Unit Testing with Quick

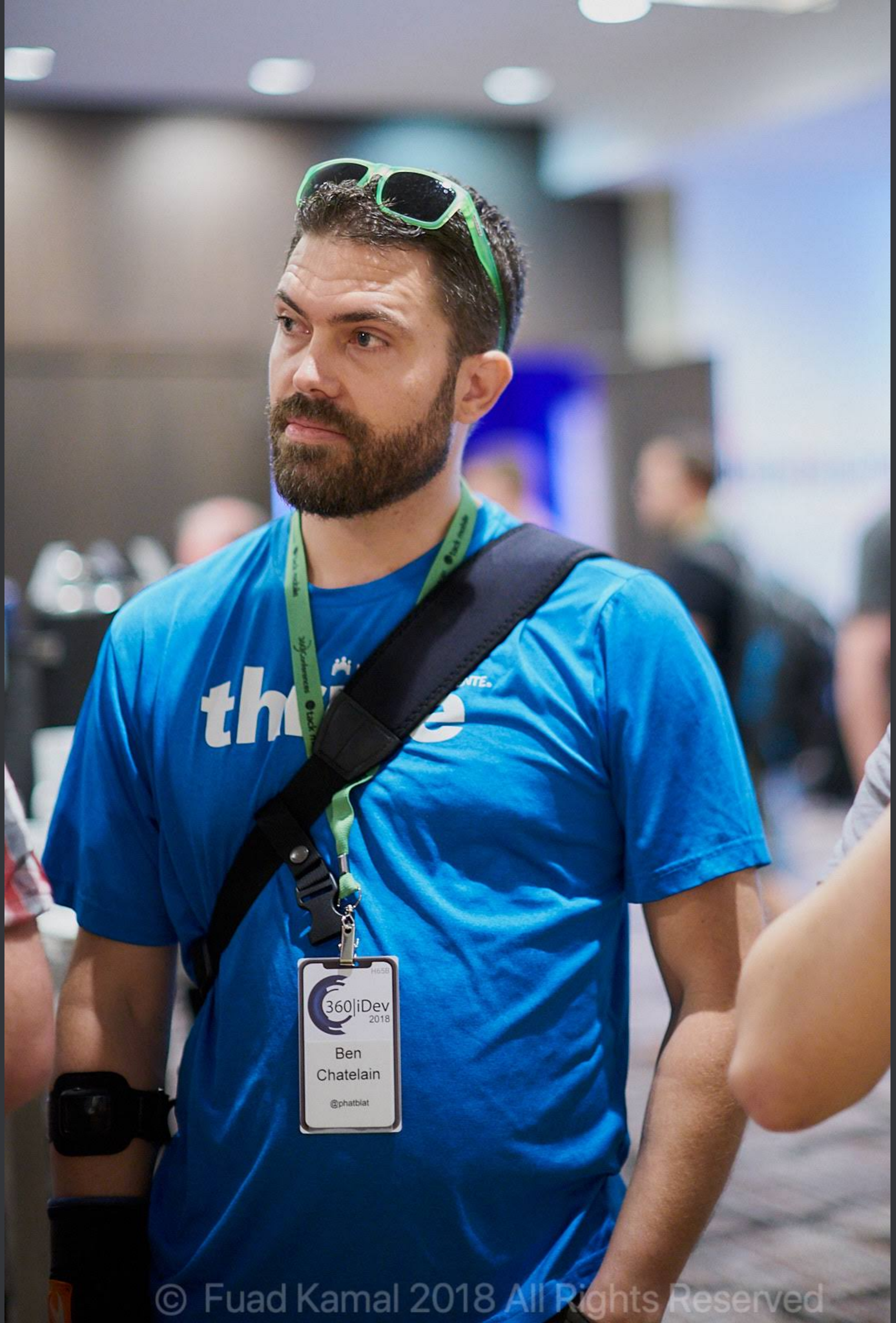
Presented at Cocoaheads Denver
2020-03-10 at Galvanize

Slides & Example Project

phatblat/UnitTestingWithQuick

@phatblat

- Ben Chatelain
- Chief iOS Engineer
- Kaiser Permanente
- Manage suite of ~30 iOS & Android libs
- open source: Quick, mas, Objective-Git
- indie app dev 2008-2009



