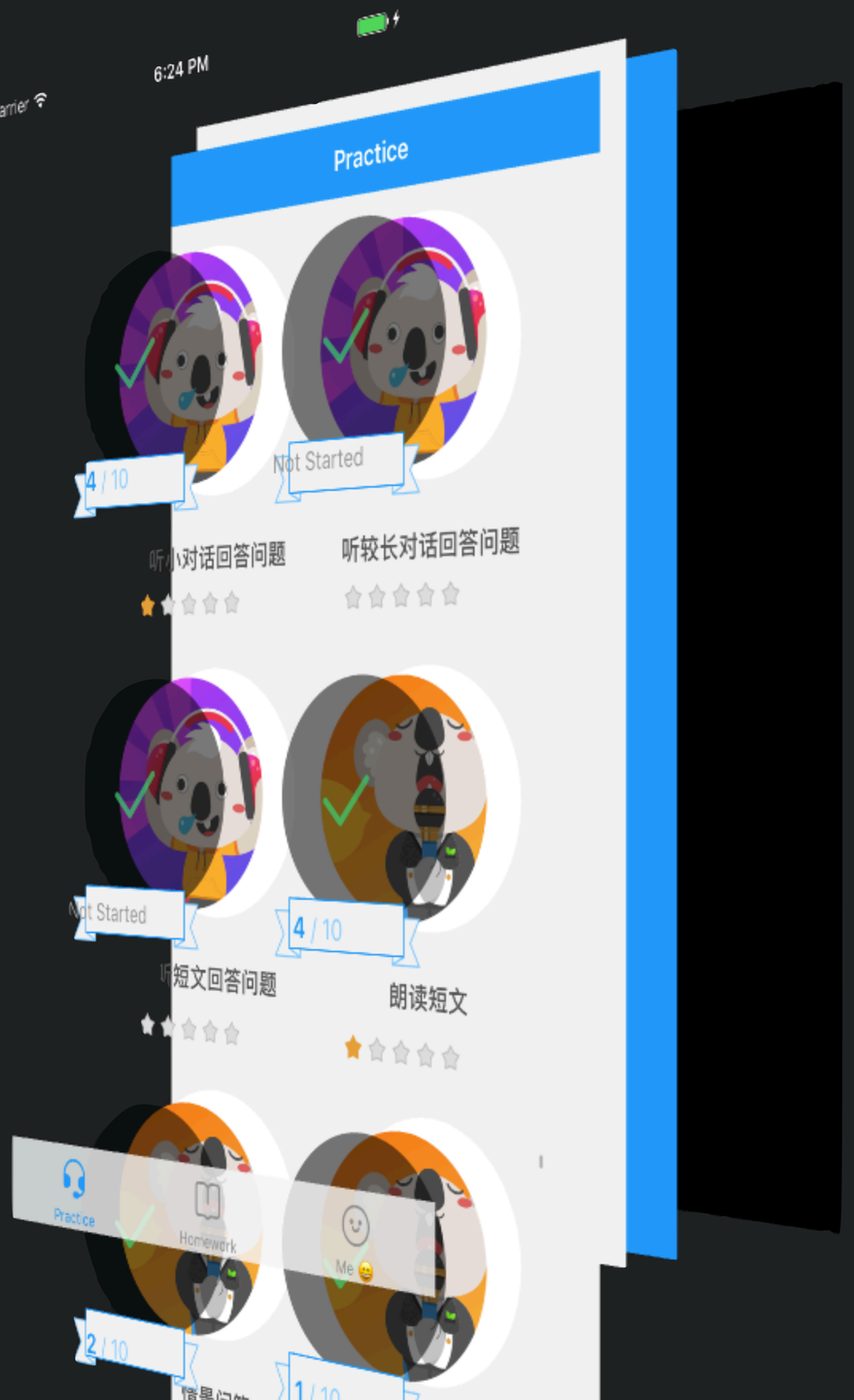
UIViewController

UICollectionViewCell

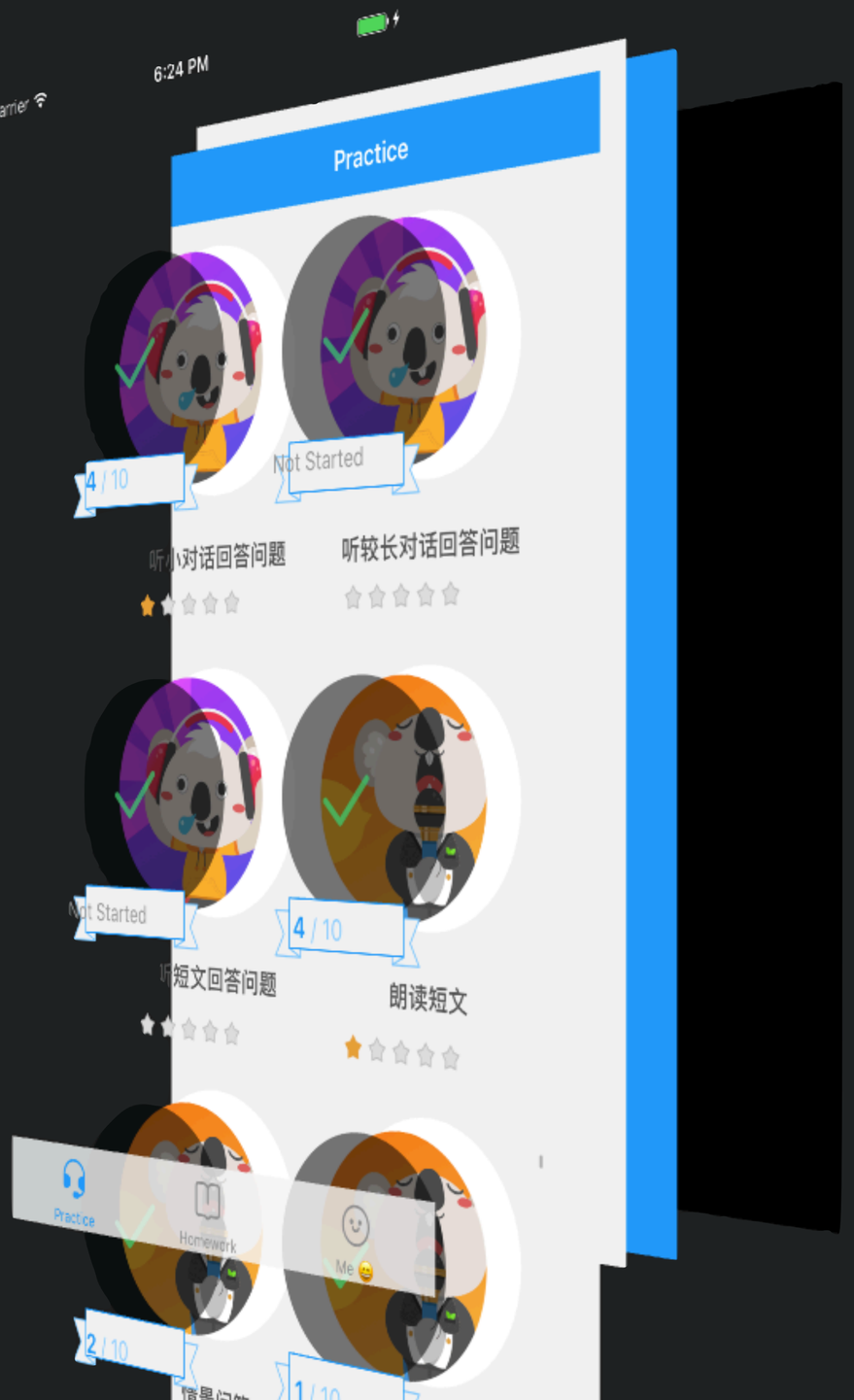
UICollectionViewDataSource

Disk Cache

Network


Carrier
6:24 PM
Practice
4 / 10
Not Started
听较长对话回答问题
Not Started
4 / 10
朗读短文
Practice
Homework
Me
2 / 10
1 / 10

Network



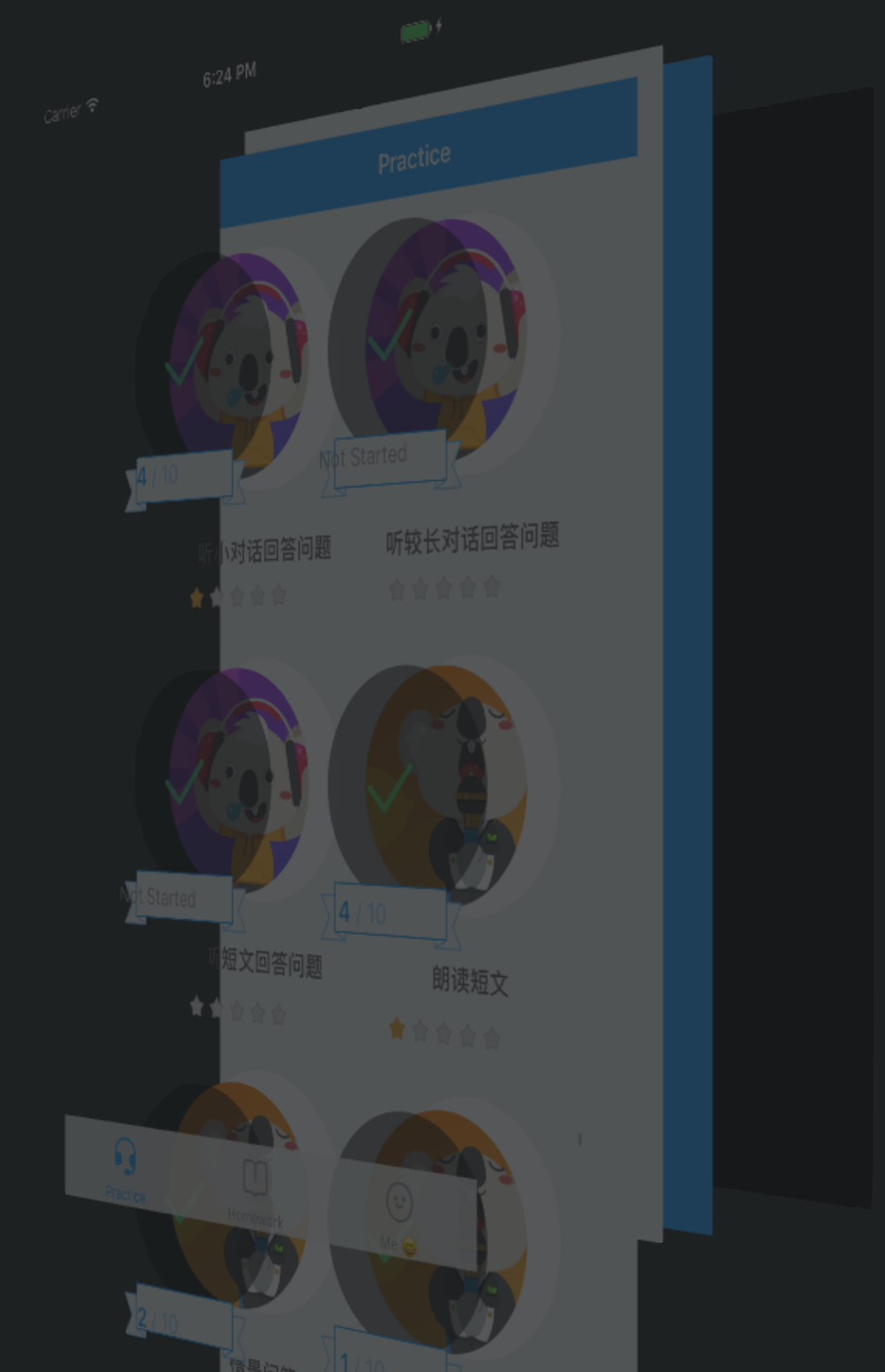
Network

HTTP

URLSession

JSON Parsing

Reachability



Network

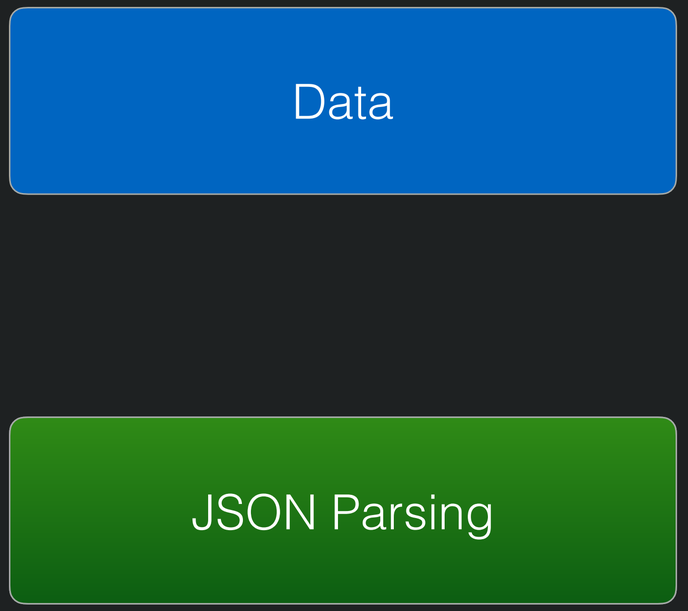
HTTP

JSON Parsing

Session

Reachability

JSON Parsing


Data
JSON Parsing

JSON Parsing

```
func parse<T>(data: Data) -> T where T: JSONDecodable
```

```
{
  "name": "Saul Mora",
  "home_town": "Tucson, Arizona",
  "home_country": "United States of America",
  "current_town": "Shanghai",
  "current_country: "China"
}
```

```
{
  "name": "毛尚立",
  "home_town": "亚利桑那州图森",
  "home_country": "美国",
  "current_town": "上海",
  "current_country: "中国"
}
```


file.json

Quick and Nimble

```
import Quick
import Nimble
@testable import TestingExample

class PersonJSONParsingSpec: QuickSpec {
  override func spec() {
     //test code here
  }
}
```

```
describe("Decoding")
```

```
describe("Decoding")
{
  context("valid person data")
  {
```

```
describe("Decoding")
{
  context("valid person data")
  {
    var testPerson: Person!
```

```
{
  context("valid person data")
  {
    var testPerson: Person!

    beforeSuite
    {
      do {
        testPerson = try parse(data:
        personData()) as Person
       }
      catch { print("Failed: \(error)") }
    }
```

```
it("results in a non-nil Person data object")
{
  expect(testPerson).toNot(beNil())
}
```

```
it("results in a non-nil Person data object")
{
  expect(testPerson).toNot(beNil())
}

it("has a non-empty name")
{
  expect(testPerson?.name).toNot(beEmpty())
}
```

```
    expect(testPerson).toNot(beNil())
}

it("has a non-empty name")
{
    expect(testPerson?.name).toNot(beEmpty())
}

it("has a non-empty hometown")
{
    expect(testPerson?.homeTown).toNot(beEmpty())
}
```

