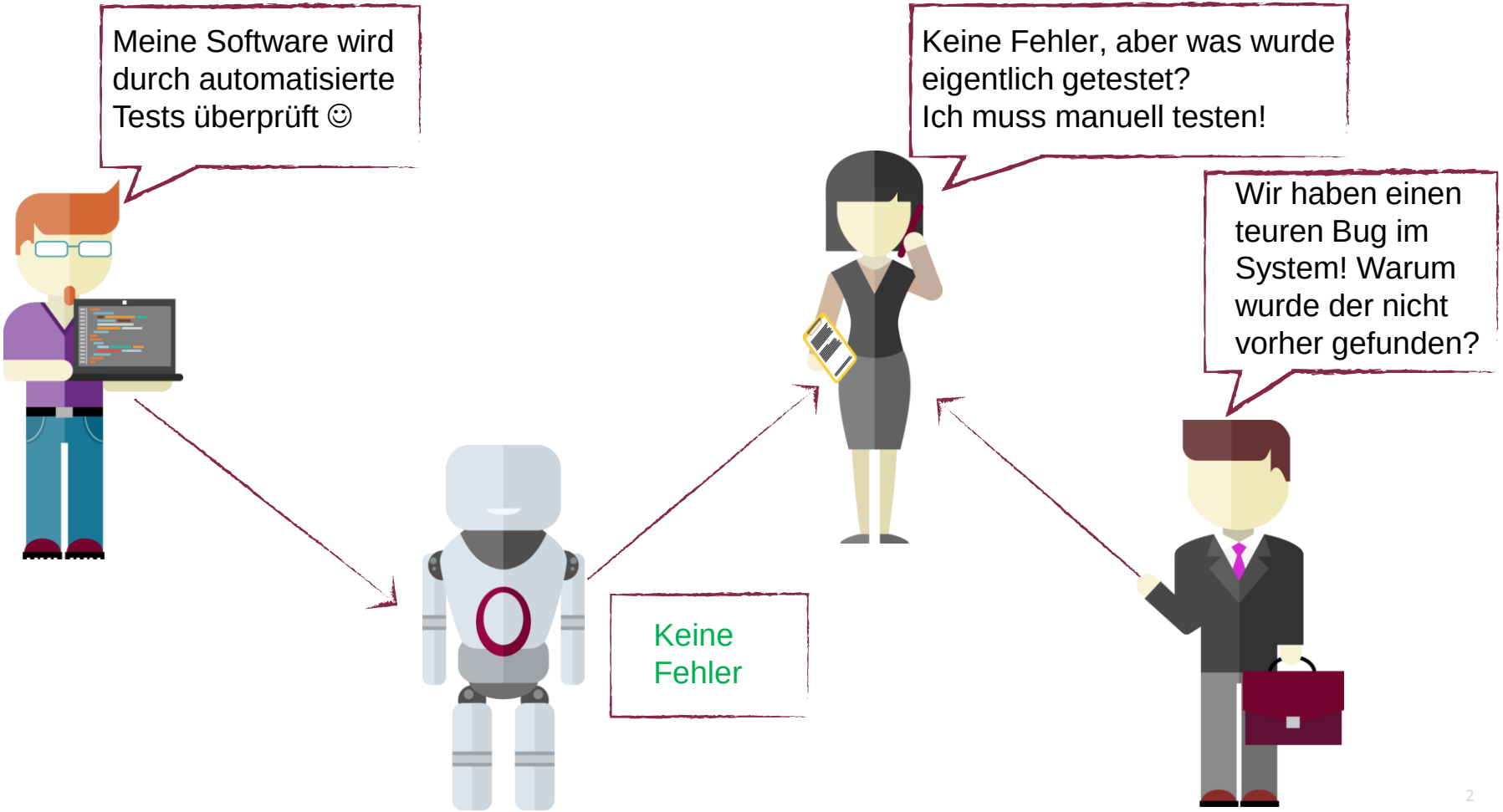
exensio
intelligente informationssysteme

Spock und Geb:
Übersichtliche und nachvollziehbare Tests!

08.03.2016

Ralf D. Müller, Freelancer

Tobias Kraft, exensio GmbH


Meine Software wird durch automatisierte Tests überprüft ☺
Keine Fehler, aber was wurde eigentlich getestet? Ich muss manuell testen!
Wir haben einen teuren Bug im System! Warum wurde der nicht vorher gefunden?
Keine Fehler

Spock – kurz und übersichtlich

Einbindung des Fachbereichs

Verständliches Reporting

Funktionales Testing mit Geb

JUnit vs. Spock

```
public class SpeakerAssignmentTest {
    @Test
    public void
        Talk t
        t.assig
        t.assig
        assertE
        List li

        assertE
    }
}
```

```
class SpeakerAssignmentSpec extends Specification {
    def "assign 2 speakers to talk"() {
        given:
            Talk t = new Talk("Spock", "description")
        when:
            t.assignSpeaker(new Speaker("Ralf","Müller"))
            t.assignSpeaker(new Speaker("Tobias","Kraft"))
        then:
            t.getSpeaker().size() == 2
            t.getSpeaker()*.firstname == ["Ralf", "Tobias"]
    }
}
```

