ZIO Test
The Ultimate Guide

What? Why? And How?

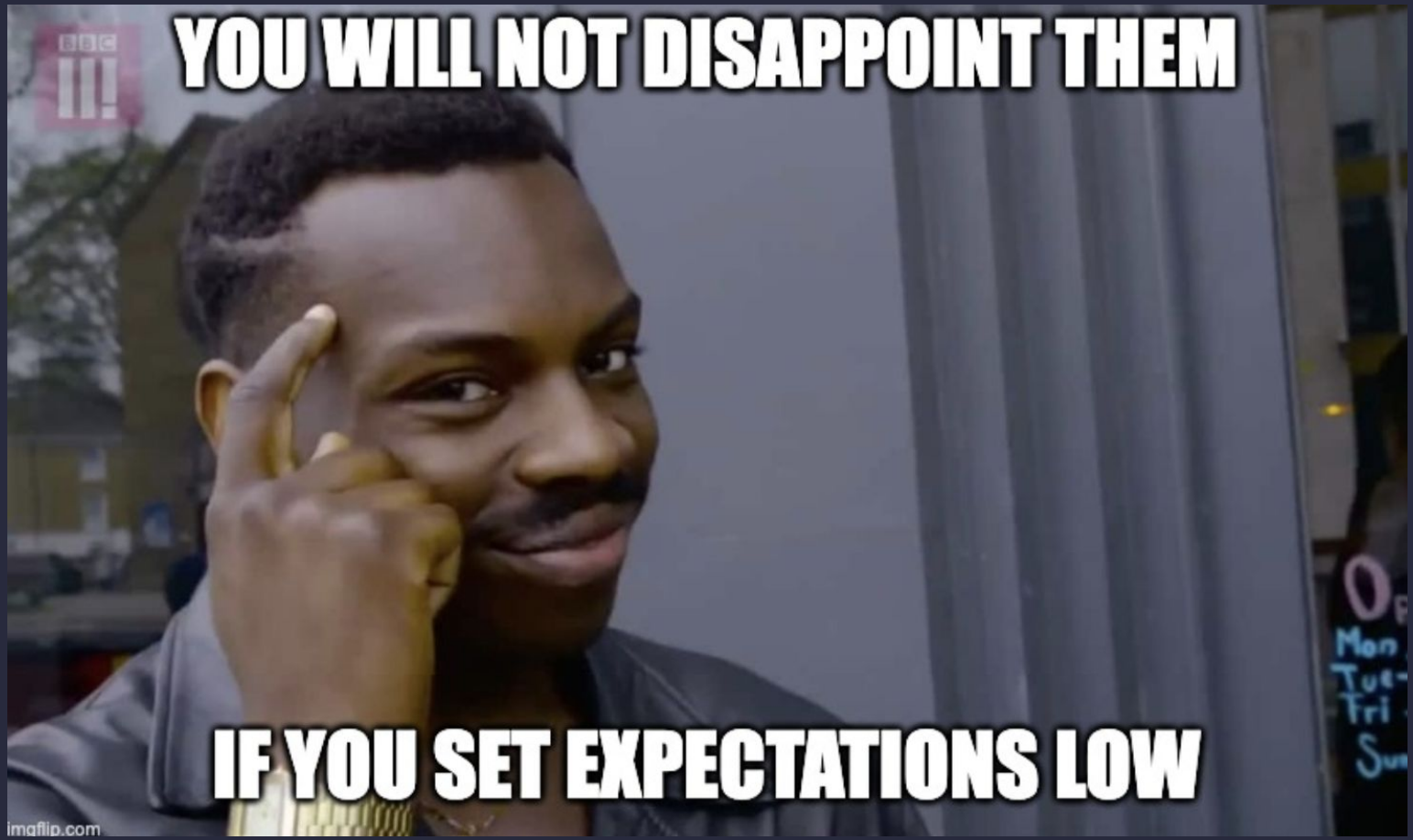
Who Am I? 🤔

- freelance Software Dev working @ Flexys
- started using Scala in 2018 📆
- beginner in mind 🧠
- selfish (did the talk for myself)
- love writing readable code 🗣️

Disclaimer

Level of this talk: entry level (or even below) 😲

* Also contains some idiotic jokes


YOU WILL NOT DISAPPOINT THEM
IF YOU SET EXPECTATIONS LOW
imgflip.com

What I Will Not Talk About? ❌

What are unit tests?

What is ZIO?

What is an effect?

Why should you have unit tests in your project?

What I Will (Try To) Talk About? ✅

How to start with ZIO Test?

How to use ZIO Test?

How to test effects with ZIO Test?

Hot to use property based testing with ZIO Test?

How can you benefit from ZIO Test?