Expected Data Values

```
it("should be me")
{
   expect(testPerson?.name) == "Saul Mora"
}
```

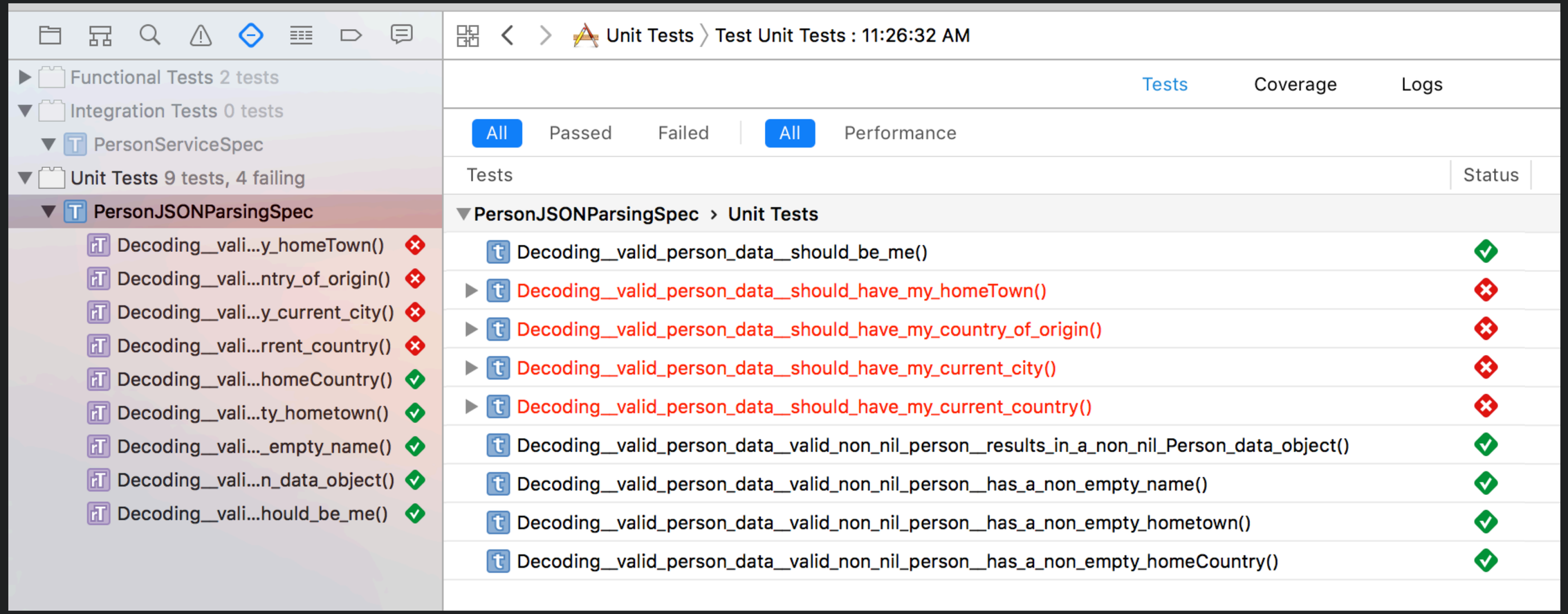
```
it("should be me")
{
   expect(testPerson?.name) == "Saul Mora"
}
it("should have my homeTown")
{
  expect(testPerson?.homeTown) ==
                          "Tucson, Arizona"
}
```

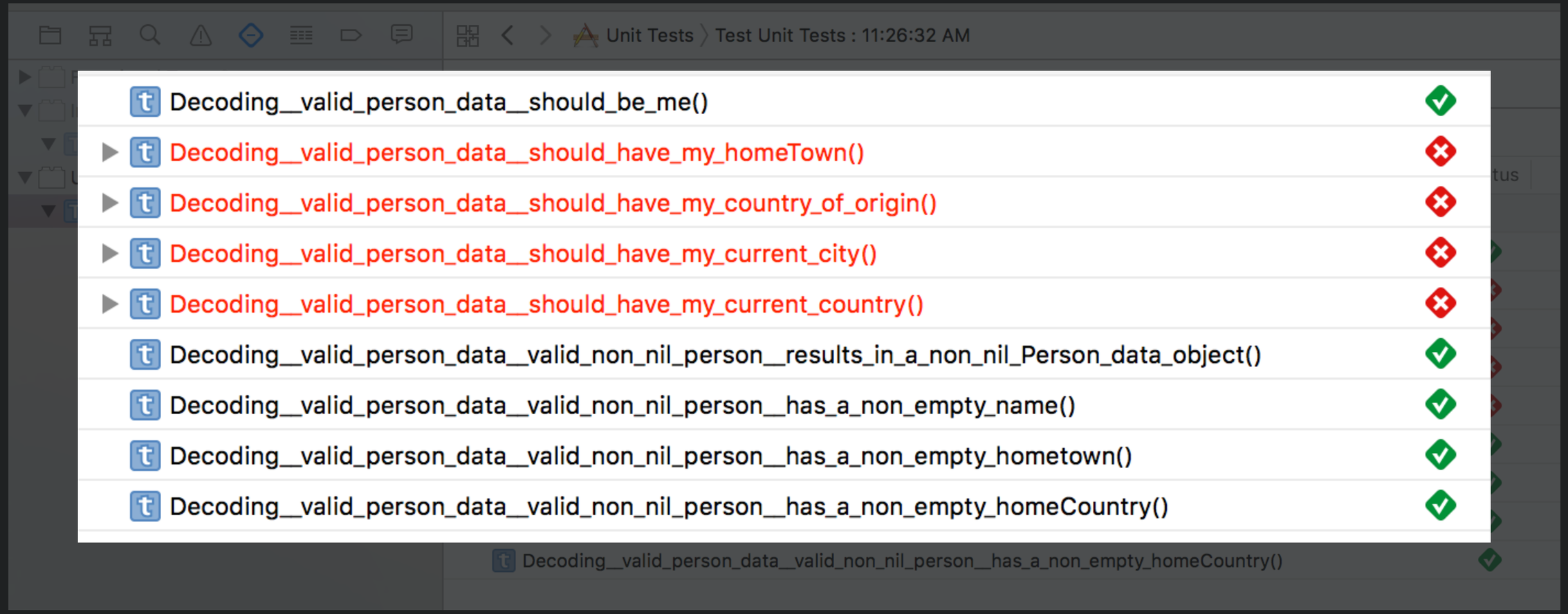
```
sharedExamples("valid non-nil person")
{ (context: @escaping SharedExampleContext) in
```

```
sharedExamples("valid non-nil person")
{ (context: @escaping SharedExampleContext) in
  var testPerson: Person?

  beforeEach
  {
    testPerson = context()["person"] as? Person
  }
```

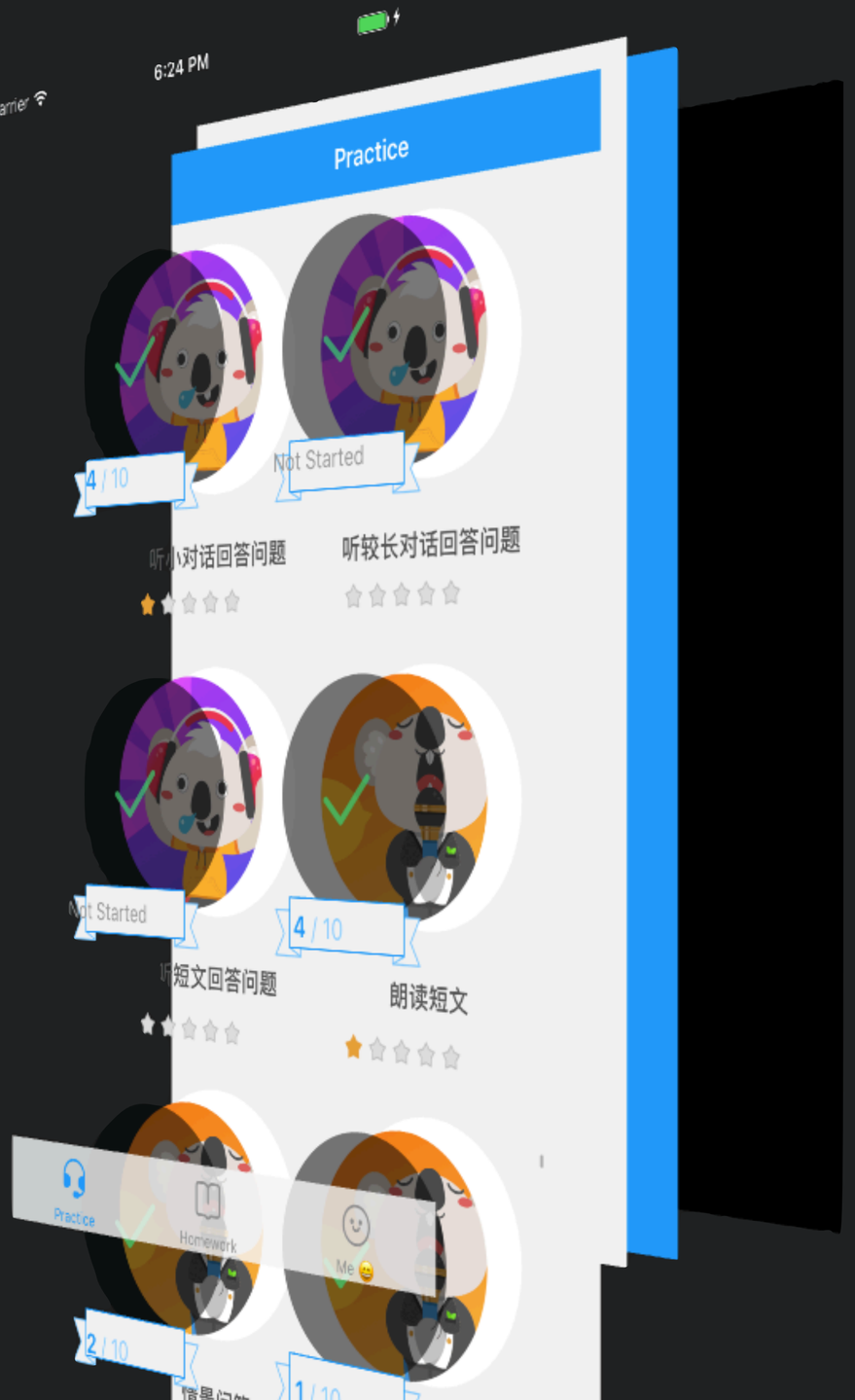
```
itBehavesLike("valid non-nil person")
{ ["person": testPerson] }
```