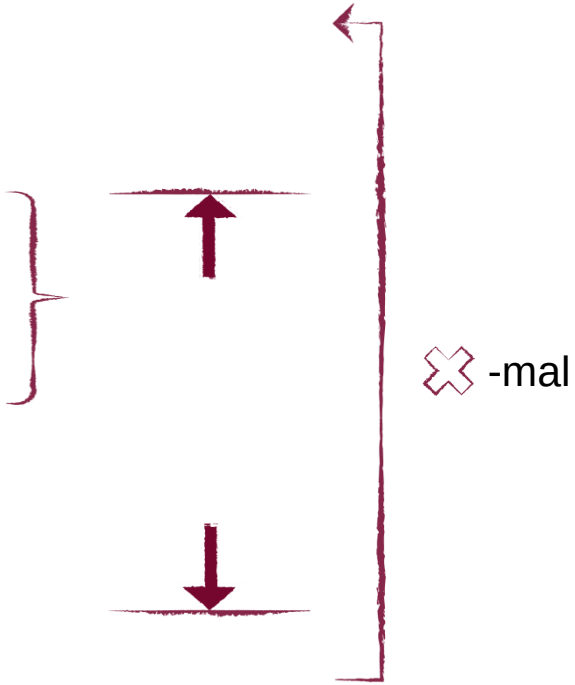
Spock im Überblick

- Test und Spezifikations-Framework

- Groovy ist Basis
- DSL (Domain Specific Language)
- Testing für JVM-fähige Sprachen
- IDE-Unterstützung

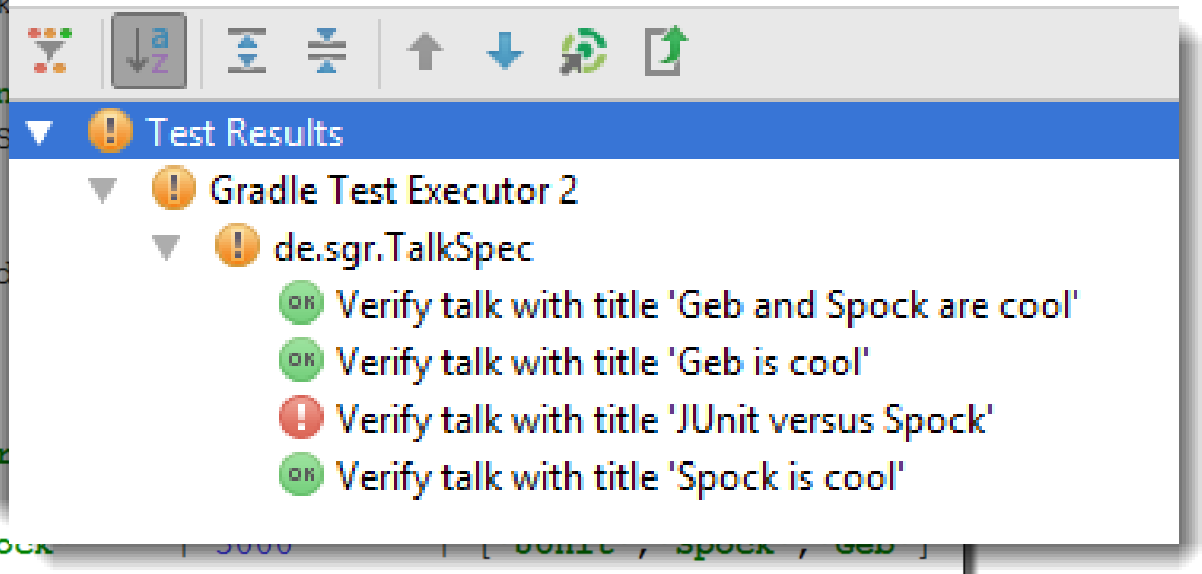
- Version 1.0 seit März 2015 verfügbar

Klare Strukturierung von Feature-Methoden mit Blöcken



Mehrfache Durchläufe

```
class TalkSpec extends Specification {
  @Unroll
  def "Verify talk with title '#title'"() {
    when:
      def description = RandomStringUtils.randomAlphabetic(descLength)
      Talk t = new Talk(
    then: 'check cuttin
      t.getDescriptionS
    and: 'keywords are
      t.validateKeyword
    where:
      title
      "Geb and Spock ar
      "Geb is cool"
      "JUnit versus Spoc
      "Spock is cool"          | 120          | ['Spock']
```



Extensions über Annotationen

```
@Ignore("Not yet implemented")
@IgnoreRest
@IgnoreIf({ properties."os.name" ==~ /Linux.*/) })
```

Ignorieren

Dokumentation

```
@Issue("http://jira.exensio.de/browse/GM-19")
@See("http://ldaley.com/post/6570075743/")
@Title("a readable title")
@Narrative("a beautiful description for a report")
```

```
@Timeout
@FailsWidth
@Requires({ env.containsKey(„DMS_ENABLED") })
@RestoreSystemProperties
```

Weitere Extensions

Was ist sonst noch interessant in Spock?

