

# SMARTER TESTING WITH SPOCK

PETER NIEDERWIESER  
PRINCIPAL ENGINEER, GRADLEWARE

# WHAT WE'LL TALK ABOUT

- ✱ Spock?!
- ✱ State Based Testing
- ✱ Data Driven Testing
- ✱ Interaction Based Testing
- ✱ Spock Extensions
- ✱ More Cool Stuff

SPOCK?!

# SPOCK IS...

- ✱ A developer testing framework...
- ✱ for Groovy and Java applications...
- ✱ based on Groovy...
- ✱ fully compatible with JUnit...
- ✱ but going beyond!



# SPOCK CAN...

- ✱ Reduce the lines of test code
- ✱ Make tests more readable
- ✱ Turn tests into specifications
- ✱ Be extended in powerful ways
- ✱ Bring back the fun to testing!

# GETTING STARTED

- ✿ Homepage  
<http://spockframework.org>
- ✿ Source Code  
<https://github.com/spockframework/spock>
- ✿ Spock Web Console  
<http://meet.spockframework.org>
- ✿ Spock Example Project  
<http://downloads.spockframework.org>  
<https://github.com/spockframework/spock/tree/groovy-1.8/spock-example>
- ✿ Slides and Code for this Presentation  
<https://github.com/spockframework/smarter-testing-with-spock>

# WHO'S USING SPOCK?

Gradle

GPars

Grails

Geb

Grails Plugin Collective

Apache Tapestry

Griffon

Spock

Spring?



# WHO'S USING SPOCK? (2)

be2

eHarmony

CTI Digital

Smarter Ecommerce

BSkyB

bemoko

Energized Work

IntelliGrape Software

SystemsForge

Software Projects

Gennemtænkt IT



# STATE BASED TESTING

# STATE BASED TESTING

- ✱ Classical Unit Testing

- ✱ Arrange

- ✱ Act

- ✱ Assert

- ✱ Given-When-Then

# STATE BASED TESTING

- ✱ Classical Unit Testing

- ✱ Arrange

- ✱ Act

- ✱ Assert

- ✱ Given-When-Then

*Show me the code!*



# RECAP: STATE BASED TESTING

- ✱ Blocks

setup: cleanup: expect: given: when: then:  
where: and:

- ✱ Fixture Methods

setup() cleanup() setupSpec() cleanupSpec()

- ✱ Instance and @Shared fields

- ✱ old() and thrown()

# DATA DRIVEN TESTING

# DATA DRIVEN TESTING

- ✱ Test the same behavior...
- ✱ with varying data!



# DATA DRIVEN TESTING

- ✱ Test the same behavior...

- ✱ with varying data!

*Show me the code!*

# RECAP: DATA DRIVEN TESTING

- ✱ where: block
- ✱ Data tables
- ✱ External data sources
- ✱ @Unroll

# INTERACTION BASED TESTING



# INTERACTION BASED TESTING

- ✱ Design and test how your objects communicate
- ✱ Mocking frameworks to the rescue
- ✱ Spock comes with its own mocking framework

# INTERACTION BASED TESTING

- ✱ Design and test how your objects communicate
  - ✱ Mocking frameworks to the rescue
  - ✱ Spock comes with its own mocking framework
- Show me the code!*

# RECAP: INTERACTION BASED TESTING

## ✻ Creating

```
def sub = Mock(Subscriber)  
Subscriber sub = Mock()
```

## ✻ Mocking

```
1 * sub.receive("msg")  
(1..3) * sub.receive(_)  
(1.._) * sub.receive(_ as String)  
1 * sub.receive(!null)  
1 * sub.receive({it.contains("m")})  
1 * _./rec.*/"msg")
```



# RECAP: INTERACTION BASED TESTING (2)

## ☼ Stubbing

```
// now returns status code
```

```
String receive(String msg) { ... }
```

```
sub.receive(_) >> "ok"
```

```
sub.receive(_) >>> ["ok", "ok", "fail"]
```

```
sub.receive(_) >>> { msg -> msg.size() > 3 ? "ok" :  
"fail" }
```