Functional Tests 2 tests
Integration Tests 0 tests
PersonServiceSpec
Unit Tests 9 tests, 4 failing
PersonJSONParsingSpec
Decoding__vali...y_homeTown()
Decoding__vali...ntry_of_origin()
Decoding__vali...y_current_city()
Decoding__vali...rrent_country()
Decoding__vali...homeCountry()
Decoding__vali...ty_hometown()
Decoding__vali..._empty_name()
Decoding__vali...n_data_object()
Decoding__vali...hould_be_me()
Unit Tests 〉 Test Unit Tests : 11:26:32 AM
Tests
Coverage
Logs
All
Passed
Failed
All
Performance
Tests
Status
PersonJSONParsingSpec › Unit Tests
Decoding__valid_person_data__should_be_me()
Decoding__valid_person_data__should_have_my_homeTown()
Decoding__valid_person_data__should_have_my_country_of_origin()
Decoding__valid_person_data__should_have_my_current_city()
Decoding__valid_person_data__should_have_my_current_country()
Decoding__valid_person_data__valid_non_nil_person__results_in_a_non_nil_Person_data_object()
Decoding__valid_person_data__valid_non_nil_person__has_a_non_empty_name()
Decoding__valid_person_data__valid_non_nil_person__has_a_non_empty_hometown()
Decoding__valid_person_data__valid_non_nil_person__has_a_non_empty_homeCountry()


Unit Tests
Test Unit Tests : 11:26:32 AM
Decoding__valid_person_data__should_be_me()
Decoding__valid_person_data__should_have_my_homeTown()
Decoding__valid_person_data__should_have_my_country_of_origin()
Decoding__valid_person_data__should_have_my_current_city()
Decoding__valid_person_data__should_have_my_current_country()
Decoding__valid_person_data__valid_non_nil_person__results_in_a_non_nil_Person_data_object()
Decoding__valid_person_data__valid_non_nil_person__has_a_non_empty_name()
Decoding__valid_person_data__valid_non_nil_person__has_a_non_empty_hometown()
Decoding__valid_person_data__valid_non_nil_person__has_a_non_empty_homeCountry()
Decoding__valid_person_data__valid_non_nil_person__has_a_non_empty_homeCountry()