- Interaction Based Testing
- Stubs, Mocks und Spies
- Exceptions

```
then:
    def e = thrown(IllegalArgumentException)
    e.getMessage().startsWith("No tagCategory")
```

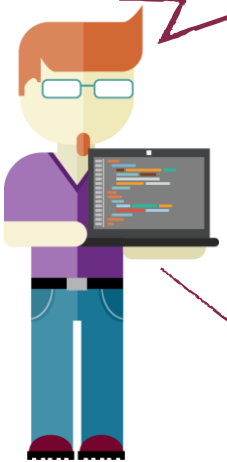
- old Methode

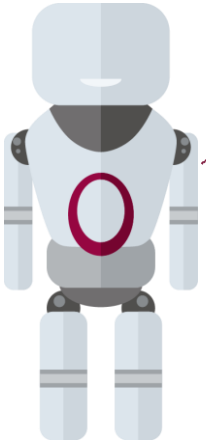
```
when:
        myList.add('foo')
then:
        myList.size() == old(myList.size()) + 1
```

- Hamcrest Matchers

```
expect:
        2.9d closeTo(3, 0.5)
```

Meine automatisierten Tests können jetzt auch vom Fachbereich gelesen werden…

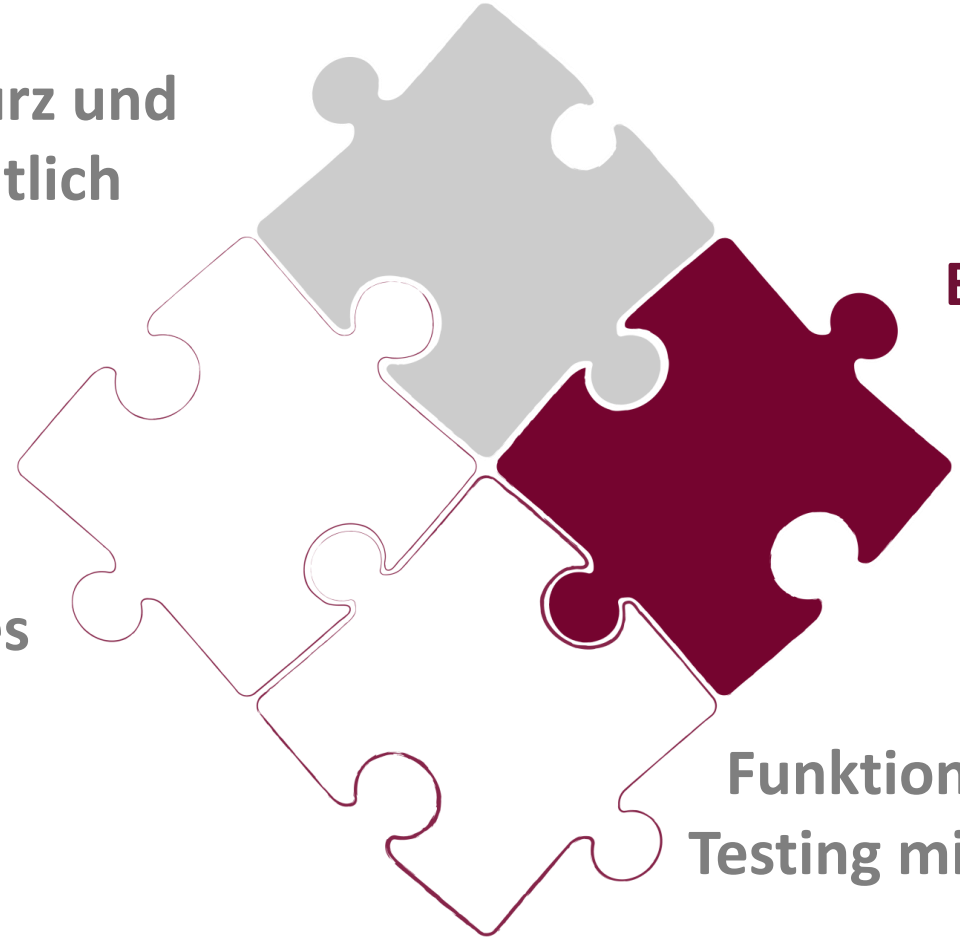




Keine Fehler

Keine Fehler, aber was wurde eigentlich getestet?
Ich spezifiziere es einfach!

Spock – kurz und übersichtlich



Einbindung des Fachbereichs

Verständliches Reporting

Funktionales Testing mit Geb

Spezifikation durch den Fachbereich

```
class TalkSpec extends Specification {
  @Unroll
  def "Verify talk with title '#title'"() {
    when:
      def description = RandomStringUtils.randomAlphabetic(descLength)
      Talk t = new Talk(title, description, keywords)

    then: 'check cutting of description'
      t.getDescriptionShort().length() == Math.min(descLength, 120)

    and: 'keywords are present in talk text'
      t.validateKeywords()

    where:
      title                    | descLength | keywords
      "Geb and Spock are cool" | 170        | ['Spock','Geb']
      "Geb is cool"            | 119        | ['Geb']
      "JUnit versus Spock"     | 3000       | ['JUnit','Spock','Geb']
      "Spock is cool"          | 120        | ['Spock']
  }
```