But Why? 🤷

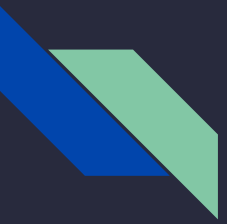
Problems with:

- leaking resources
- concurrency
- dependencies on other services
- multiple versions & platforms
- Future’s

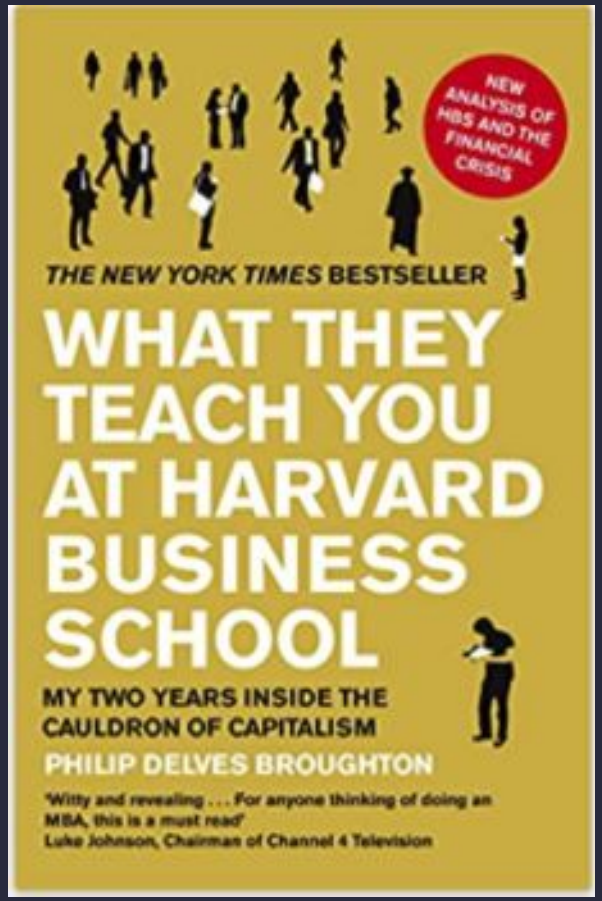
We Already Have It! 👌

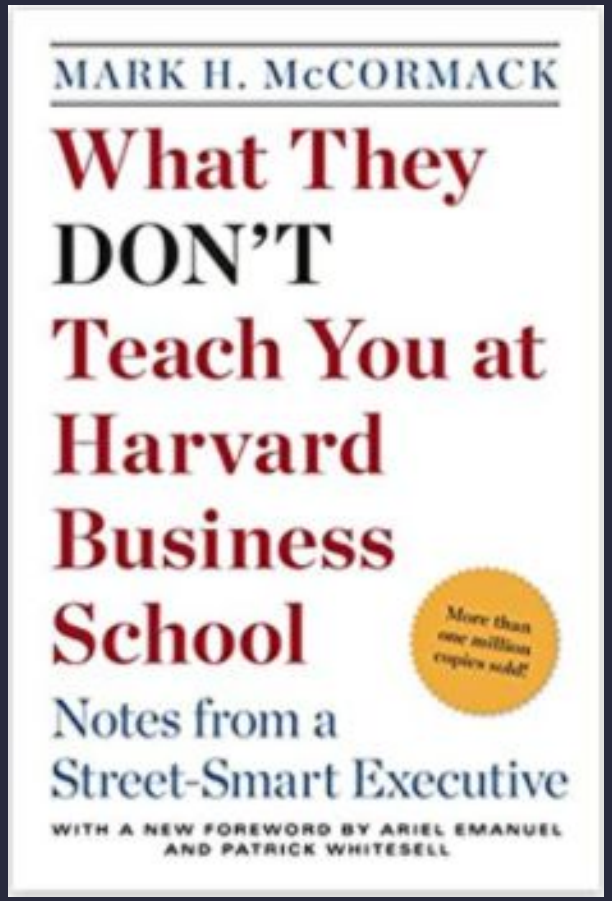
Functional effect systems:

- referential transparency
- resource safety
- environment type
- interruptibility



If You Don’t Know ZIO Or ZIO Test...


NEW ANALYSIS OF HBS AND THE FINANCIAL CRISIS
THE NEW YORK TIMES BESTSELLER
WHAT THEY TEACH YOU AT HARVARD BUSINESS SCHOOL
MY TWO YEARS INSIDE THE CAULDRON OF CAPITALISM
PHILIP DELVES BROUGHTON
'Witty and revealing . . . For anyone thinking of doing an MBA, this is a must read'
Luke Johnson, Chairman of Channel 4 Television


MARK H. McCORMACK
What They DON'T Teach You at Harvard Business School
More than one million copies sold!
Notes from a Street-Smart Executive
WITH A NEW FOREWORD BY ARIEL EMANUEL AND PATRICK WHITESELL

Let’s Start

```
val zioVersion = "some_version"

libraryDependencies ++= Seq(
 "dev.zio" %% "zio-test"          % zioVersion % "test",
 "dev.zio" %% "zio-test-sbt"      % zioVersion % "test"
)
```