Aspects

方面



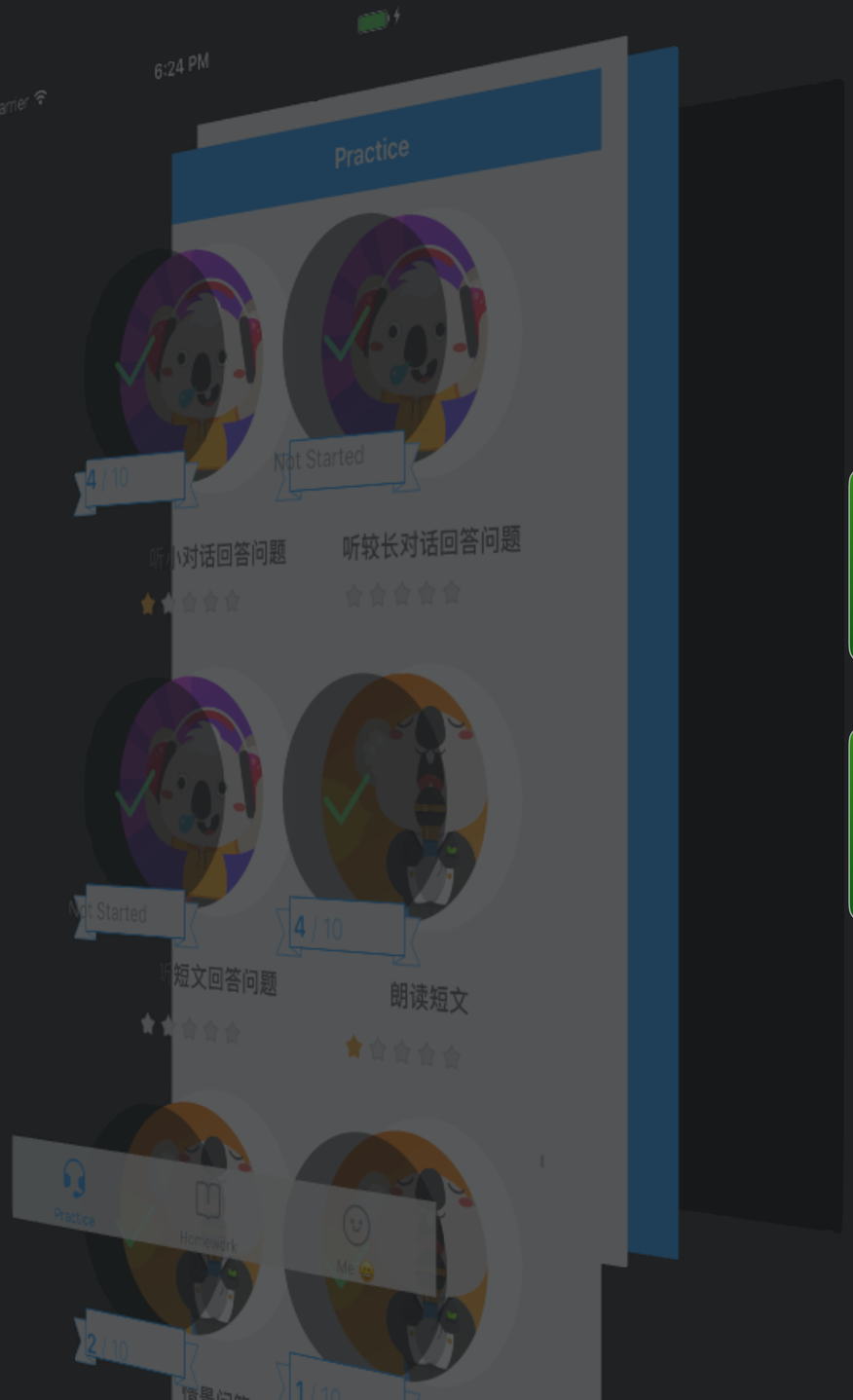
Network

HTTP

URLSession

JSON Parsing

Reachability



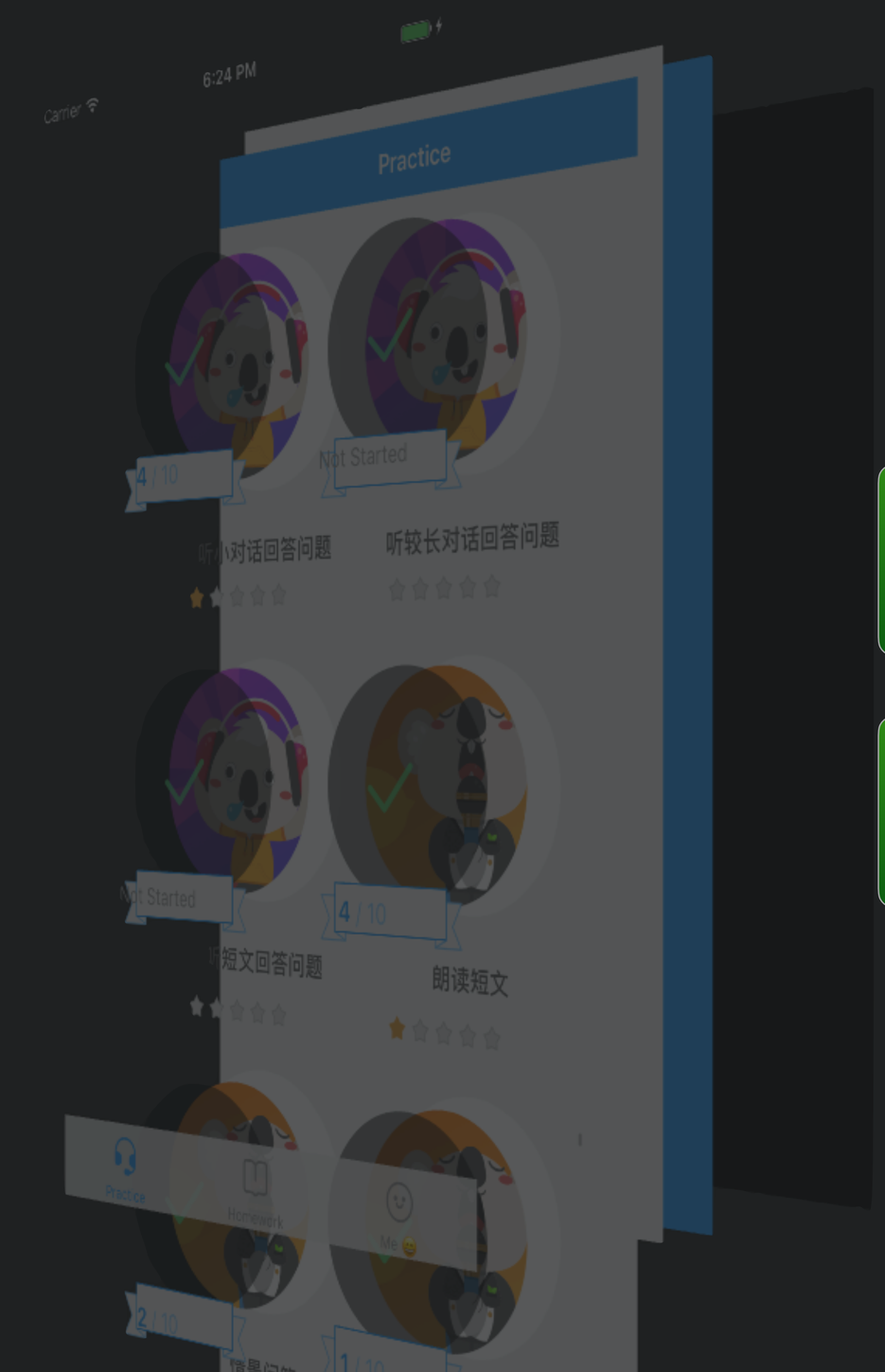
Network

HTTP

URLSession

Parsing

Reachability



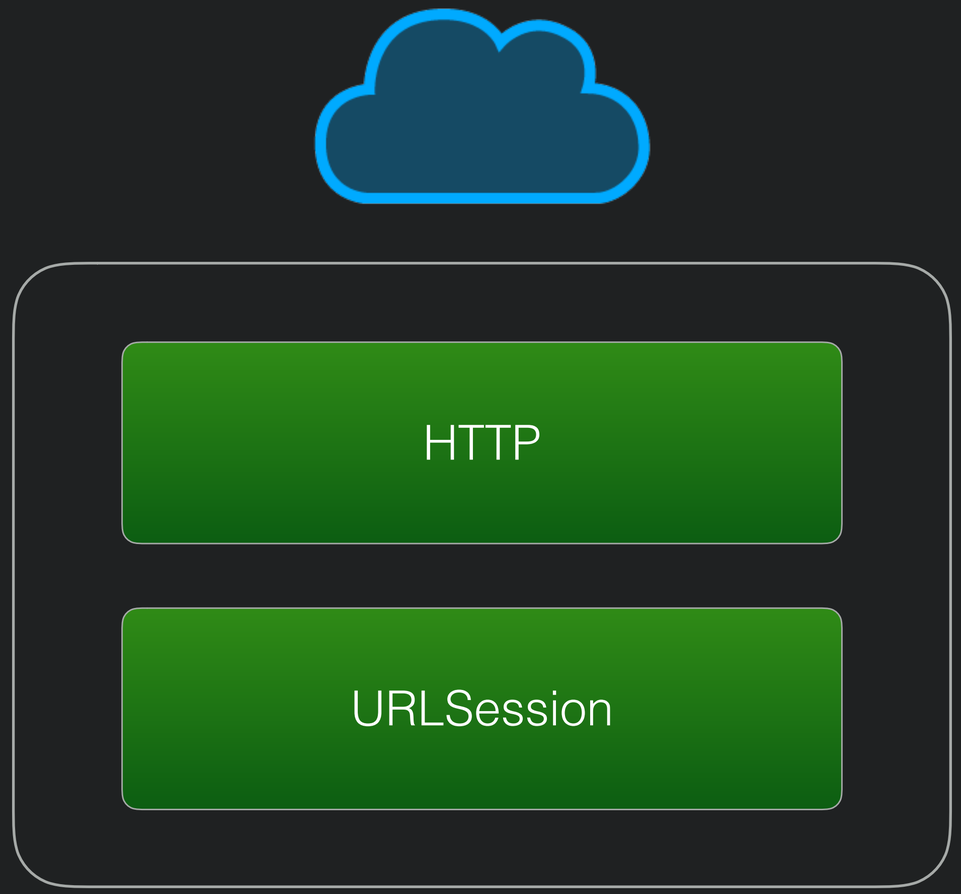
Network

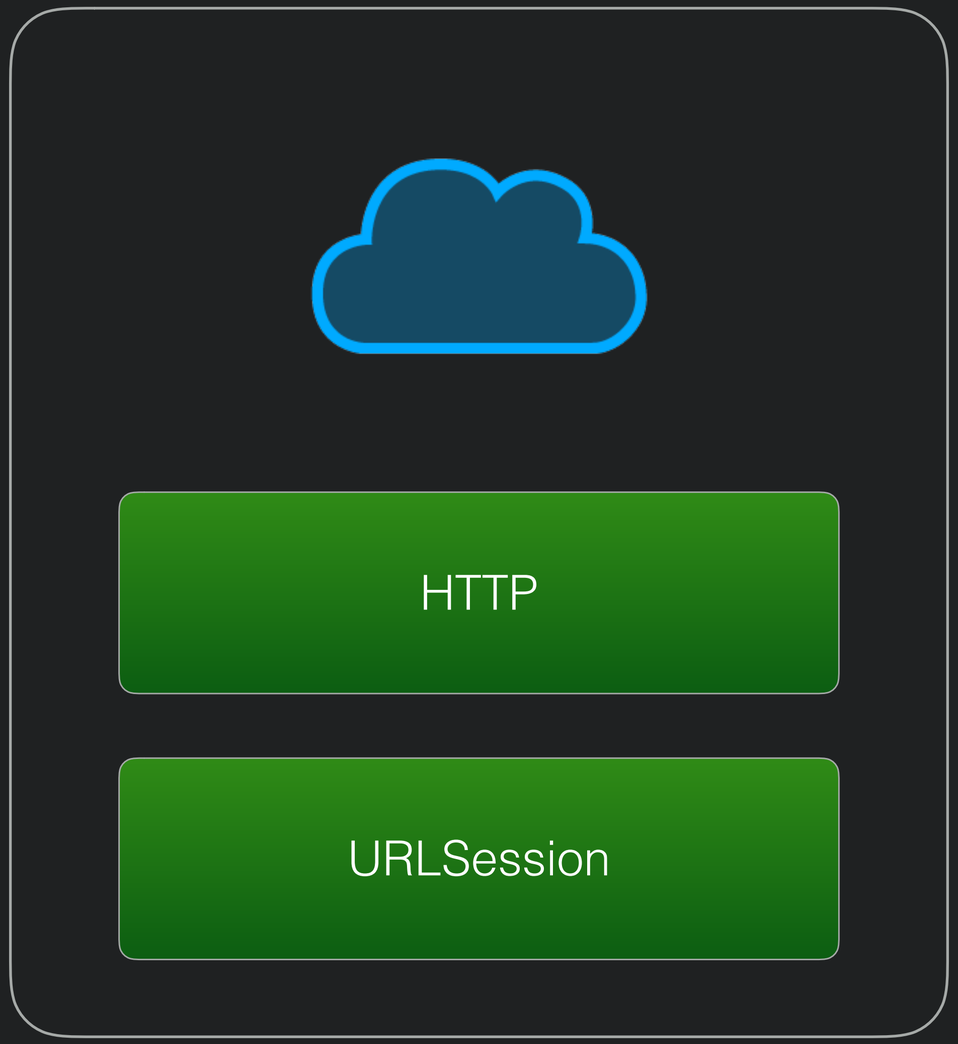
HTTP

URLSession

Parsing

Reachability


HTTP
URLSession


HTTP
URLSession

Functional Tests

Embassy

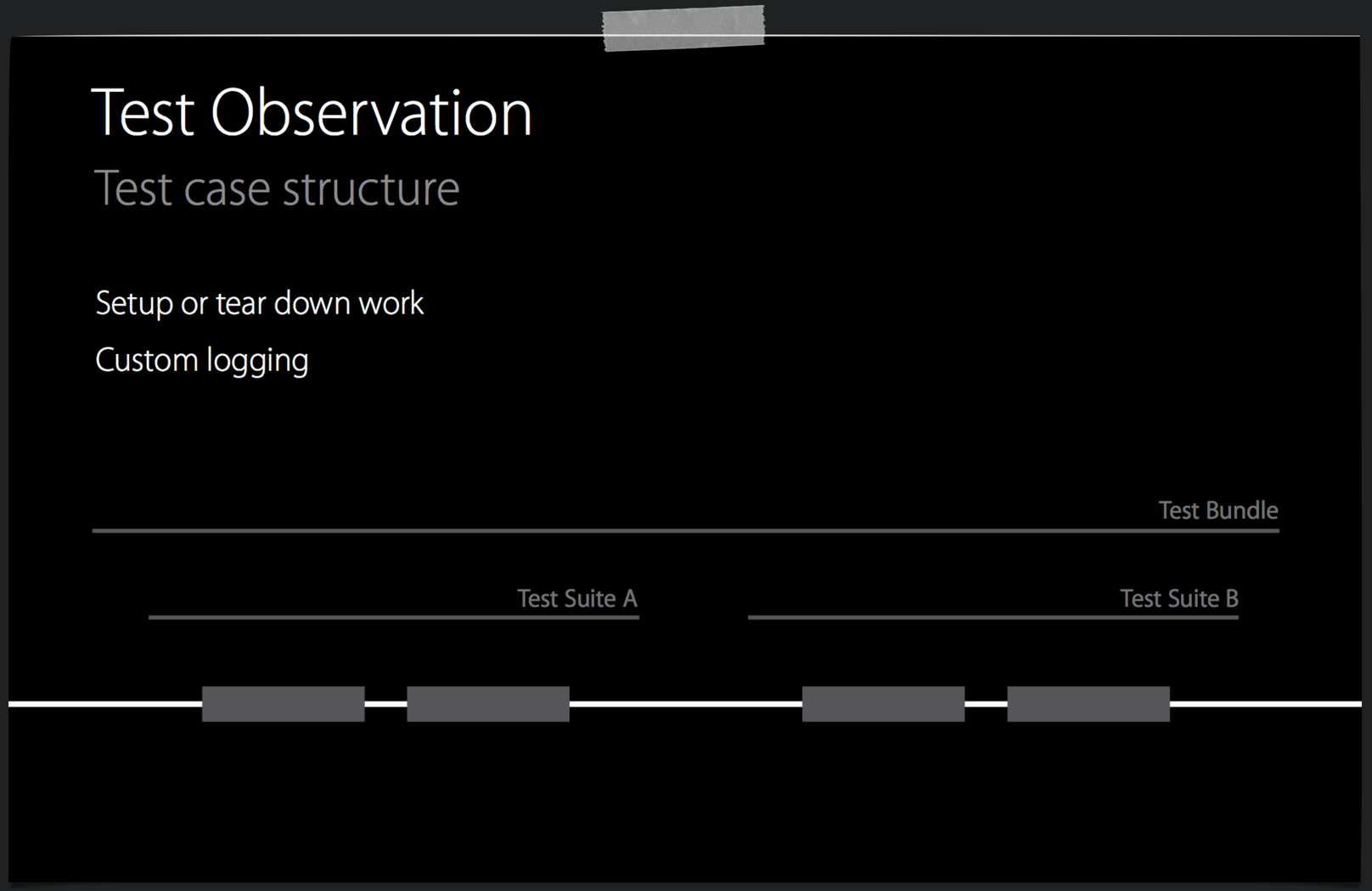
Embassy + XCUnit

Test Observation

Test case structure

Setup or tear down work

Custom logging



https://developer.apple.com/videos/play/wwdc2016/409/

```
private(set) var testHTTPHost: TestHTTPHost = TestHTTPHost()

class TestSuiteObserver: NSObject, XCTestObservation
{
  override init() {
    super.init()
    XCTestObservationCenter.shared().addTestObserver(self)
  }

  func testBundleWillStart(_ testBundle: Bundle) {
    Nimble.AsyncDefaults.Timeout = 5
    testHTTPHost.start()
  }

  func testCaseDidFinish(_ testCase: XCTestCase) {
    testHTTPHost.httpHandler = defaultHandler
  }
}
```

```
describe("Person data service")
{
  context("returning valid data")
  {
```

```
describe("Person data service")
{
  context("returning valid data")
  {
    beforeSuite
    {
      testHTTPHost.httpHandler =
          servePersonData(env:response:body:)
    }
```

```
var testPerson: Person!
let semaphore = DispatchSemaphore(value: 0)

beforeSuite
{
  request(apiHost: .localhost) { person in
    testPerson = person
    semaphore.signal()
  }
  _ = semaphore.wait(timeout: .now() + 10)
}
```

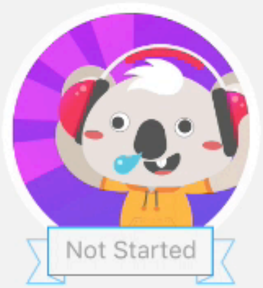
```
itBehavesLike("valid non-nil person")
                { ["person": testPerson] }
```