## ☼ Mocking and Stubbing

```
3 * sub.receive(_) >>> ["ok", "ok", "fail"]
```

## ☼ Impressing your friends

```
(_.._) * _._(*_) >> _
```

# SPOCK EXTENSIONS

# SPOCK EXTENSIONS

- ✿ Listeners
- ✿ Interceptors
- ✿ Annotation-driven extensions
- ✿ Global extensions

# BUILT-IN EXTENSIONS

- ✿ @Ignore
- ✿ @IgnoreRest
- ✿ @FailsWith
- ✿ @Timeout
- ✿ @AutoCleanup
- ✿ @Stepwise
- ✿ @RevertMetaClass
- ✿ @Rule



# BUILT-IN EXTENSIONS

✱ @Ignore

✱ @IgnoreRest

✱ @FailsWith

✱ @Timeout

✱ @AutoCleanup

✱ @Stepwise

✱ @RevertMetaClass

✱ @Rule

*Show me the code!*

# EXTERNAL EXTENSIONS

- ✿ spock-grails
- ✿ spock-spring
- ✿ spock-guice
- ✿ spock-tapestry
- ✿ spock-unitils
- ✿ spock-griffon
- ✿ spock-arquillian
- ✿ spock-extensions <http://github.com/robletcher/spock-extensions>

# GRAILS EXTENSION

- ✱ <http://grails.org/plugin/spock>
- ✱ <https://github.com/spockframework/spock-grails>
- ✱ `grails install plugin spock 0.6-SNAPSHOT`
- ✱ `grails test-app`
- ✱ `grails test-app integration:spock 'C*'`



# GRAILS EXTENSION (2)

```
class MyUnitSpec extends UnitSpec {  
    def "domain mocking"() {  
        setup:  
            mockDomain(Person)  
  
        when:  
            new Person(name: name).save()  
  
        then:  
            Person.findByName(name) != null  
  
        where:  
            name = "bill"  
    }  
}
```



# SPRING EXTENSION

```
class InjectionExamples extends Specification {  
    @Autowired  
    IService1 byType  
  
    @Resource  
    IService1 byName  
  
    @Autowired  
    ApplicationContext context  
}
```

# OTHER COOL STUFF

# CONFIGURING SPOCK

- ✱ ~/.spock/SpockConfig.groovy, or on class path, or with -Dspock.configuration

```
runner {  
    filterStackTrace false  
    include Fast  
    exclude Slow  
    optimizeRunOrder true  
}
```

```
@Fast  
class MyFastSpec extends Specification {  
    def "I'm fast as hell!"() { expect: true }  
  
    @Slow  
    def "sorry, can't keep up..."() {  
        expect: false  
    }  
}
```



# TOOLING

- ✱ Eclipse, IDEA
- ✱ Ant, Maven, Gradle
- ✱ Jenkins, Bamboo, TeamCity
- ✱ Spock runs everywhere JUnit and Groovy run!



# SPOCK UNDER THE HOOD

☼ You write...

```
a = 1; b = 2; c = 4
expect: sum(a, b) == c
```

☼ Spock generates...

```
def rec = new ValueRecorder()
verifier.expect(
    rec.record(rec.record(sum(rec.record(a),
        rec.record(b))) == rec.record(c)))
```

☼ You see...

```
sum(a, b) == c
|   |   |   |   |
3   1   2   |   4
           false
```

# Q&A

- ✻ Homepage  
<http://spockframework.org>
- ✻ Source Code  
<https://github.com/spockframework/spock>
- ✻ Spock Web Console  
<http://meet.spockframework.org>
- ✻ Spock Example Project  
<http://downloads.spockframework.org>  
<https://github.com/spockframework/spock/tree/groovy-1.8/spock-example>
- ✻ Slides and Code for this Presentation  
<https://github.com/spockframework/smarter-testing-with-spock>