Smarter Testing With Spock

Peter Niederwieser
Principal Engineer, Gradleware

What we'll talk about

- What is Spock?
- State Based Testing
- Data Driven Testing
- Interaction Based Testing
- Spock Extensions
- More Cool Stuff

What is 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!

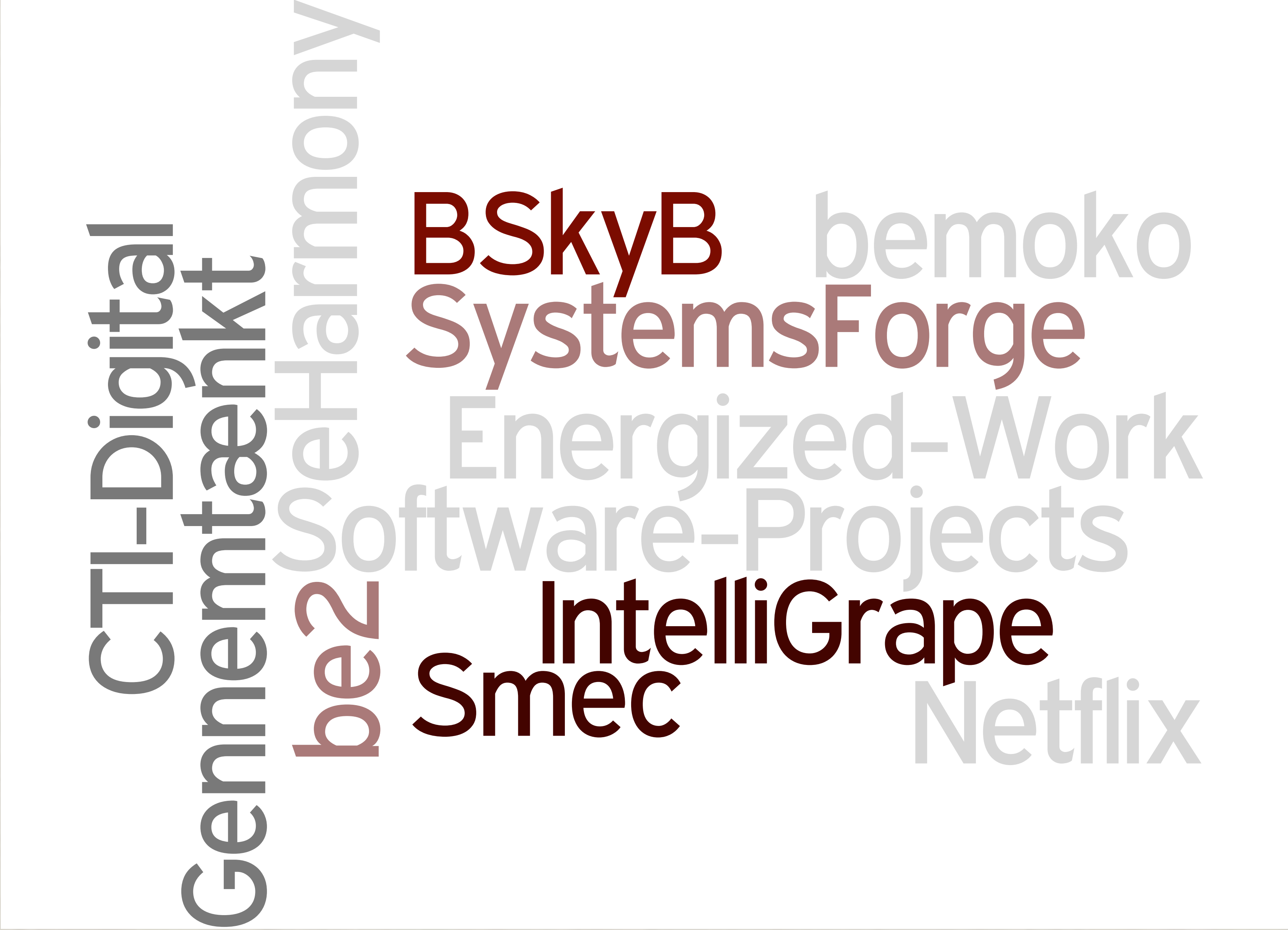
Getting Started

- Homepage
  http://spockframework.org
- Source Code
  http://github.spockframework.org/spock
- Documentation
  http://docs.spockframework.org/en/latest
- Spock Web Console
  http://webconsole.spockframework.org
- Spock Example Project
  http://github.spockframework.org/spock-example
- Slides and Code for this Presentation
  http://github.spockframework.org/smarter-testing-with-spock

Who's Using Spock?

Spock
GPars
Griffon
Grails
Tapestry
Asgard
Gradle
Geb
Spring

Who's Using Spock? (2)



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

- How objects communicate
- Mocking frameworks simplify the job
- Spock has its own mocking framework

Interaction Based Testing

- How objects communicate
- Mocking frameworks simplify the job
- Spock has 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

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

New in 0.7

What's to come?