UI Testing

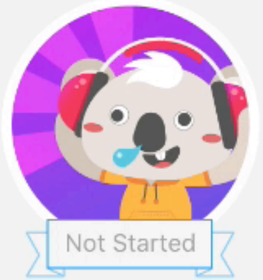
Carrier 6:24 PM

Practice

听小对话回答问题



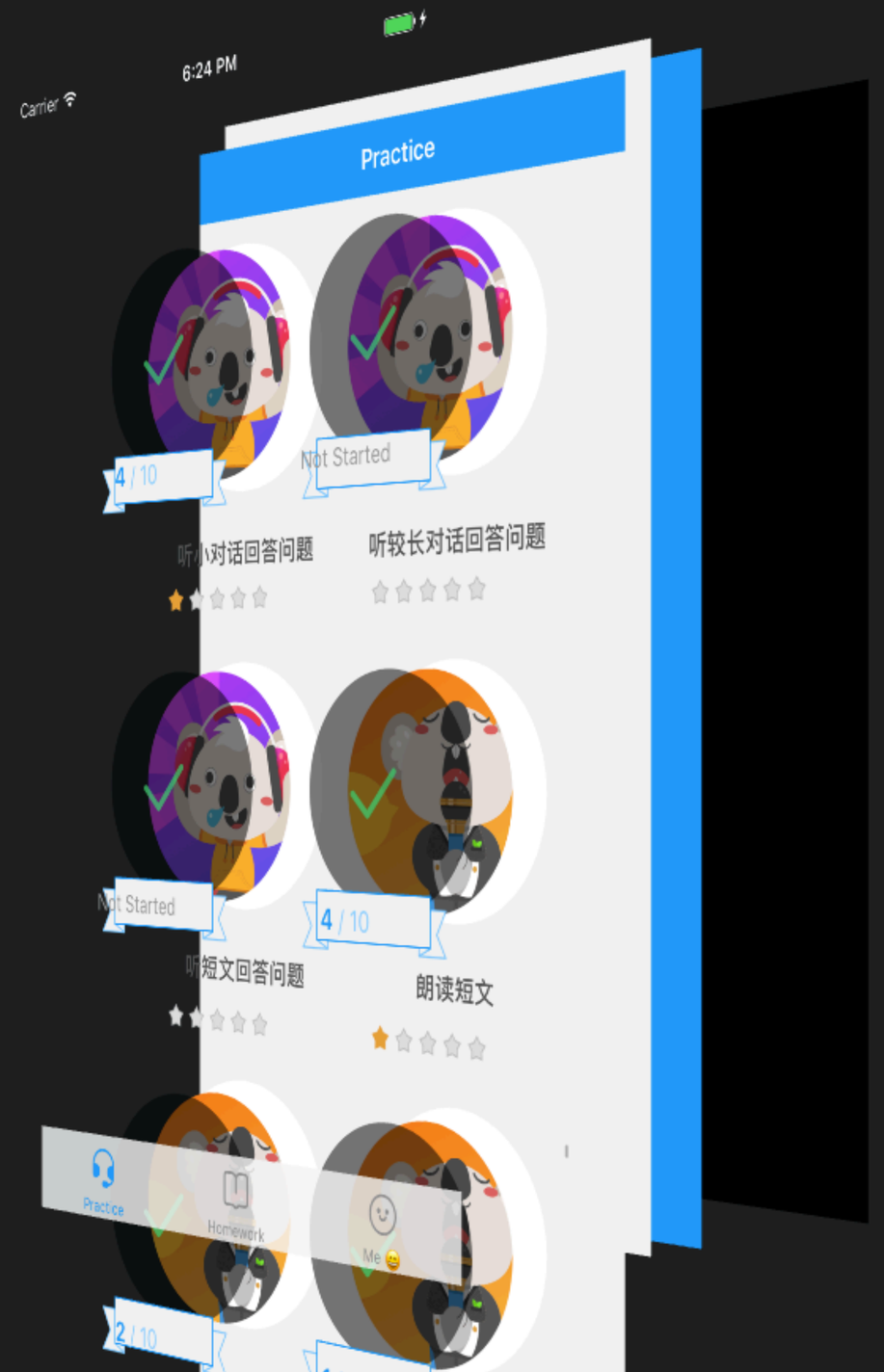
听较长对话回答问题

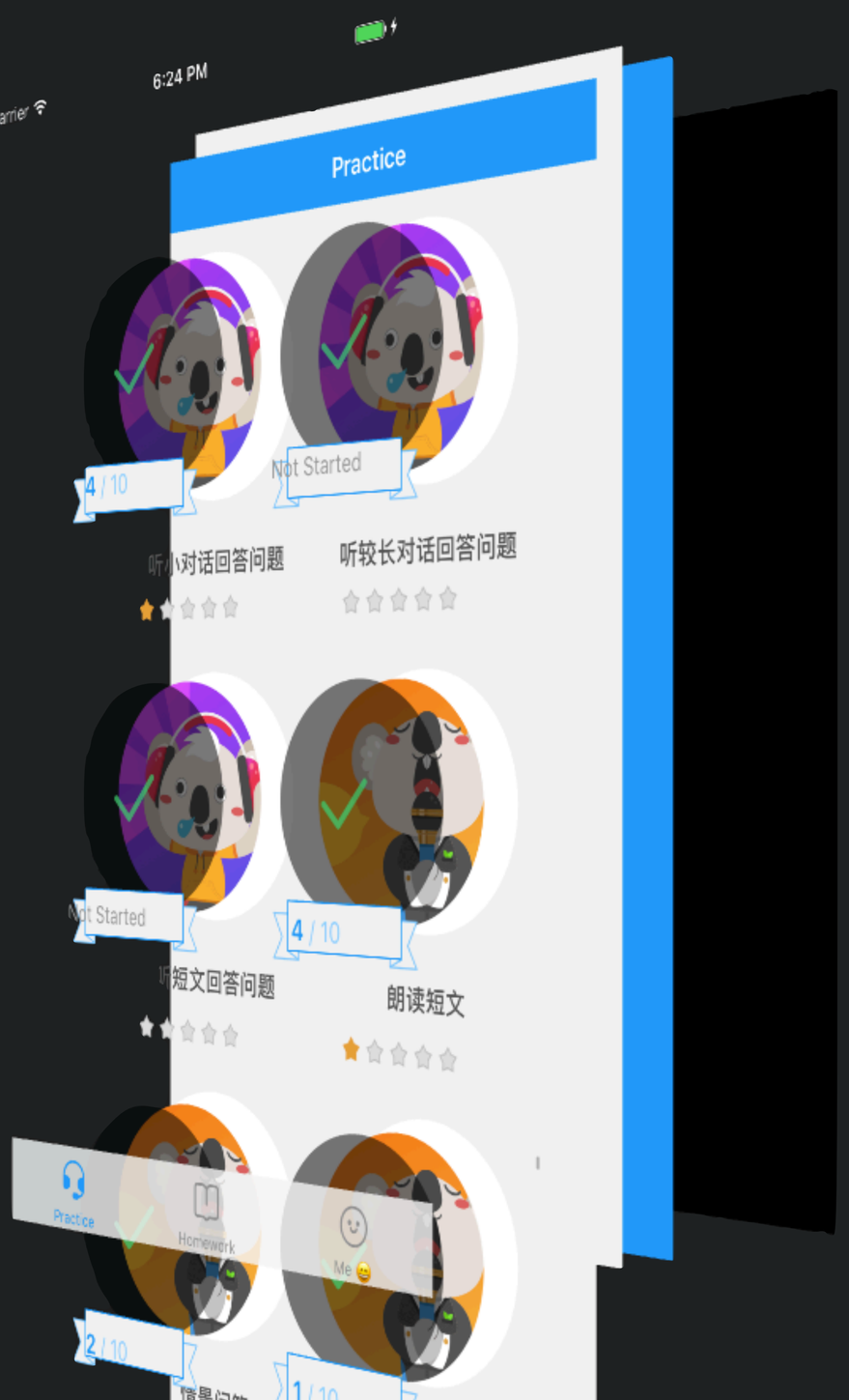


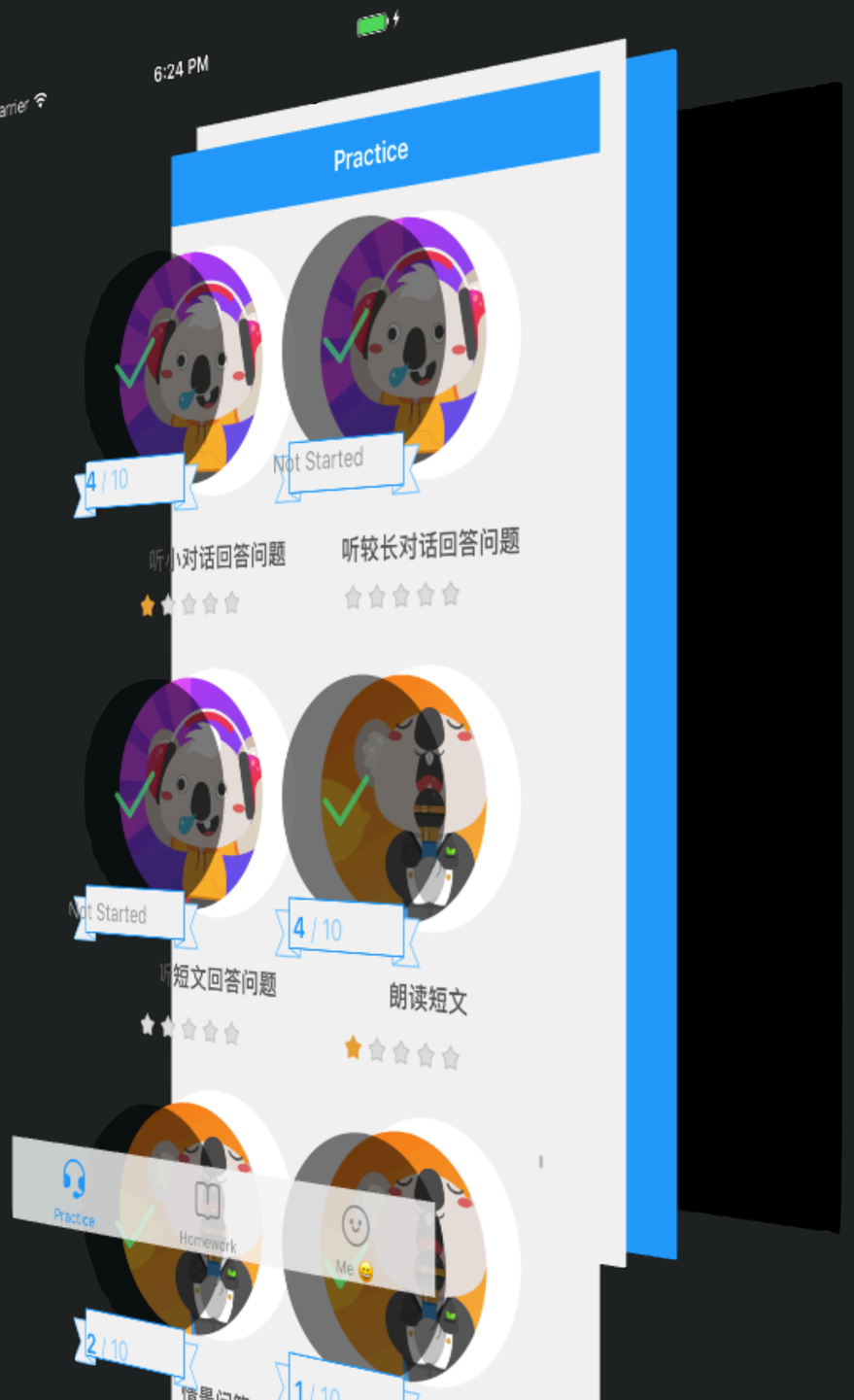
听短文回答问题

朗读短文

Practice Homework Me


Carrier
6:24 PM
Practice
4 / 10
Not Started
听较长对话回答问题
Not Started
4 / 10
朗读短文
Practice
Homework
Me
2 / 10


Carrier
6:24 PM
Practice
4 / 10
Not Started
听较长对话回答问题
Not Started
4 / 10
朗读短文
Practice
Homework
Me
2 / 10
1 / 10



