SOLVING TODAY AND TOMORROW'S CHALLENGES

Crafting code

Guillaume (G) Belrose / @gbelrose

Testing with properties, or how to test more by typing less

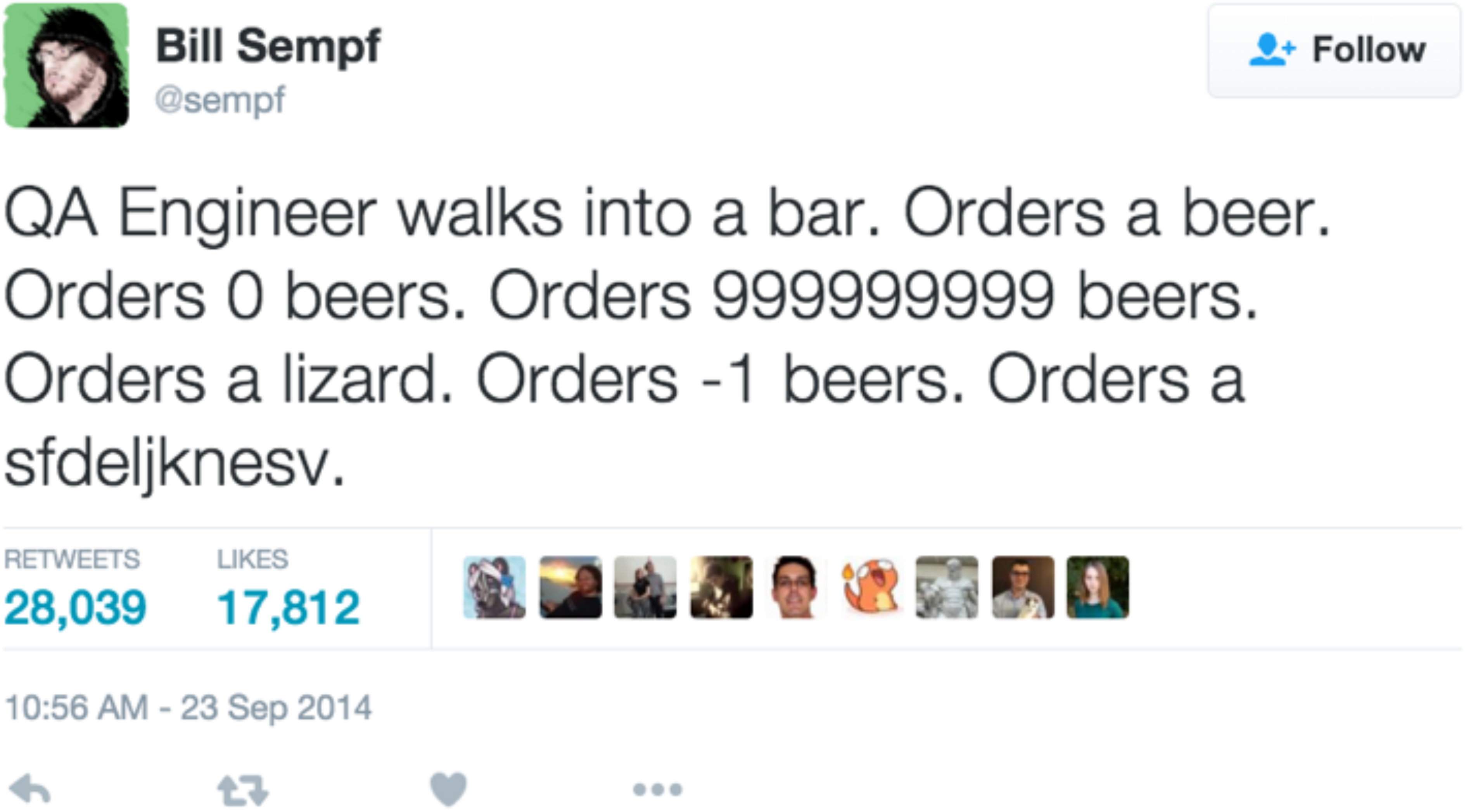
What is a test?