Spezifikation durch den Fachbereich

	A	B	C	D	E	
1	**Feature**	**Block**	**Beschreibung**	**erwartete Antwort**	**Screenshot**	**Kommentar**
2	**Aufruf der Vortragsbeschreibung**					
3		Gegeben:	Interessent befindet sich auf der Startseite	Startseite	Startseite	
4		Wenn	der Benutzer auf "Vortrag" klickt,	Abstract	Abstract des Vortrags	
5		dann	sieht er die Zusammenfassung des Vortrags			
6	**Ansicht der Referenten**					
7		Gegeben:	Interessent befindet sich auf der Startseite	Startseite		
8		Wenn	der Benutzer auf "Speaker" klickt,	Referenten	Liste Referenten	test
9		dann	sieht er die Beschreibung der 2 Referenten			auf 2 Beschreib
10		und	Name des zweiten Referenten ist "Ralf Müller"			
11	**Zurück**					
12		Gegeben:	Interessent befindet sich auf der Vortrags-Seite	Abstract		
13		Wenn	der Benutzer auf "Home" klickt,	Startseite	Startseite	
14		dann	findet er sich wieder auf der Startseite			

Gegeben:
Wenn
dann
und
wobei

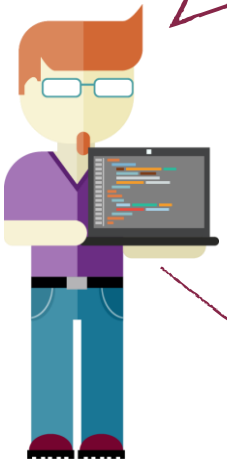
fachlicher Name	URL	technischer Name
Startseite	http://rdmueller.github.io/etka15	StartPage
Abstract	http://rdmueller.github.io/etka15/vortrag.html	AbstractPage
Referenten	http://rdmueller.github.io/etka15/speaker.html	SpeakerPage
Downloads	http://rdmueller.github.io/etka15/downloads.html	DownloadsPage

Spezifikation in Code wandeln

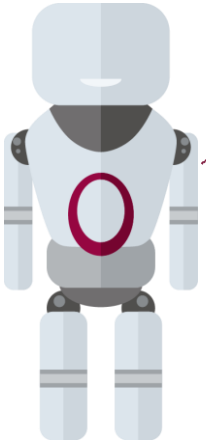
	A	B	C	D	E	F
1	**Feature**	**Block**	**Beschreibung**	**erwartete Antwort**	**Screenshot**	**Kommentar**
2	**Aufruf der Vortragsbeschreibung**					
3		Gegeben:	Interessent befindet sich auf der Startseite	Startseite	Startseite	
4		Wenn	der Benutzer auf "Vortrag" klickt,	Abstract	Abstract des Vortrags	
5		dann	sieht er den ersten Konferenztag			
6	**Ansicht der Referenten**					
7		Gegeben:	Interessent befindet sich auf der Sta			
8		Wenn	der Benutzer auf "Speaker" klickt,			
9		dann	sieht er die Beschreibung der 2 Refe			
10	**Zurück-Link**					
11		Gegeben:	Interessent befindet sich auf der Vo			
12		und	der Benutzer auf "Home" klickt,			
13		dann	findet er sich wieder auf der Start			
14		Gegeben:				

```groovy
import geb.spock.GebReportingSpec
import pages.StartPage
import pages.AbstractPage
import pages.SpeakerPage

class EntwicklertagSpec extends GebReportingSpec {

    def 'Aufruf der Vortragsbeschreibung'() {
        given:  "Interessent befindet sich auf der Startseite"
            at Startseite
            report 'Startseite'
            //TODO: implement test step
        when:   "der Benutzer auf 'Vortrag' klickt,"
            at Abstract
            report 'Abstract des Vortrags'
            //TODO: implement test step
        then:   "sieht er den ersten Konferenztag"
            //TODO: implement test step
    }

```

Screenshot?
Link klicken?
Formular ausfüllen?