UIView

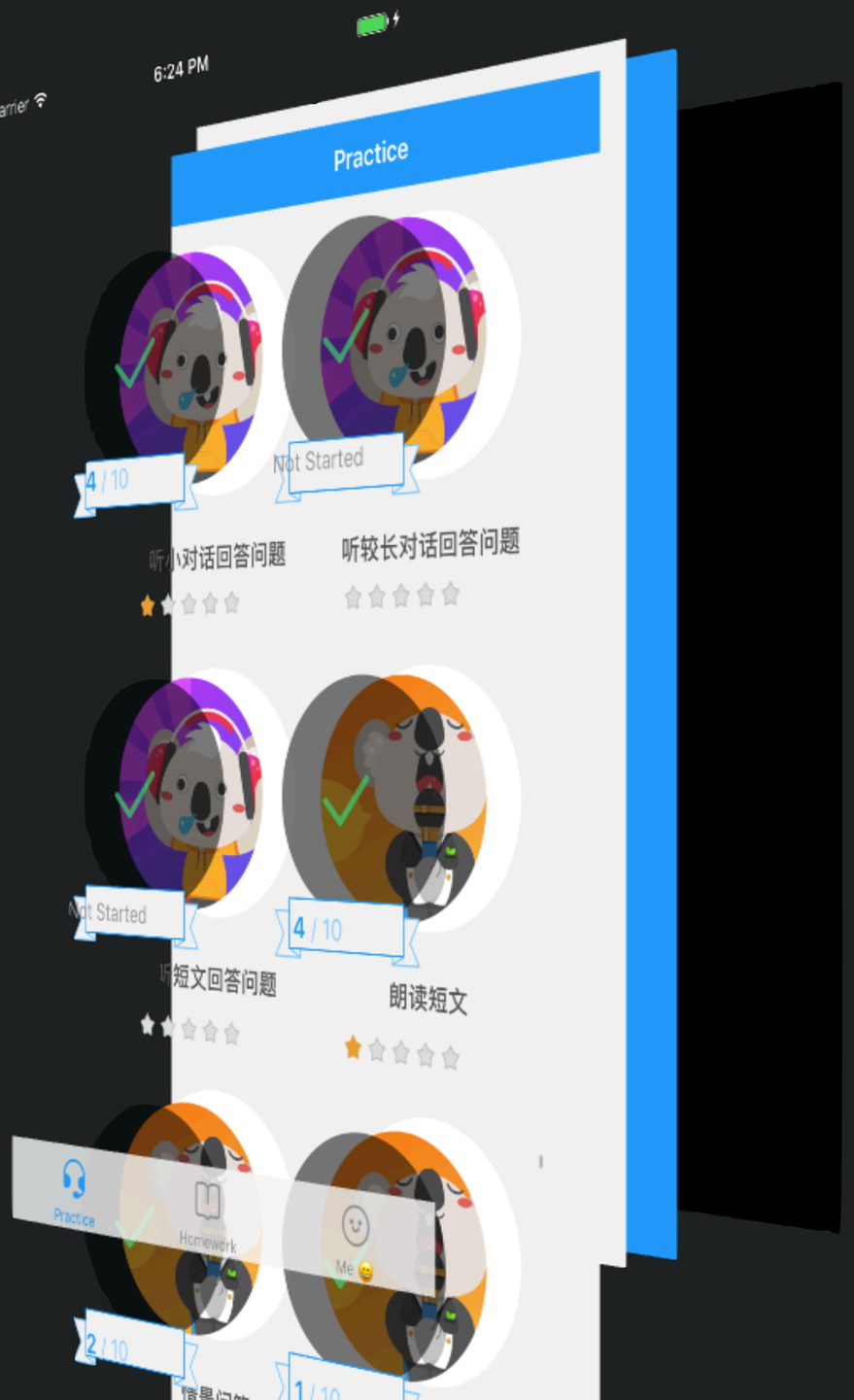
UIViewController

UICollectionViewCell

UICollectionViewDataSource

Disk Cache

Network



UIView

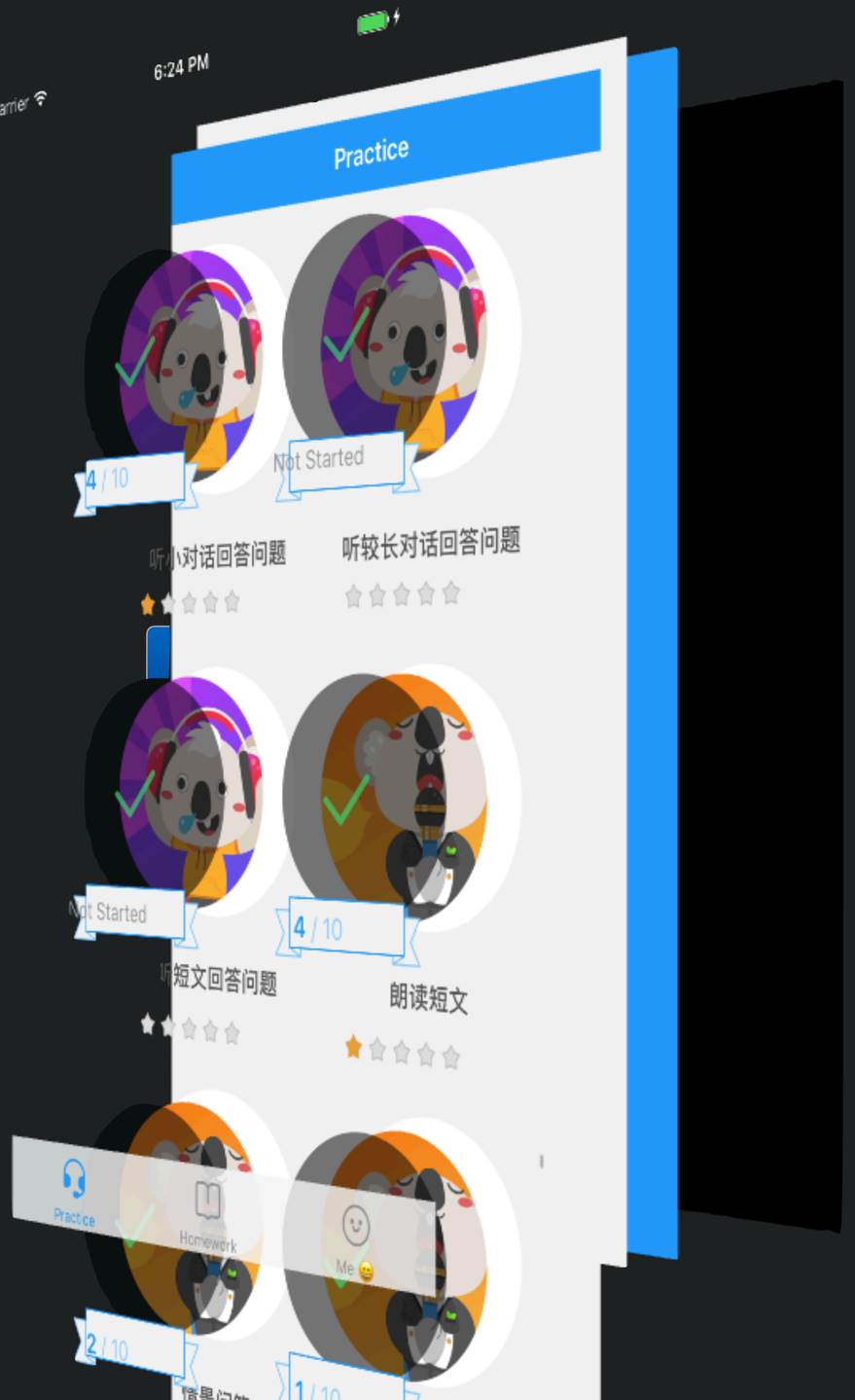
UIViewController

UICollectionViewCell

UICollectionViewDataSource

Disk Cache

Network



AppCore

Disk Cache

Network

Architecture

OCSlim + Fitnesse

What about…

Singletons

"I can't test that shared instance in Swift code!"

```
class SlowAndHeavyHardwareInterface
{
  static let shared: SlowAndHeavyHardwareInterface
         = SlowAndHeavyHardwareInterface()

  private init() {
    ///slow initialization code
  }
}
```

```
class CoreApplicationLogic
{
  let hardwareAPI: SlowAndHeavyHardwareInterface
  init(hardwareAPI: SlowAndHeavyHardwareInterface)
  {
    self.hardwareAPI = hardwareAPI
  }
}
```

```
class CoreApplicationLogic
{
  let hardwareAPI: SlowAndHeavyHardwareInterface
  init(hardwareAPI: SlowAndHeavyHardwareInterface = .shared)
  {
    self.hardwareAPI = hardwareAPI
  }
}
```

Subclass?

真的吗?

```
protocol HardwareInterface
{
   func hardwareFunction()
}
```

```
extension SlowAndHeavyHardwareInterface: HardwareInterface
{
  func hardwareFunction()
  {
    self.slowHardwareFunction()
  }
}
```

```
class CoreApplicationLogic
{
  let hardwareAPI: SlowAndHeavyHardwareInterface
  init(hardwareAPI: SlowAndHeavyHardwareInterface = .shared)
  {
    self.hardwareAPI = hardwareAPI
  }
}
```

```
class CoreApplicationLogic
{
  let hardwareAPI: HardwareInterface
  init(hardwareAPI: HardwareInterface =
                SlowAndHeavyHardwareInterface.shared)
  {
    self.hardwareAPI = hardwareAPI
  }

  func useHardware()
  {
    hardwareAPI.hardwareFunction()
  }
}
```

...so testing sounds good...

这测试看上去挺好的

but it can't really work for me...

但是对于我来说没啥用

Short

Short

短了

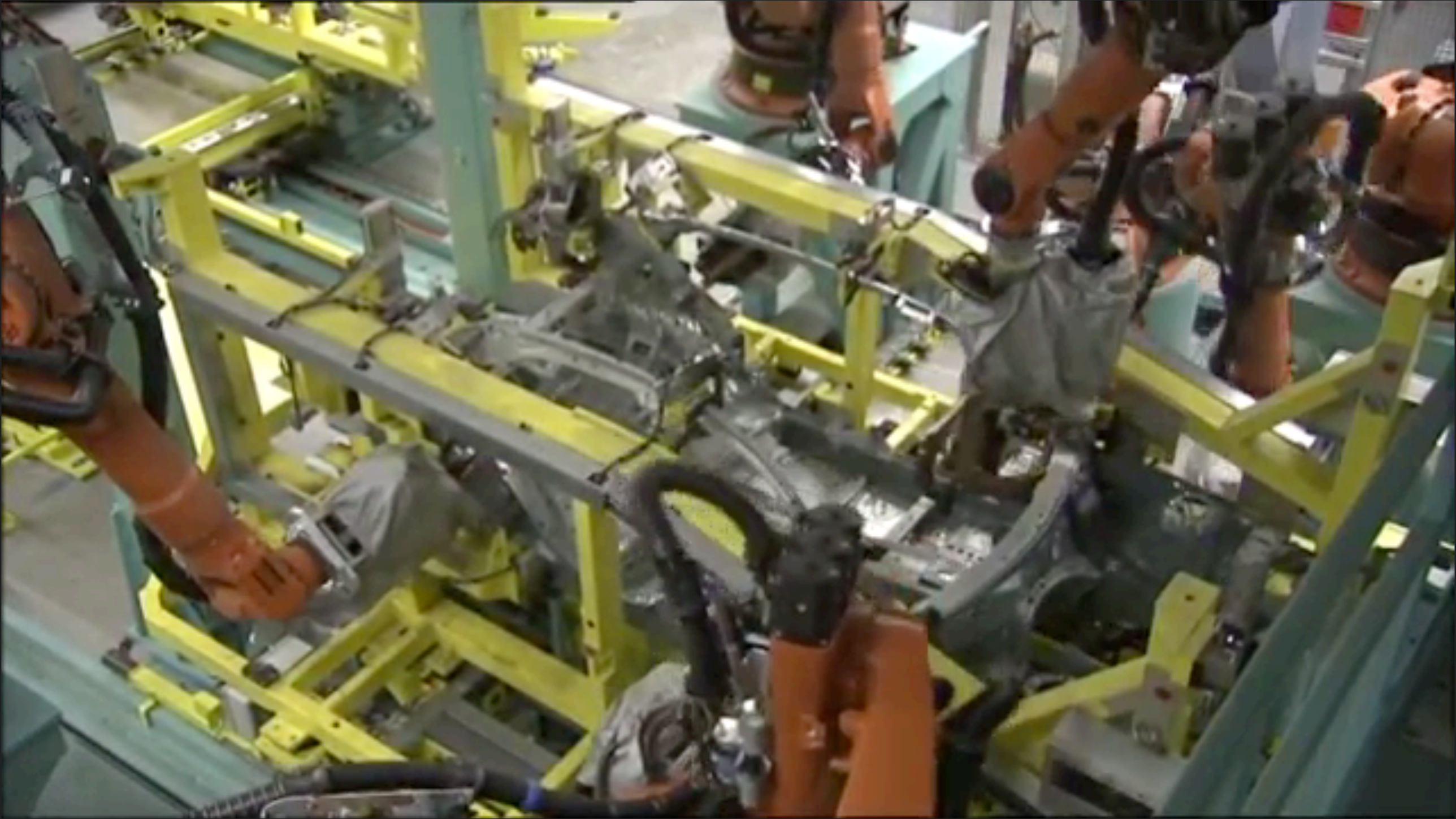
Flexible

灵活

```
it("should be me")
{
  expect(testPerson?.name) == "Saul Mora"
}
it("has a non-empty name")
{
  expect(testPerson?.name).toNot(beEmpty())
}
```

Clear

明白了




Pull On
Volume
Push On-Lock
Color
Push AFT
Tint
Bright
Color
VHF
UHF


Pull On
Volume
Push On-Lock
Color
Push AFT
Tint
Bright
Color
VHF
UHF

你们都是工程师

你们都是工程师

You are all Engineers.

加油大家

谢谢

Building Confidence

Testing your Apps

Saul Mora - saul.mora@liulishuo.com
@casademora

Principles

- Keep it Simple
- Plan for tests to fail
- Failures should be easy and quick to fix
- Test Early (fail fast)
- Test one thing at a time (isolation)
- Test Public APIs only (flexibility)
- Don't Test code that isn't yours*

Practices

Inject Singletons

One assertion per test case

Clearly express intent

Arrange, Act, Assert

Write Tests before Code

Why do tests fail?

Why do tests fail?

- Functionality is broken.
- Test is broken
- System changed
- Test isn't clear what it's testing