- “Any **automated** process used to check that a system **functions** correctly given a set of **assumptions** (input values)”

- Test **!=** Proof

Spot the (deadly) bug….

The old-fashion way



Bill Sempf
@sempf

QA Engineer walks into a bar. Orders a beer. Orders 0 beers. Orders 999999999 beers. Orders a lizard. Orders -1 beers. Orders a sfdeljknesv.

RETWEETS 28,039
LIKES 17,812

10:56 AM - 23 Sep 2014

Haskell QuickCheck

Let the machines do the work

- Define **properties** of test values
- Tool **generates** lots of values for you
- **Shrinking**: tool finds the buggy value for you
- Profit!!!!

ScalaTest, ScalaCheck

Spot the (deadly) bug, part 2

```
/** An awesome blog engine which is going to change the world.*/
object Blog{

  /** Create a blog post with some text.*/
  def write(blogName: String, blogContent: String) : Unit = {
    Utils.writeStringToFile(new File(blogName), blogContent)
  }

  /** Fetch all the text from a blog post.*/
  def read(blogName: String) : String = {
    Utils.readFileToString(new File(blogName))
  }
}
```

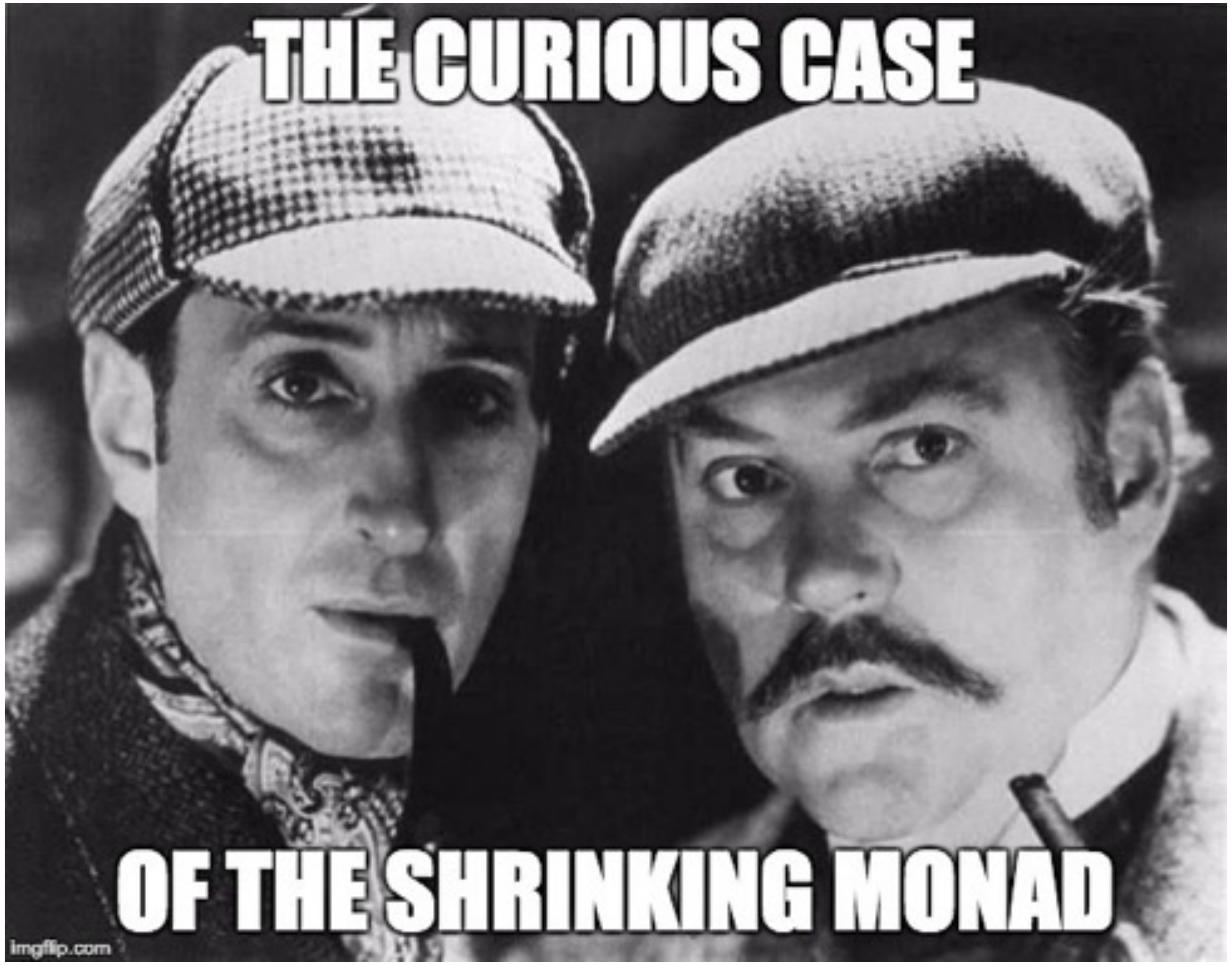
Test you might write

```scala
class NaiveBlogSpec extends WordSpec with Matchers{

  "The blog" should {

    "correctly record text" in {

      val text =
        """
          |L’iPhone constitue la vache à lait du groupe.
          |Il représente plus de 60 % du chiffre d’affaires et encore davantage en termes de profits."""

      Blog.write("infrench.txt", text)
      Blog.read("infrench.txt") should be (text)
    }
  }
}
```