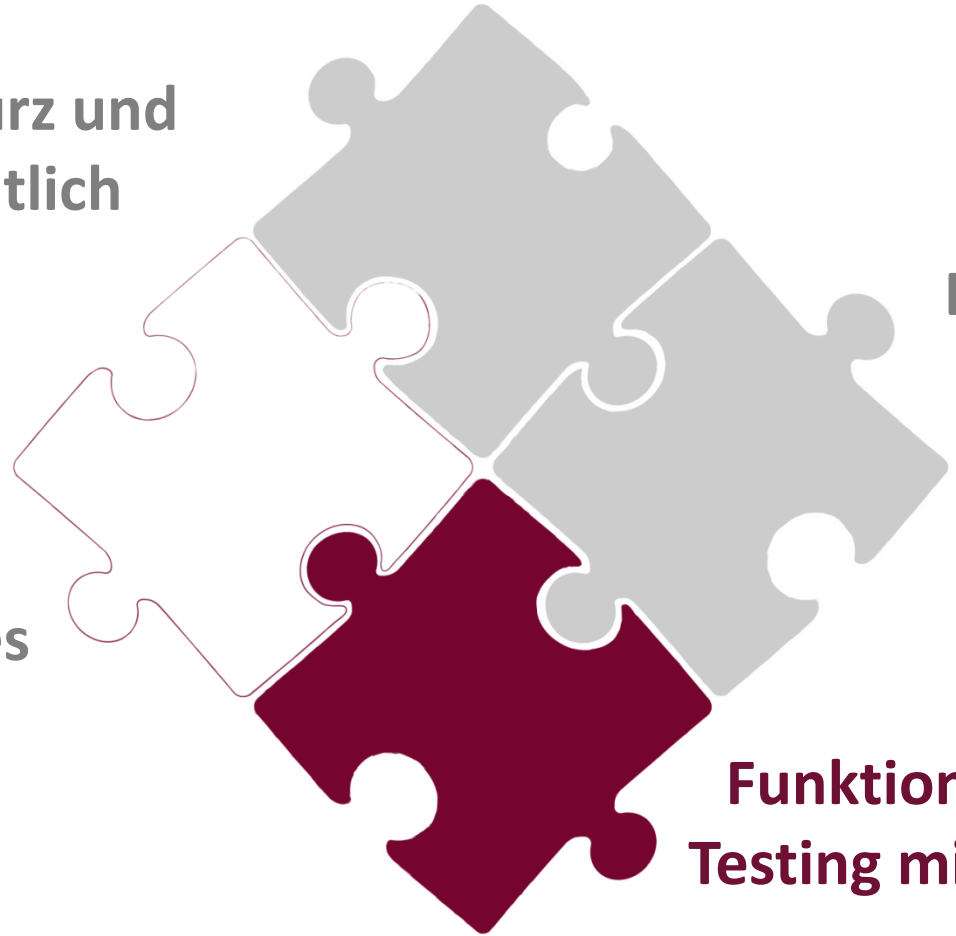


Keine Fehler, aber was wurde eigentlich getestet?
Ich spezifiziere es einfach!