Types of Tests

Unit

Integration

Functional

Acceptance

User Interface

Unit

Integration

Functional

Acceptance

User Interface

Unit

Integration

Functional

Acceptance

User Interface

Unit

Integration

Functional

Acceptance

User Interface

Unit

Integration

Functional

Acceptance

User Interface

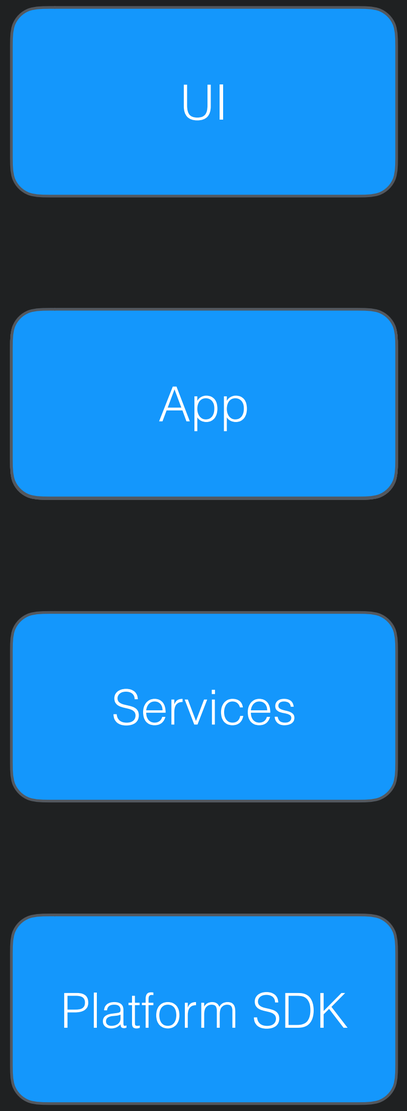
Unit

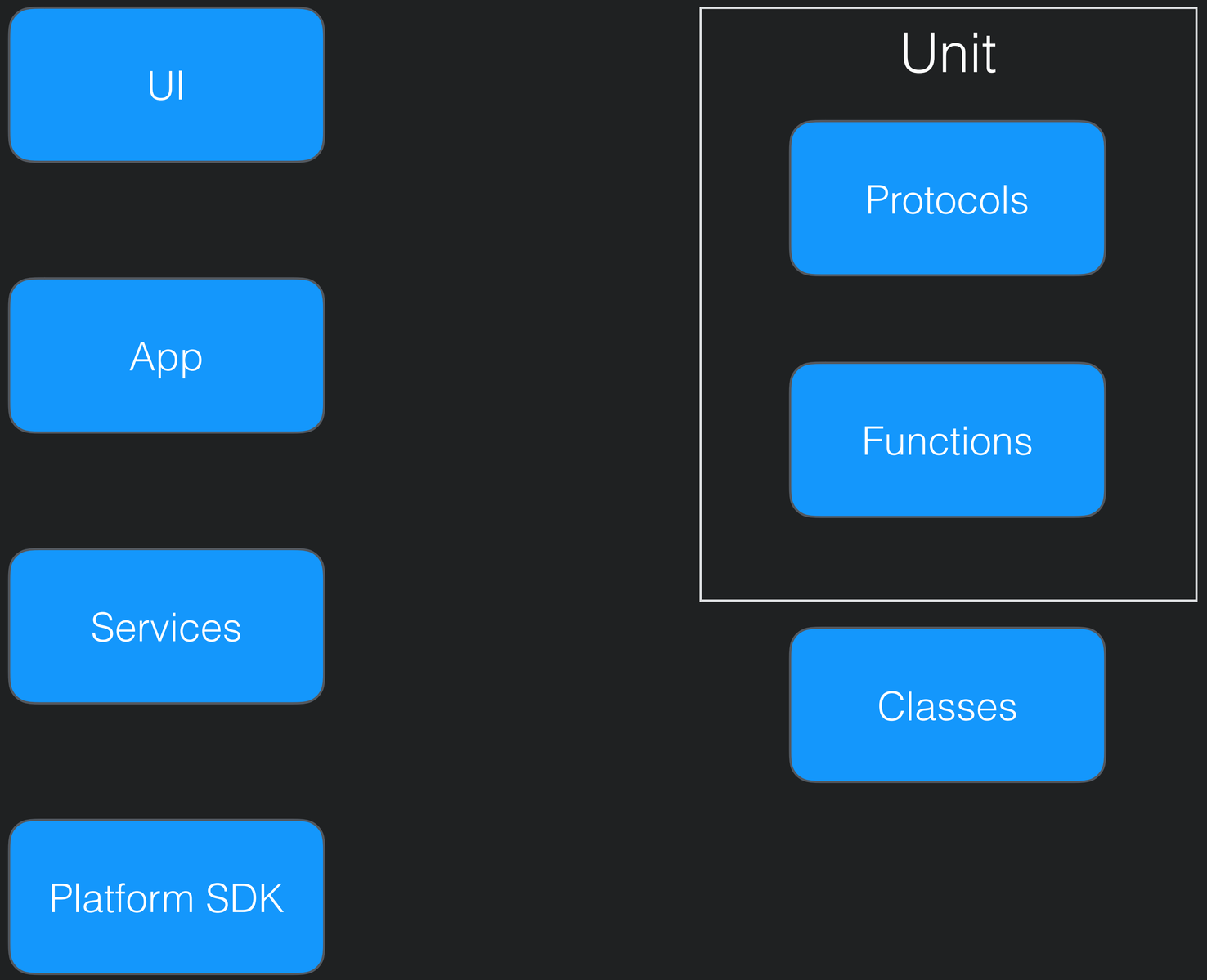
Integration

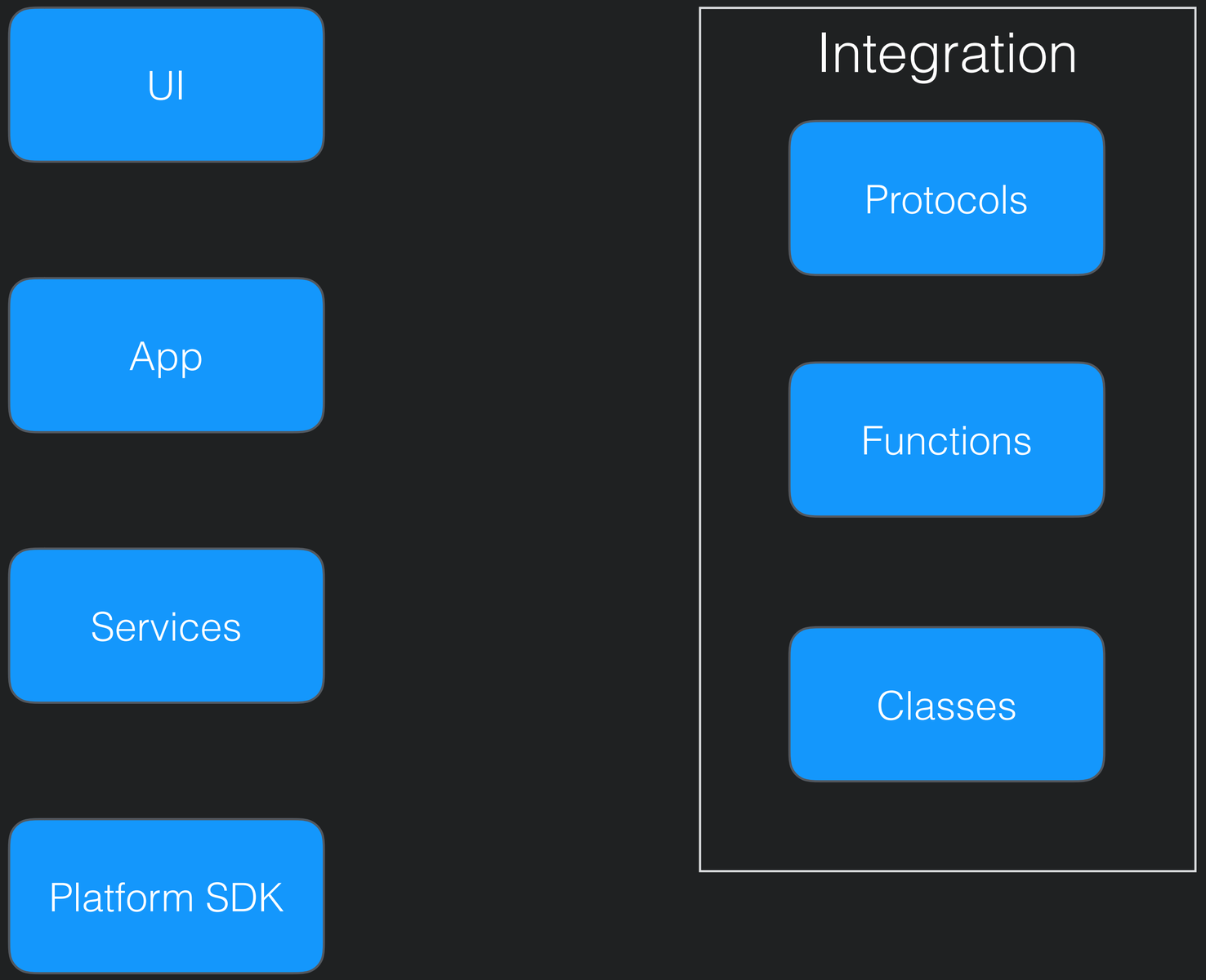
Functional

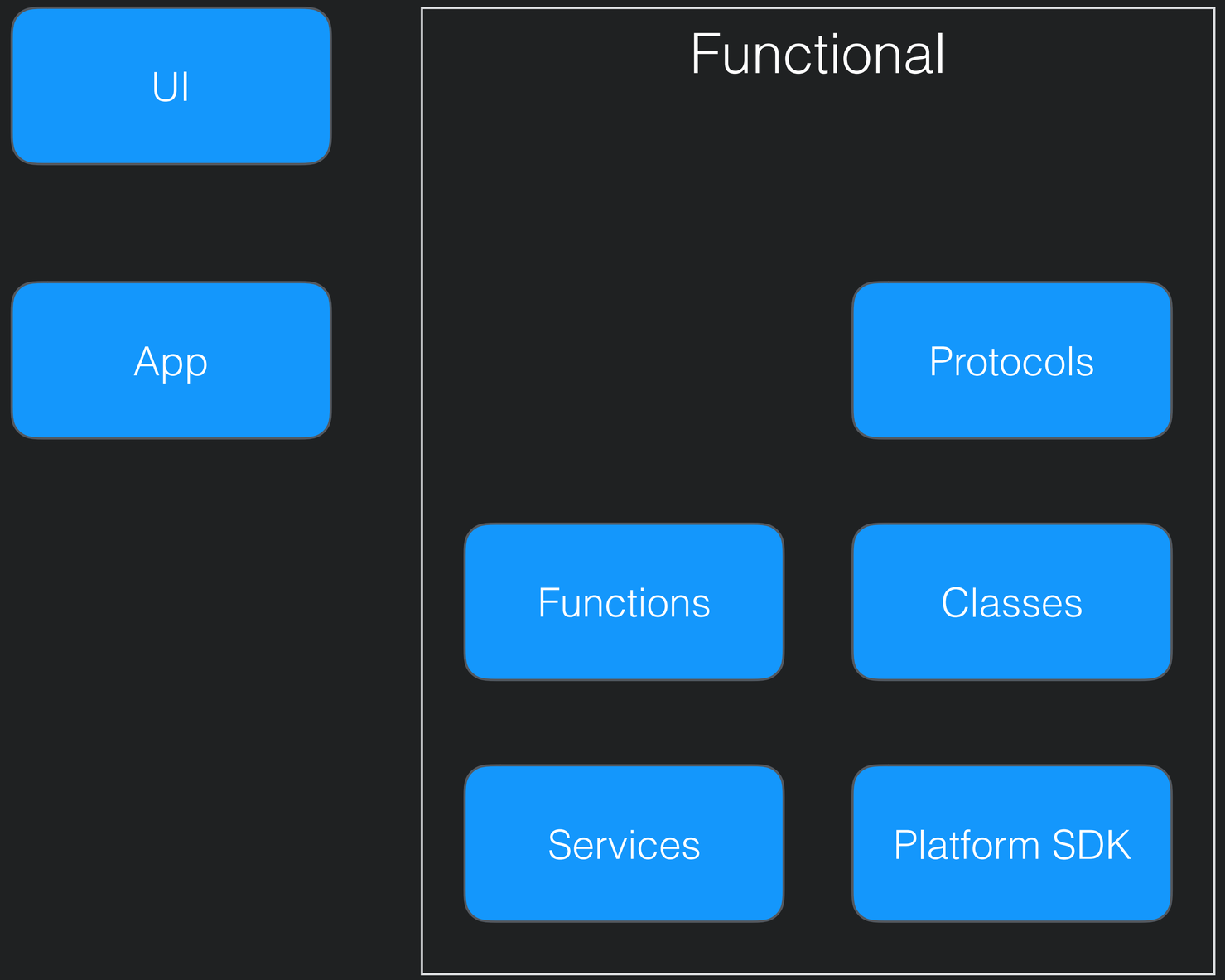
Acceptance

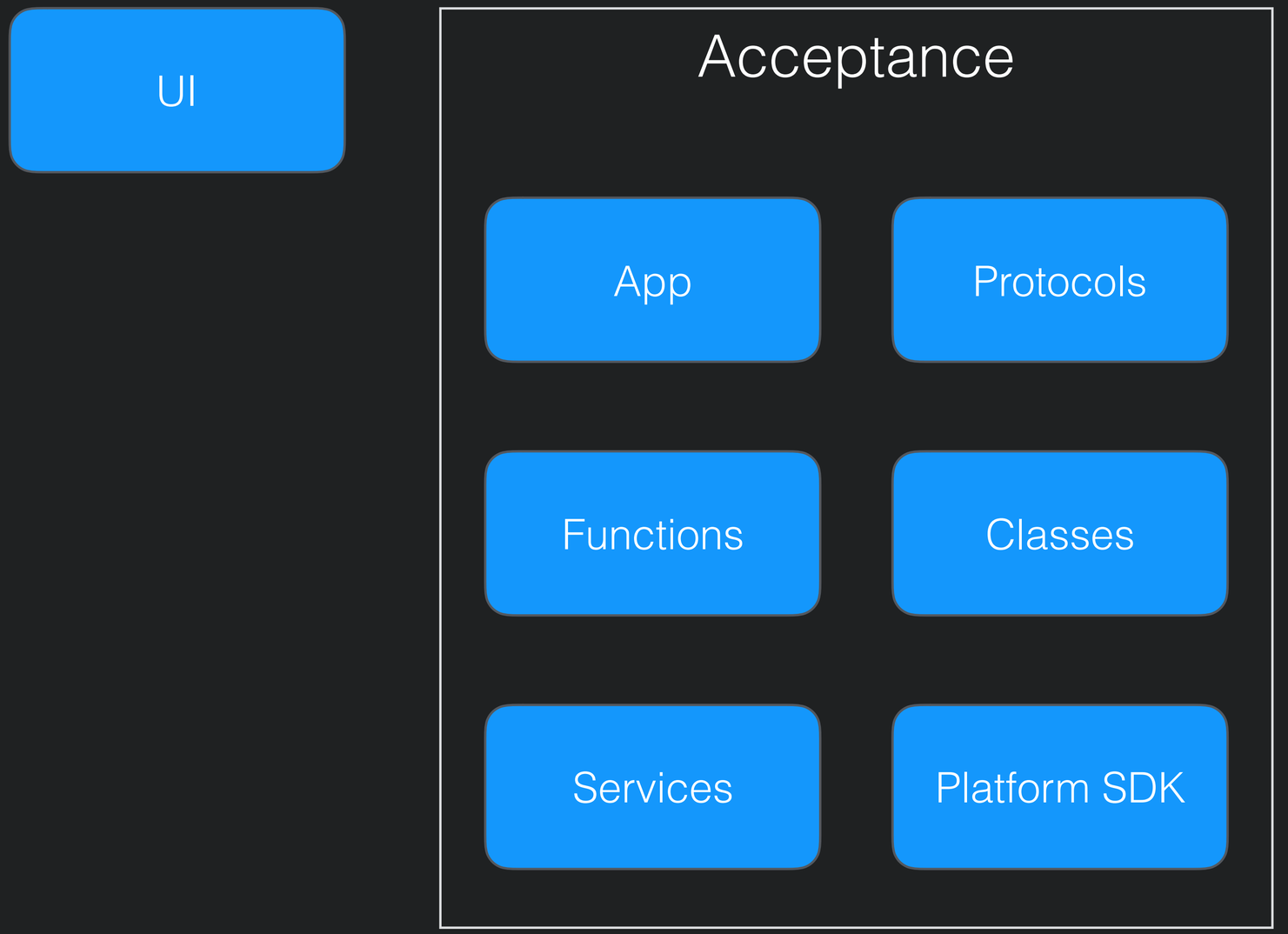
User Interface

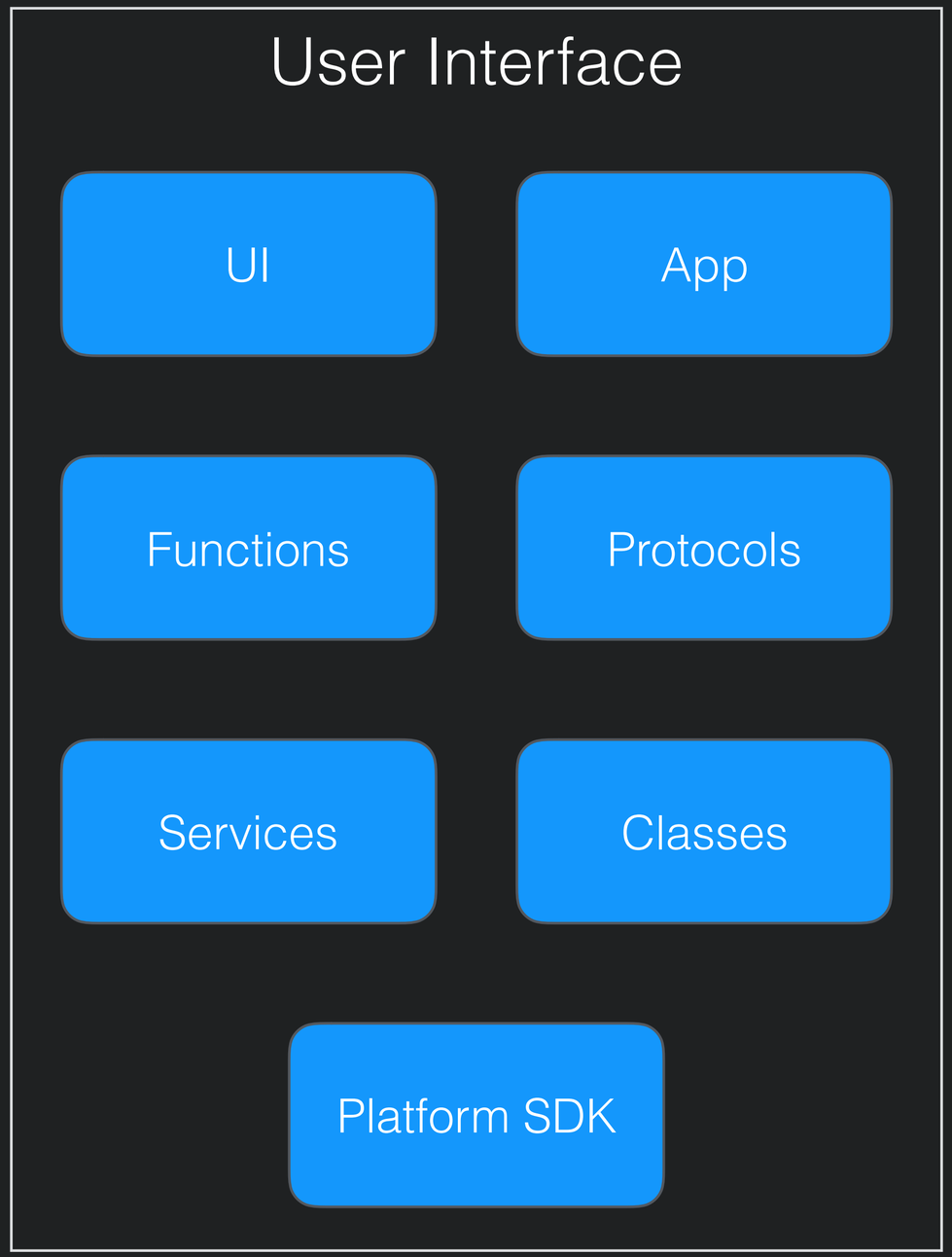

UI
App
Services
Platform SDK


UI
App
Services
Platform SDK
Unit
Protocols
Functions
Classes


UI
App
Services
Platform SDK
Integration