Quick & Nimble

Open Source

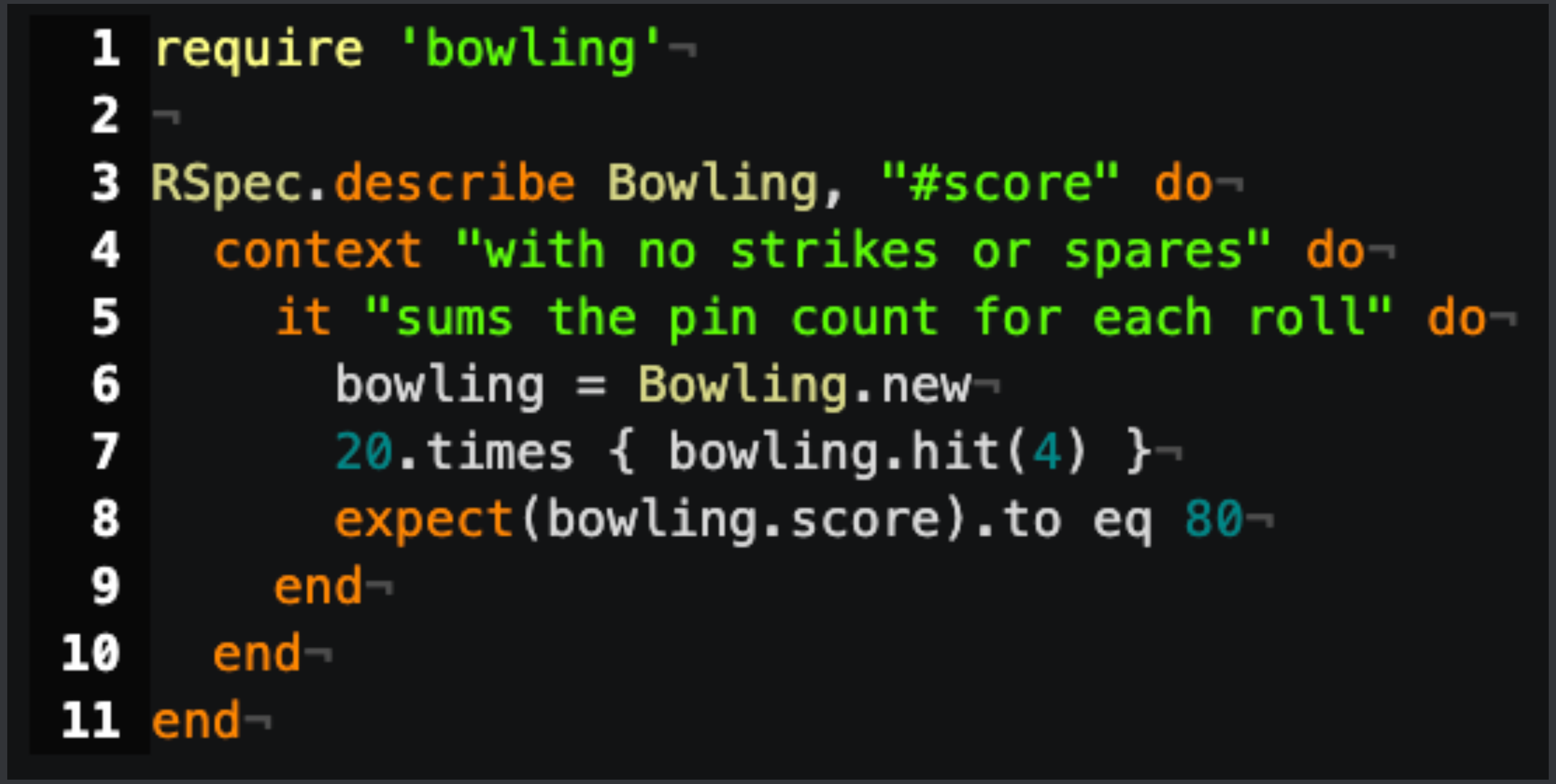
- Quick/Quick
  - Used to define examples.
- Quick/Nimble
  - Matcher framework used to express expectations.

BDD

- *Behavior-Driven Development*
- Don't test code
- Verify behavior
- Semi-formal format for behavior specification
- Similar to user story
- Object-oriented design

RSpec

- Behaviour Driven Development for Ruby.
- "Making TDD Productive and Fun."



```ruby
require 'bowling'

RSpec.describe Bowling, "#score" do
  context "with no strikes or spares" do
    it "sums the pin count for each roll" do
      bowling = Bowling.new
      20.times { bowling.hit(4) }
      expect(bowling.score).to eq 80
    end
  end
end
```