Keine Fehler

Spock – kurz und übersichtlich



Einbindung des Fachbereichs

Verständliches Reporting

Funktionales Testing mit Geb

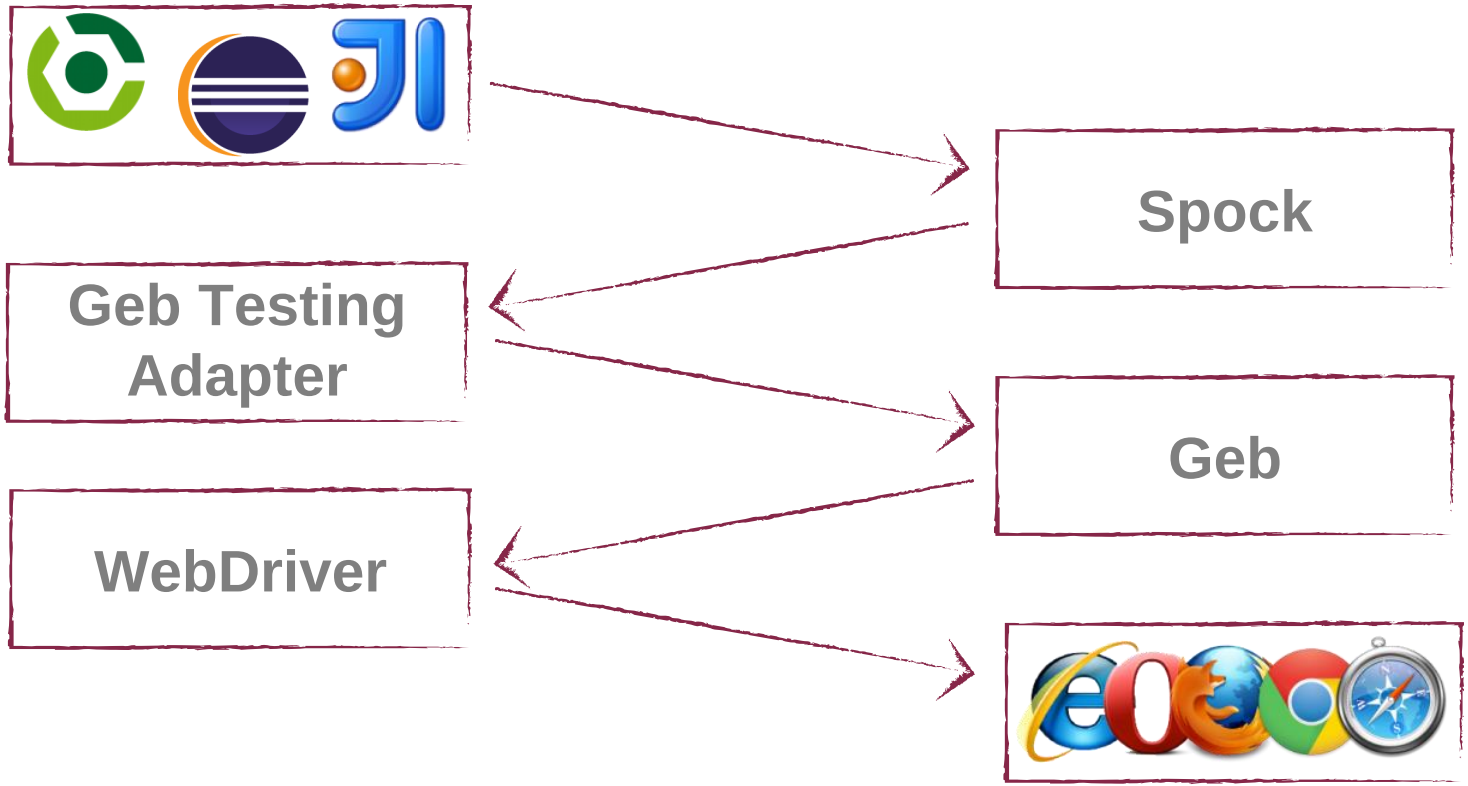
Geb im Überblick

- Webtesting und Screen-Scraping

- DSL auf Groovy basierend
  - jQuery ähnliche Syntax
  - Nutzung von WebDriver
- IDE-Unterstützung
- Integration mit Spock, TestNG oder JUnit

- Version 0.13.1

Spock in Kombination mit Geb



Einfach Selektieren mit JQuery-ähnlicher Syntax

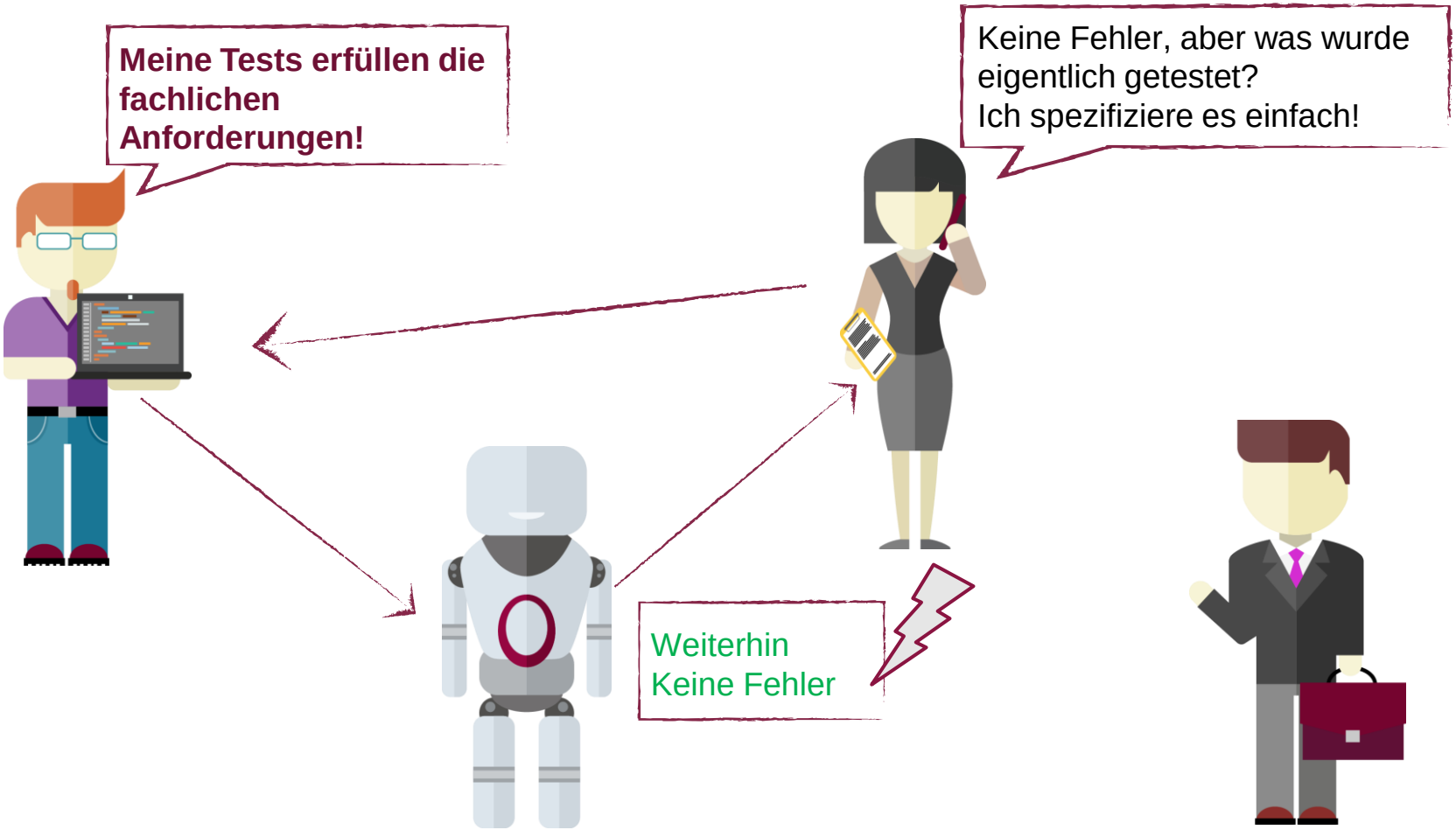
```
class ReportSpec extends GebReportingSpec {

    def "check homepage for talk"() {
        when:
            go "http://rdmueller.github.io/etka15"
        then:
            title =~ "Spock u
            $('li').size() ==
            $("#logo").@alt
        when:
            $("#report").clic
        then:
            title == "Spock T
            waitFor {
                $("div#result"
            }
    }
```