Protocols
Functions
Classes


UI
App
Functional
Protocols
Functions
Classes
Services
Platform SDK


UI
Acceptance
App
Protocols
Functions
Classes
Services
Platform SDK


User Interface
UI
App
Functions
Protocols
Services
Classes
Platform SDK

Unit

Integration

Functional

Acceptance

User Interface

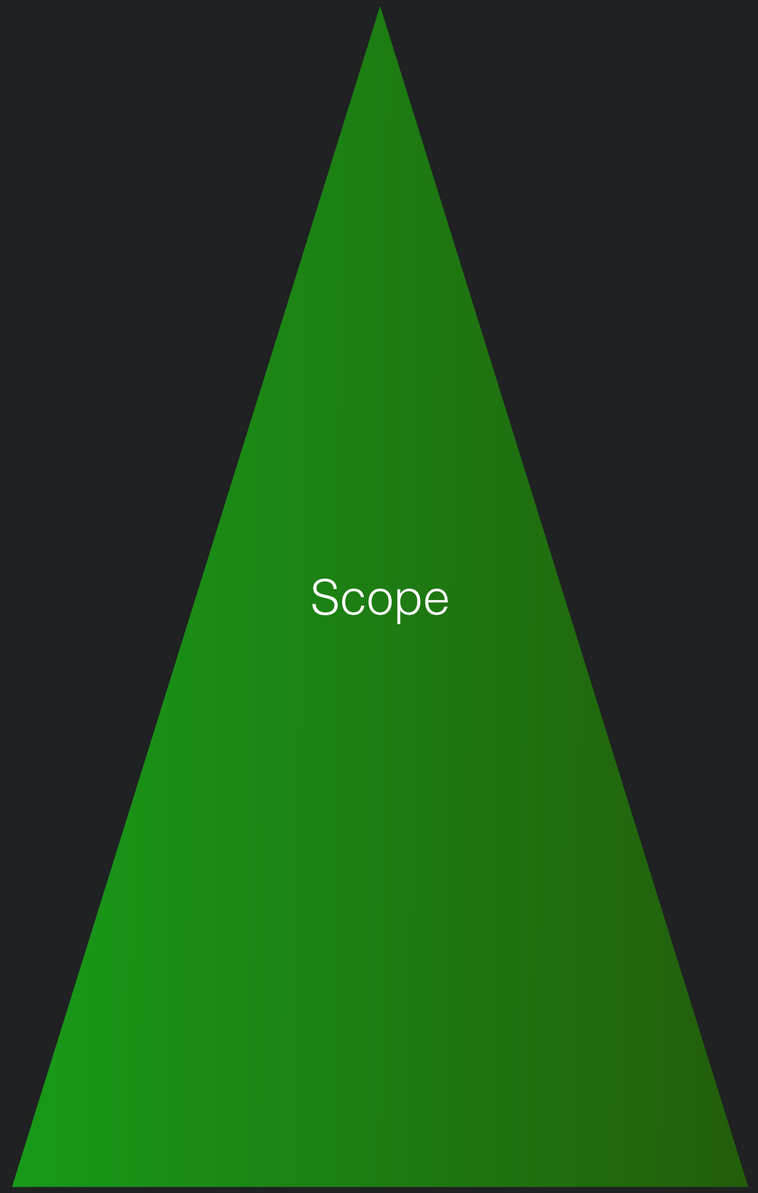
Unit

Integration

Functional

Acceptance

User Interface



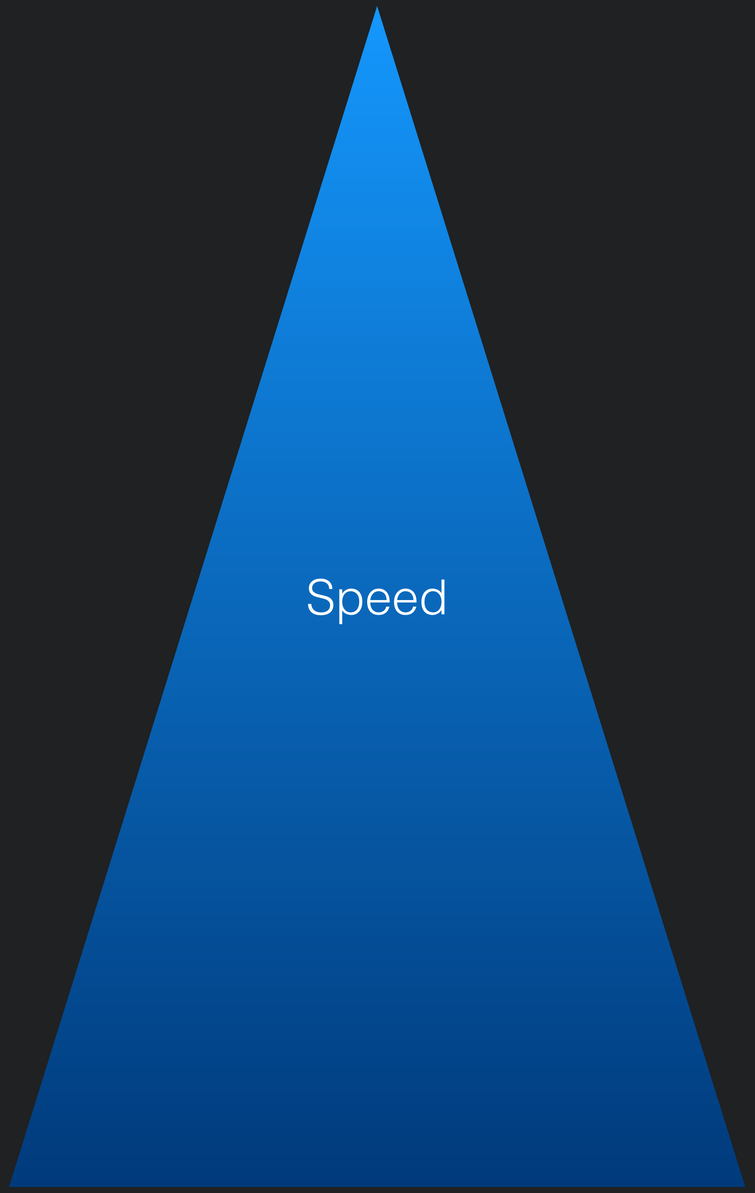
Unit

Integration

Functional

Acceptance

User Interface



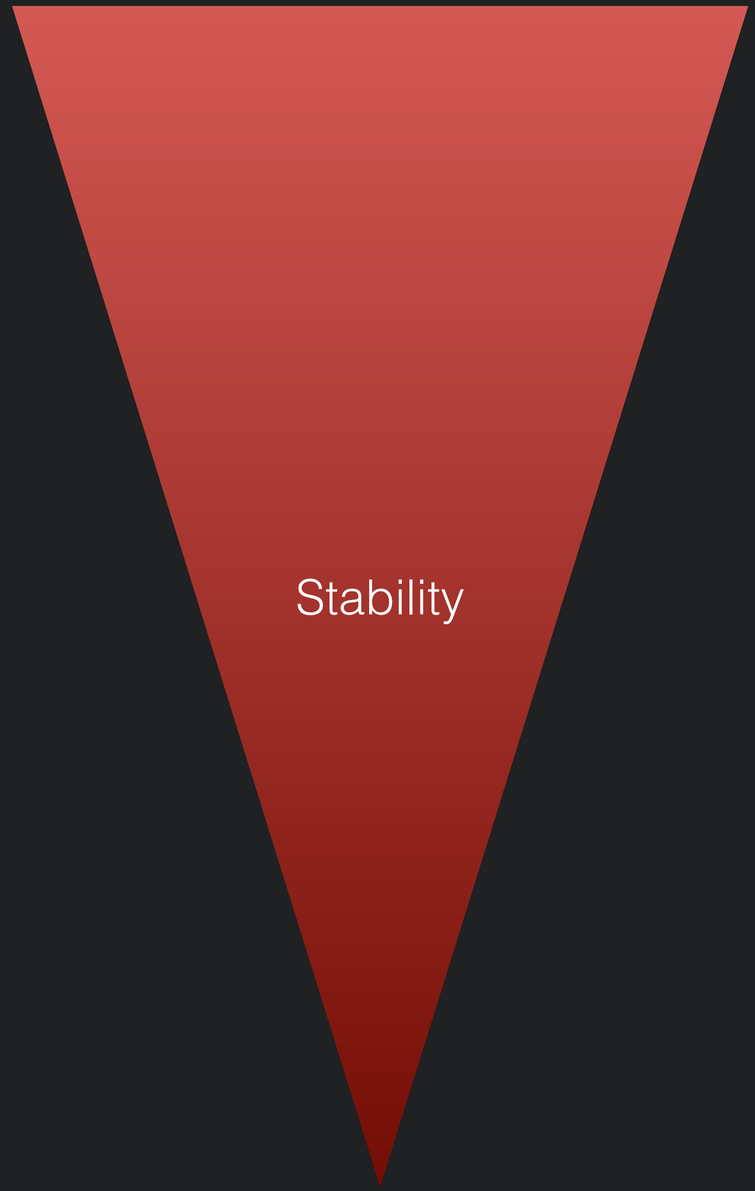
Unit

Integration

Functional

Acceptance

User Interface



Unit	Every Build
Integration	Staging Builds
Functional	Manually, Automatically, Staging
Acceptance	Release Builds
User Interface	Staging, Release