New in 0.7 (1)

- Group conditions/interactions w/ same target

```
with(person) {
    name == “Fred”
    age == 30
    sex == “male”
}

with(service) {
    1 * start()
    1 * stop()
}
```

- Better mocking failure messages

New in 0.7 (2)

- Stubs

```
def person = Stub(Person)
```

- Spies

```
def person = Spy(Person, constructorArgs: ["Fred", 30])
```

- Groovy mocks

```
def person = GroovyMock(person) // GroovyStub, GroovySpy
```

- Declare interactions upfront

```
def person = Stub(Person) {
    1 * sleep()
    sing() >> “tra-la-la”
}
```

- More at:

http://docs.spockframework.org/en/latest/new_and_noteworthy.html

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

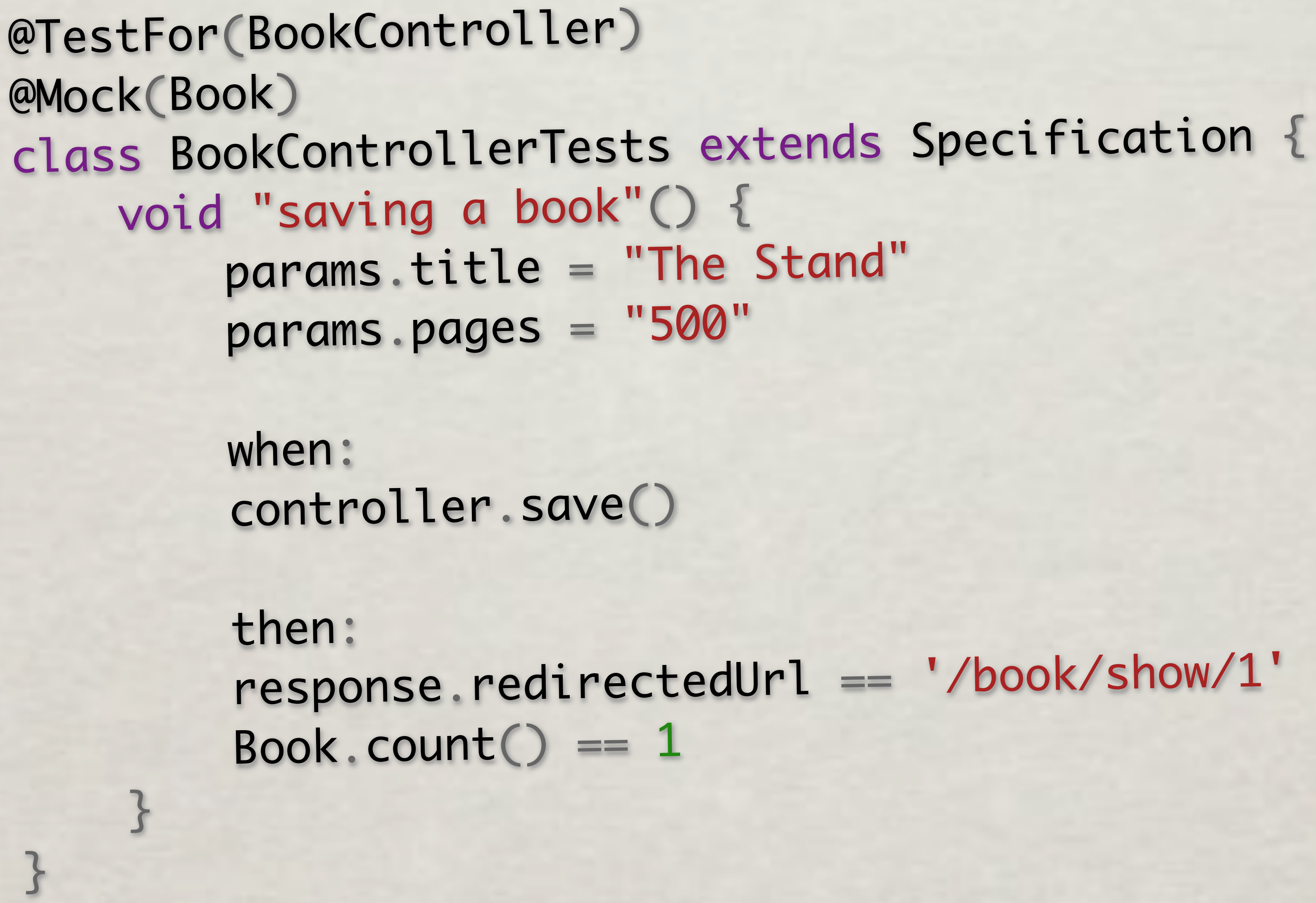
- spock-grails
- spock-spring
- spock-guice
- spock-tapestry
- spock-unitils

- spock-griffon
- spock-arquillian
- spock-extensions http://github.com/robfletcher/spock-extensions

Grails Extension

- http://grails.org/plugin/spock
- http://github.spockframework.org/spock-grails
- grails install plugin spock 0.6
- grails test-app
- grails test-app integration:spock ‘C*’

Grails Extension (2)



```
@TestFor(BookController)
@Mock(Book)
class BookControllerTests extends Specification {
    void "saving a book"() {
        params.title = "The Stand"
        params.pages = "500"

        when:
        controller.save()

        then:
        response.redirectedUrl == '/book/show/1'
        Book.count() == 1
    }
}
```

Spring Extension

```
@ContextConfiguration(locations = ["my-app-context.xml"])
class InjectionExamples extends Specification {
    @Autowired
    Service1 injectedByType

    @Resource
    Service1 injectedByName

    @Autowired
    ApplicationContext context
}
```

More 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()
SpockRuntime.verifyCondition(
    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
  http://github.spockframework.org/spock
- Documentation
  http://docs.spockframework.org/en/latest
- Spock Web Console
  http://webconsole.spockframework.org
- Spock Example Project
  http://github.spockframework.org/spock-example
- Slides and Code for this Presentation
  http://github.spockframework.org/smarter-testing-with-spock