Beispiele für Selektoren
$("div", 2)
$("td.odd", 0..2)*.text()
$("img", alt: "Titel")
$("img", alt: "Titel").previous()
$("div#heading").find("div", 3)
$("div", text: iStartsWith ("groovy"))
$("form").find("input", name: "speaker").value()

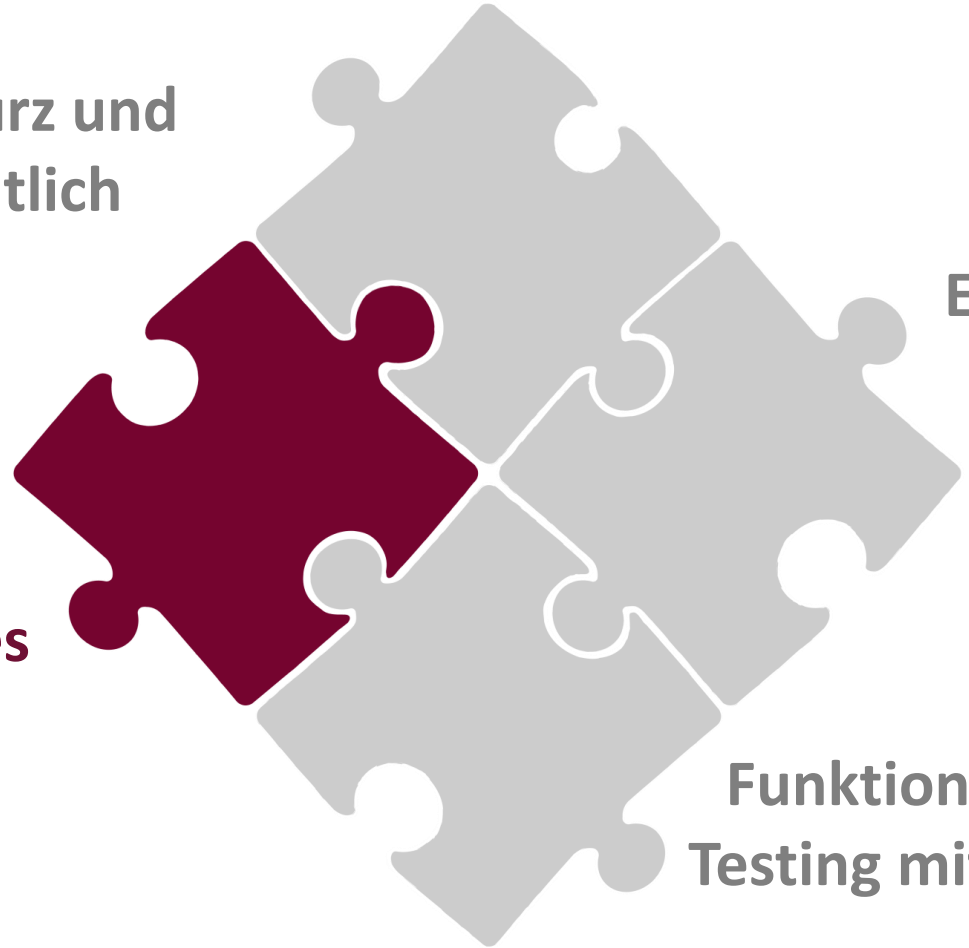
Page Objects – zur Repräsentation von Webseiten

```
class SpeakerPage ext

    static url = "htt

    static at = { tit

    static content =
        homeLink(to:
        mainContent {
        speaker {numb
        speakerName {
        details(requi
    }
```

```
def 'Ansicht der Referenten'() {
    given: "Besucher befindet sich auf der Startseite"
        to StartPage

    when: "der Benutzer auf 'Speaker' klickt,"
        speakerLink.click()

    then: "sieht er die Referenten des Vortrags"
        at SpeakerPage
        report 'Liste Referenten'
        speakerName(1) == 'Ralf D. Müller'
        mainContent.find('h3').size() == 2

    when: "der Benutzer auf 'Home' klickt,"
        homeLink.click()

    then: "befindet er sich wieder auf der Startseite"
        at StartPage
        report 'Startseite'
```


Meine Tests erfüllen die fachlichen Anforderungen!
Keine Fehler, aber was wurde eigentlich getestet?
Ich spezifiziere es einfach!
Weiterhin
Keine Fehler

Spock – kurz und übersichtlich



Einbindung des Fachbereichs

Verständliches Reporting

Funktionales Testing mit Geb

Probleme beim Reporting

- Testing klappt oder klappt nicht
- Power-Assertions sind schon sehr hilfreich

```
"Dies ist ein Test" == "dies ist Ein Test"
                    |
                    false
                    2 differences (88% similarity)
                    (D)ies ist (e)in Test
                    (d)ies ist (E)in Test
```