QuickSpec

```swift
class TableOfContentsSpec: QuickSpec {
  override func spec() {
    describe("the 'Documentation' directory") {
      it("has everything you need to get started") {
        let sections = Directory("Documentation").sections
        expect(sections).to(contain("Organized Tests with Quick Examples and Example Groups"))
        expect(sections).to(contain("Installing Quick"))
      }

      context("if it doesn't have what you're looking for") {
        it("needs to be updated") {
          let you = You(awesome: true)
          expect{you.submittedAnIssue}.toEventually(beTruthy())
        }
      }
    }
  }
}
```

describe

```
describe("the thing") { /* closure */ }
```

- Describes the thing being tested.
- Groups examples.
- Serves as a prefix for the actual test name.
- Analogous to **`XCTestCase`**

`it`

```
it("calculates an average score") { /* closure */ }
```

- Describes an example behavior.
- Contains assertions (expectations).
- *One expectation per example.*

context

```
context("when dark mode is enabled") { /* closure */ }
```

- Optional alternate 2nd-Nth level of grouping for examples.
- Arbitrary nesting. 🎎

Disabled Test (Quick)

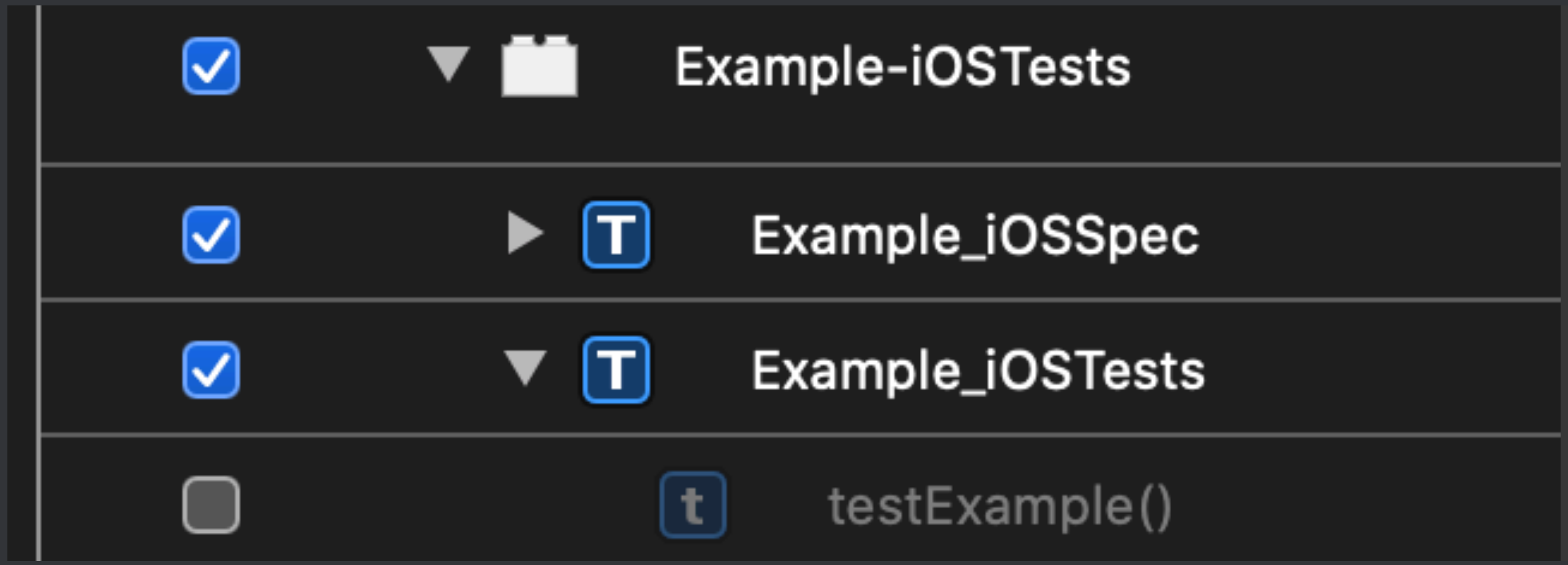
```
xit("this example is disabled") { code.compiles() == yes }
```

- Prefix any example with **x** to disable.
- `xit`

Disabled Test (Xcode)