Test you should write

```
class PropertyBasedBlogSpec extends WordSpec with Matchers with GeneratorDrivenPropertyChecks {

  implicit override val generatorDrivenConfig = PropertyCheckConfig(
    minSuccessful = 100,
    minSize=50
  )

  val strings = Arbitrary.arbitrary[String]

  "The blog" should {

    "correctly record arbitrary text" in {

      forAll(strings) { s: String =>
        println(s)
        Blog.write("random.txt",s)
        Blog.read("random.txt") should be (s)
      }
    }
  }
}
```

Is this magic?



Monads

- “A **pattern** used to represent a computation in such a way that it can be **modified**, **chained** or **composed**”

Monads

Examples

```scala
val numbers = Arbitrary.arbitrary[Int]

val evenNumbers = numbers.filter( n => n % 2 == 0)

val evenPairs = for{
  a <- evenNumbers;
  b <- evenNumbers
} yield (a,b)

val fruits = Gen.oneOf(“Apple","Blueberry","Banana")

val bakings = Gen.oneOf("tart", "crumble", “muffin")

val desserts = for {
  quantity <- positiveNumbers;
  fruit <- fruits;
  baking <- bakings
} yield (quantity, fruit + " " + baking)
```

Math spec

```
class MathSpec extends WordSpec with Matchers with
GeneratorDrivenPropertyChecks {

  val numbers = Arbitrary.arbitrary[Int]

  "The square root of a number squared" should {

    "always be the number itself" in {

      forAll(numbers){ n =>
        Math.sqrt( n * n ) should be (n)
      }
    }
  }
}
```

Use cases

- Serialisation / Deserialisation
- System-wide logic (app-> DB -> app)
- Compilers
- User Interfaces
- Testing automotive software
- Distributed systems and fault tolerance

Not your type?

> Java, C++

> C# F# (FsCheck)

> Erlang (QuickCheck), Ruby, Clojure (test.check)