Let’s Begin With Something Simple! 🤯

```
trait TestAspect[+R0, -R1, +E0, -E1, +S0, -S1] {
def apply[R >: R0 <: R1, E >: E0 <: E1, S >: S0 <:
S1](spec: ZSpec[R, E, L, S]): ZSpec[R, E, L, S]
}
```

Our First Test 🚀

```
object InitialTest extends DefaultRunnableSpec {
 def spec = suite("Example Spec")(
     test("Introduce someone") {
       val result = introduceSomeone("Paul")
       assert(result)(equalTo("Hello, this is Paul"))
     }
   )
}
```

Tests As Effects 🥰

No more `.unsafeRunSync()`

```
testM("expected values") {
 for {
   result <- someStream.take(3).runCollect
 } assert(result)(equalTo(someExpectedStream))
}
```

Environment Is Also Here 👉🏽

```
testM("expect call for overloaded method") {
 val app = random.nextInt
 val env = MockRandom.NextInt(value(42))
 val out = app.provideLayer(env)
 assertM(out)(equalTo(42))
}
```

Resources ⤵️

```
val myLayer: ULayer[MyLayer] = ZLayer.fromManaged { ... }


def spec = suite("Resources")(
    testM("My Test"){
      ???
    }.provideCustomLayer(myLayer)
)
```

Shared Resources 🤝

```
val myHyperExpensiveResource: ULayer[HyperExpensiveResource] =
ZLayer.fromManaged { ... }

override def spec = suite("WithResourceSpec")(
  testM("test 1") { ... },
  testM("test 2") { ... }
).provideLayerShared(myHyperExpensiveResource)
```

Generators? Here You Are 🥳

```
import zio.random.Random
import zio.test.{Gen, Sized}
import zio.test.Gen._


object Generators {
 val name: Gen[Random with Sized, Name] = anyString.map(Name.apply)
 val id: Gen[Random with Sized, Id] = anyInt.map(Id.apply)
}
```

Property Based Testing 🧪

```
testM("Test Generators") {
        check(Generators.name) { name =>
          assert(name.value)(isNonEmptyString)
        }
      }
```

Aspects 🤩

ZIO Test provides you:

- nonFlaky
- flaky
- timeout
- ignore
- jvmOnly
- after

Just tag your test or suite eg. `@@ nonFlaky @@ jvmOnly`

Assertions 🤓

```
assert(result)(hasLength(equalTo(5))) &&
assert(result)(isSorted)
```

OR

```
assertEqual(Option("Paul").get, "Paul")
```

Exception Testing 💥

```
object ThrowableSpec extends DefaultRunnableSpec {
 def spec = suite("Exception Suite")(
   testM("Example of testing for expected failure") {
     assert(ZIO.fail("fail").run)(fails(equalTo("fail")))
   }
 )
}
```

Reporting 🗂️

```
+ Initial Spec
  + Example suite
    + Introduce someone
    + Do more stuff
    + Test #3
    + Test #4
  + Another Suite
    + Some Test
Ran 5 tests in 1 s 527 ms: 5 succeeded, 0 ignored, 0 failed
```

Reporting 🗂️

```
- Initial Spec
  + Example suite
    + Introduce someone
    + Do more stuff
    + Test #3
    + Test #4
  - Another Suite
    + Some Test
    - Failing Test
      Paul did not satisfy isEmptyString()
      Right(Paul) did not satisfy isRight(isEmptyString())
      `result` = Some(Right(Paul)) did not satisfy isSome(isRight(isEmptyString()))
      at /Users/marcinkrykowski/Documents/zio_test/src/test/scala/InitialTest.scala:39
Ran 6 tests in 1 s 149 ms: 5 succeeded, 0 ignored, 1 failed
- Initial Spec
  - Another Suite
    - Failing Test
      Paul did not satisfy isEmptyString()
      Right(Paul) did not satisfy isRight(isEmptyString())
      `result` = Some(Right(Paul)) did not satisfy isSome(isRight(isEmptyString()))
      at /Users/marcinkrykowski/Documents/zio_test/src/test/scala/InitialTest.scala:39
```

Conclusion 💁

Amazing community!

Well organised code

Many resources to learn from

Easy to onboard - great dev experience

Available to everyone, thanks to interop package

ZIO 2.0 just around the corner

Go, explore, and have fun writing your tests using ZIO Test.

References 📚

Wiem Zine Elabidine - https://medium.com/@wiemzin/get-started-with-zio-test-7a27da355498

Adam Fraser - multiple ZIO Test talks

Pavels Sisojevs - https://scala.monster/zio-test

ZIO Test documentation & zio.dev

Special thanks 🙏

- You (who watched)
- ZIO contributors
- ZIO users
- Adam Fraser
- John De Goes
- Medidata & Signify

Questions? 

Twitter: @marcinkrykowski

Linkedin: /marcinkrykowski

Github: github.com/marcinkrykowski

Email: marcin.krykowski@gmail.com