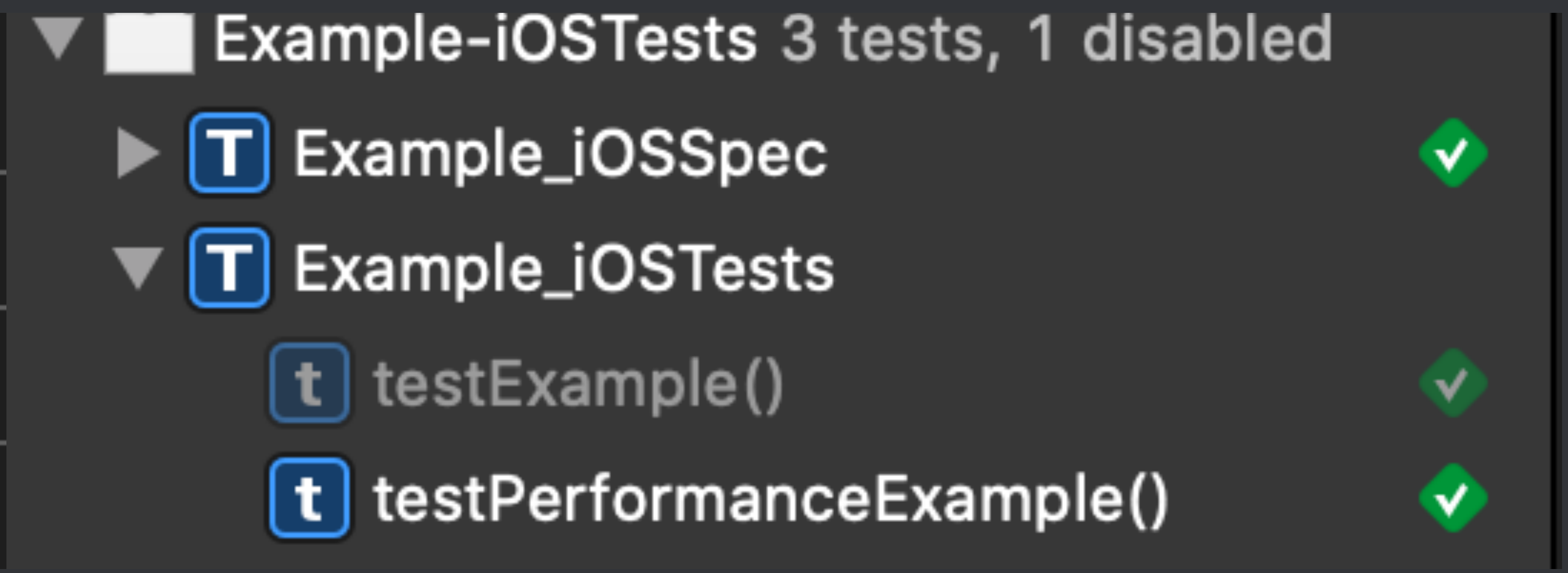
- Must edit scheme to disable tests.
- Shows test method as disabled.

Scheme > Test action



☑ Example-iOSTests
☑ Example_iOSSpec
☑ Example_iOSTests
☐ testExample()

Test Navigator



Example-iOSTests 3 tests, 1 disabled
Example_iOSSpec ✓
Example_iOSTests
testExample() ✓
testPerformanceExample() ✓

✨ Disabled Test Suite

```
xdescribe("the thing") {
xcontext("when dark mode is enabled") {
xit("this example is disabled") { code.compiles() == yes }
```

- Prefix any Quick function(s) with **x** to disable everything under that scope.
- Any combination of disabled examples is skipped.

✨ Focused Test

```
fit("is focused") { expect(example).toRun(true) }
it("will be ignored") { code.compiles() == yes }
```

- Prefix any example with **f** to focus.
- Only the focused example(s) will be run.

✨ Focused Test Suites

```
fdescribe("the thing") {
fcontext("when dark mode is enabled") {
fit("is focused") { expect(example).toRun(true) }
```

- Prefix any Quick function(s) with **f** to focus everything under that scope.
- Only the focused example(s) will be run.
- All focused examples will be run.

Quick Caveats

- Easy to forget disabled/focused tests.
- Clicking on Quick example in Test Navigator doesn't navigate to code.

Xcode UI

Test Name

Type Handling

Nimble

- Matcher framework
- Swift and Objective-C

```
expect(1 + 1).to(equal(2))
expect(1.2).to(beCloseTo(1.1, within: 0.1))
expect(3) > 2
expect("seahorse").to(contain("sea"))
expect(["Atlantic", "Pacific"]).toNot(contain("Mississippi"))
```

Custom Failure Message

```
expect(1 + 1).to(equal(3))
// failed - expected to equal <3>, got <2>

expect(1 + 1).to(equal(3), description: "Make sure libKindergartenMath is loaded")
// failed - Make sure libKindergartenMath is loaded
// expected to equal <3>, got <2>
```

Async Test

```
expect(ocean.isClean).toEventually(beTruthy())
```

Custom Matcher

Closures

Properties

Local Variables

iOS Examples

macOS Examples

This is an awe-inspiring quote.

— Someone Famous

References

- Unit Testing With Quick - Slides & Code Samples
- Quick
  - Quick Docs
- Nimble
- Jon Reid
  - Quality Coding
  - iOS Unit Testing By Example 📕