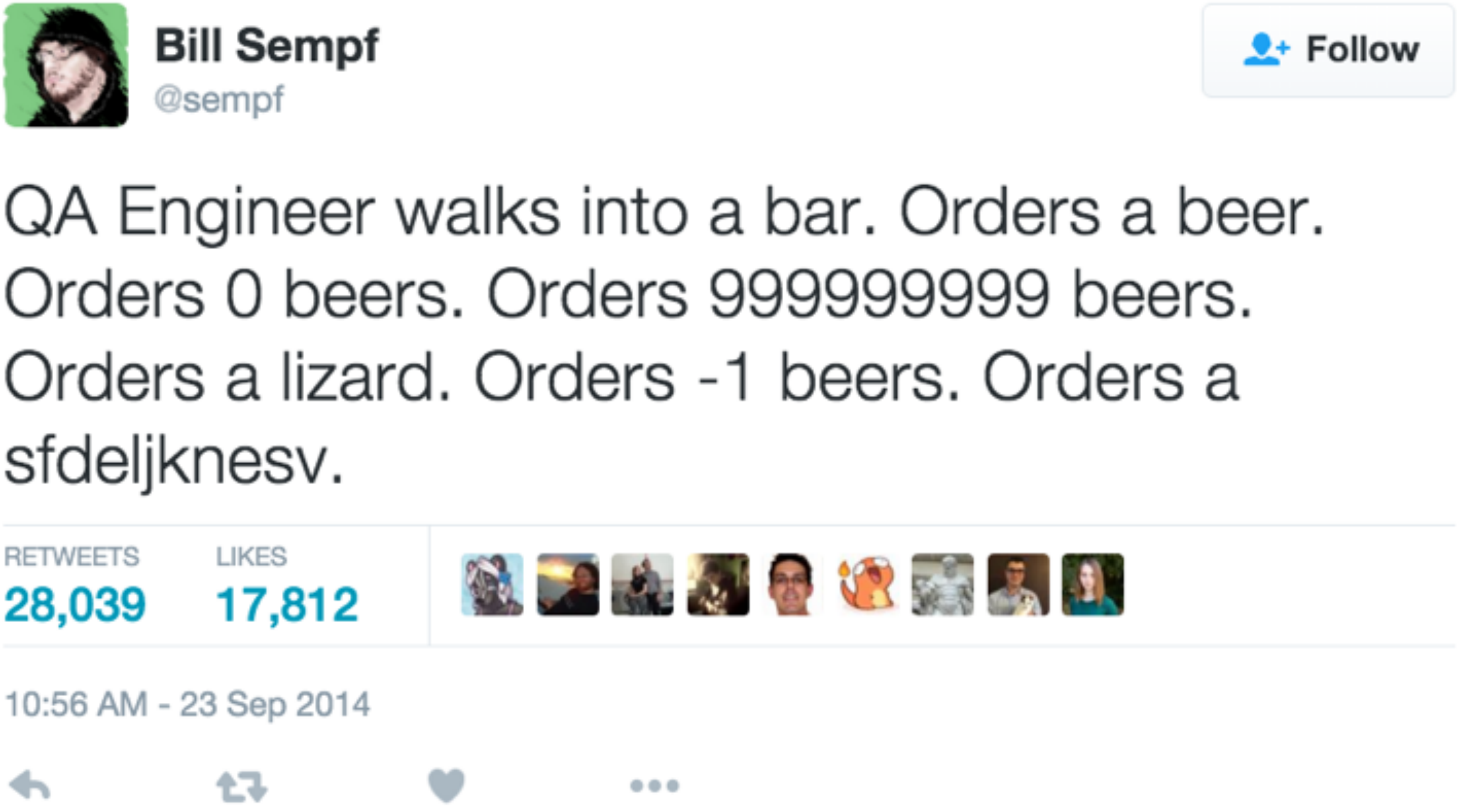
> etc…

Let the machines do the work

“Dont’ **write** tests, **generate** them”

John Hughes

Out with the old….



Bill Sempf
@sempf

QA Engineer walks into a bar. Orders a beer. Orders 0 beers. Orders 999999999 beers. Orders a lizard. Orders -1 beers. Orders a sfdeljknesv.

RETWEETS 28,039 LIKES 17,812

10:56 AM - 23 Sep 2014

In with the new

“A 蕾隟 walks into a bar and order 1227633174 訉 ?洅巽媥 〔刈昙粭.”

devconf;

SOLVING TODAY AND TOMORROW'S CHALLENGES

/* Merci */

Guillaume Belrose

@gbelrose
kafecho@gmail.com
083 227 0908

http://www.devconf.co.za/