- 3 Report-Engines

Standard Gradle Reports

Class VortragSpec

all > default-package > VortragSpec

4	1	1	6.526s
tests	failures	ignored	duration

66%
successful

Failed tests | **Tests** | Standard output | Standard error

Test	Duration	Result
Ansicht der Referenten	0.942s	failed
Aufruf der Vortragsbeschreibung	4.744s	passed
Zurueck Link	0.840s	passed
irgendwas anderes	-	ignored

Standard Gradle Reports

Class VortragSpec

all > default-package > VortragSpec

4	1	1	6.568s
tests	failures	ignored	duration

66% successful

Failed tests | Tests | Standard output | Standard error

Ansicht der Referenten

```
Condition not satisfied:

speakerName(1) == 'Ralf Müller'
|              |
Ralf D. Müller false
               3 differences (78% similarity)
               Ralf (D. )Müller
               Ralf (---)Müller

        at VortragSpec.Ansicht der Referenten(VortragSpec.groovy:33)
```

Spock 1.0-SNAPSHOT Report

http://spockframework.github.io/spock/sampleReports/Ninja%20Commander.html

Spock-Reports Plugin von Renato Athaydes

Features:

- Aufruf der Vortragsbeschreibung
- Ansicht der Referenten
- Zurueck Link
- irgendwas anderes

Aufruf der Vortragsbeschreibung

Return

Given: Besucher befindet sich auf der Startseite
When: der Benutzer auf 'Vortrag' klickt,
Then: sieht er die Zusammenfassung des Vortrags

Ansicht der Referenten

Return

Given: Besucher befindet sich auf der Startseite
When: der Benutzer auf 'Speaker' klickt,
Then: sieht er die Referenten des Vortrags
When: der Benutzer auf 'Home' klickt,
Then: befindet er sich wieder auf der Startseite

The following problems occurred:

- Condition not satisfied:

```
speakerName(1) == 'Ralf Müller'
|              |
Ralf D. Müller false
               3 differences (78% similarity)
               Ralf (D. )Müller
               Ralf (---)Müller
```

Zurueck Link

Return

Given: Interessent befindet sich auf der Vortrags-Seite
When: der Benutzer auf 'Home' klickt,
Then: befindet er sich wieder auf der Startseite

https://github.com/renatoathaydes/spock-reports

Aber was wurde getestet?

- Fehler geben Hinweis auf tatsächliche Tests
- Ansonsten geben die Reports nur die Spezifikation wieder

Lösung Reporting – Geb Screenshots

001-001-Aufruf der Vortragsbeschreibung-Startseite.png

001-002-Aufruf der Vortragsbeschreibung-Abstract des Vortrags.png

001-003-Aufruf der Vortragsbeschreibung-Abstract des Vortrags.png

002-001-Ansicht der Referenten-Liste Referenten.png

002-002-Ansicht der Referenten-Startseite.png

003-001-Zurueck Link-Zusammenfassung des Vortrags.png

Lösung Reporting – Spock Reports mit Geb Screenshots

Table of Contents

1. Specification run results
   1.1. Specifications summary
   1.2. Specifications
2. Report for EntwicklertagSpec
   2.1. Summary
   2.2. Features

2.2. Features

2.2.1. Aufruf der Vortragsbeschreibung

Result: **PASS**

- **Gegeben:** Interessent befindet sich auf der Startseite
- **Wenn** der Benutzer auf 'Vortrag' klickt,
- **dann** sieht er die Zusammenfassung des Vortrags

Startseite .

Abstract des Vortrags .

https://github.com/rdmueller/etka15

Lösung Reporting – Spock Reports mit Geb Screenshots

Table of Contents

1. Specification run results
   1.1. Specifications summary
   1.2. Specifications
2. Report for EntwicklertagSpec
   2.1. Summary
   2.2. Features

2.2.2. Ansicht der Referenten

Result: **FAIL**

- **Gegeben:** Interessent befindet sich auf der Startseite
- **Wenn** der Benutzer auf 'Speaker' klickt,
- **dann** sieht er die Beschreibung der 2 Referenten

Liste Referenten .

The following problems occurred:

```
Condition not satisfied:

headlines[1].text() == "Ralf Müller"
|         |   |      |
|         |   |      false
|         |   |      3 differences (78% similarity)
|         |   |      Ralf (D. )Müller
|         |   |      Ralf (---)Müller
|         |   Ralf D. Müller
```

Fazit

- Übersichtliche Tests mit Spock und Geb
- Fachbereich hilft bei Testerstellung

- Weniger manuelle Tests
- Nachvollziehbarkeit der Tests

- Steigerung der Qualität

exensio
intelligente informationssysteme

Fragen?

Vielen Dank!

@RalfDMueller
ralf.d.mueller@gmail.com

@tokraft
tobias.kraft@exensio.de

Referenzen

- Spock und Geb: Übersichtlich und nachvollziehbar Testen für alle!
  Vortrag, Beispiele Source Code
  http://rdmueller.github.io/etka15/
- Teil 1 : Mit Mr. Spock beim Testeinsatz, Javamagazin (12/2014)
  Teil 2 : Mr. Spock ruft Geb, Javamagazin (01/2015)
  http://www.exensio.de/articles/