ONE SUMMIT
November 15-16, 2022
Seattle, WA

Achieving Gold Standard Quality Goals Using Groovy & Spock Testing

UNIT TESTING IN THE 2020'S

ERICSSON ONAP

Toine Siebelink &
Lee Anjella Macabuhay

CPS

1

Unit Testing Goals

ERICSSON ONAP ONE SUMMIT

Goals of Unit Testing

- ✓**Validation**: Proves what the code does
- ✓**Assurance**: Prevent regression
- ✓**Design**: TDD / BDD
- ✓**Documentation**: Describes the behavior (happy path, edge cases, error scenarios)

2

Misunderstandings

Common Misunderstandings

"Unit test is for Code Coverage"

"Unit test should test only one class"
Don't be dogmatic, be pragmatic!

"Mocking is difficult"
No, not since Spock!

"Unit test are redundant I have good integration test"
Please read up on the testing pyramid!

3 / 15

3

Advantages of Spock and Groovy

LANGUAGE STRUCTURE AND DESCRIPTIONS PERFECTLY SUITED FOR HUMAN READABLE TEST

DATA DRIVEN MADE EASY USING PIPES OR WHERE: BLOCK

GROOVY MAGIC:
-ACCESS TO PRIVATE FIELDS
- GROOVY COLLECTIONS & MAPS
- AND MUCH MORE

SETTING UP STUBS, MOCKS AND BEHAVIOUR EASIER THEN EVER BEFORE

4 / 15

4

Method Names and Labels

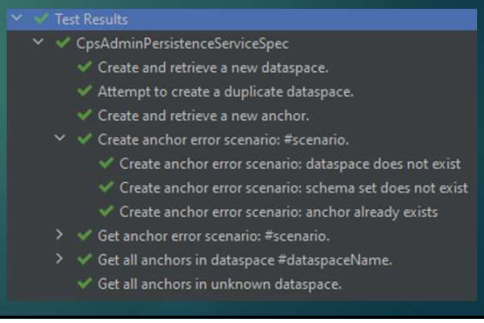
ERICSSON ONAP ONE SUMMIT

Traditional jUnit

```
@Test
public void testCreateAndRetrieveDataspace() {
  // When a new dataspace is created

  // Then it can be retrieved

}
```

Spock & Groovy

```
def 'Create and retrieve a new dataspace.'() {
   when: 'a new dataspace is created'

   then: 'it can be retrieved'

}
```



5 / 15

5

Blocks (incl. data-driven)

ERICSSON ONAP ONE SUMMIT

Blocks (labels)

given: / setup:

when:

then:

expect:

cleanup:

where:

Extend 'labels' with descriptions

Data Driven

```
when: '#x is raised to the power #y'
    def result = Math.pow(x,y)

then: 'the result is #expectedResult'
    assert result == expectedResult

where: 'the following values are used'
    x | y  || expectedResult
    3 | 2  || 9
    2 | 3  || 8
    3 | 0  || 1
    4 | -1 || 0.25
```

6 / 15

6

Access to private members

ERICSSON ONAP ONE SUMMIT

Traditional jUnit

```
AdminService {
    public void setCpsCacheForTest (CpsCache cpsCache) {
        this.cpsCache = cpsCache
    }
}

AdminServiceTest {
    objectUnderTest.setCpsPersistenceService(Mock(CpsCache))
}
```

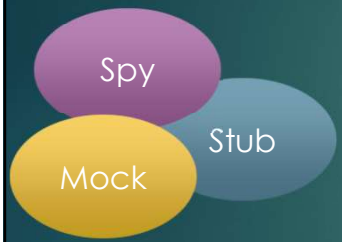
Spock & Groovy

```
AdminServiceSpec {
        setup: { objectUnderTest.cpsCache = Mock(CpsCache) }
}
```


7

Easy Mocking & Invocation Checks

ERICSSON ONAP ONE SUMMIT



Defining behavior (strict)

```
def mockCpsPersistenceService = Mock(CpsPersistenceService)
def anchors = [new Anchor('a1'), new Anchor('a2') ]
mockCpsPersistenceService.getAnchors('someDataspace') >> anchors
```

Asserting invocations (strict)

```
1 * mockCpsPersistenceService.createAnchor('someDataspace',
                                            'someSchemaSet',
                                            'someAnchorName')
```

Asserting invocations (relaxed)

```
0 * mockCpsPersistenceService.createAnchor(*_)
```

Define and Asserting in one

```
1 * mockCpsPersistenceService.getAnchors(_) >> anchors
```


8

If We Had More Time…

Advanced Groovy and Spock features
- old()
- with()
- GString (Groovy Strings)

Combining with other frameworks
- @SpringBootTest (injection, config)
- @WebMvcTest (REST, contract test)
- @Testcontainers (docker images)
  - PostgreSQLContainer

9 / 15

9

Shameless Plug

CPS: Configuration and Persistence Service

CPS

- Developers Landing page
  - https://wiki.onap.org/display/DW/Configuration+Persistence+Service+Developer%27s+Landing+Page
- Repos
  - https://gerrit.onap.org/r/admin/repos/cps
  - https://gerrit.onap.org/r/admin/repos/cps/ncmp-dmi-plugin

10 / 15

10

Tutorials

ERICSSON ONAP ONE SUMMIT

1. Basic Spock Specification
2. Data Driven Specifications
3. Expecting Exceptions
4. Mocking & Interaction Checking
5. Groovy Magic
6. Demo Test Container (CPS)

https://github.com/emaclee/one-summit-groovy-2022

11 / 15

11

Results

ONE SUMMIT

CPS

- All team's projects >97% coverage (SonarQube definition)
- OpenSSF Gold-Standard for Coverage
- High confidence level
- Minimal Integration Tests

SonarQube Statistics

Overall Code	
Coverage	97.1%
Lines to Cover	3,257
Uncovered Lines	63
Line Coverage	98.1%
Conditions to Cover	703
Uncovered Conditions	51
Condition Coverage	92.7%

Tips:

- Make Unit Test are part of **Definition-of-Done**
- Unit test coverage threshold of build **gate** (on new code)
- Include test in **review** process

12 / 15

12

Reference Material

- https://www.baeldung.com/groovy-spock
- http://spockframework.org/spock/docs
- https://github.com/spockframework/spock-example
- http://groovy-lang.org/documentation.html
- http://spockframework.org/spock/docs/2.3/data_driven_testing.html
- https://www.jetbrains.com/help/idea/getting-started-with-groovy.html#create_groovy_project

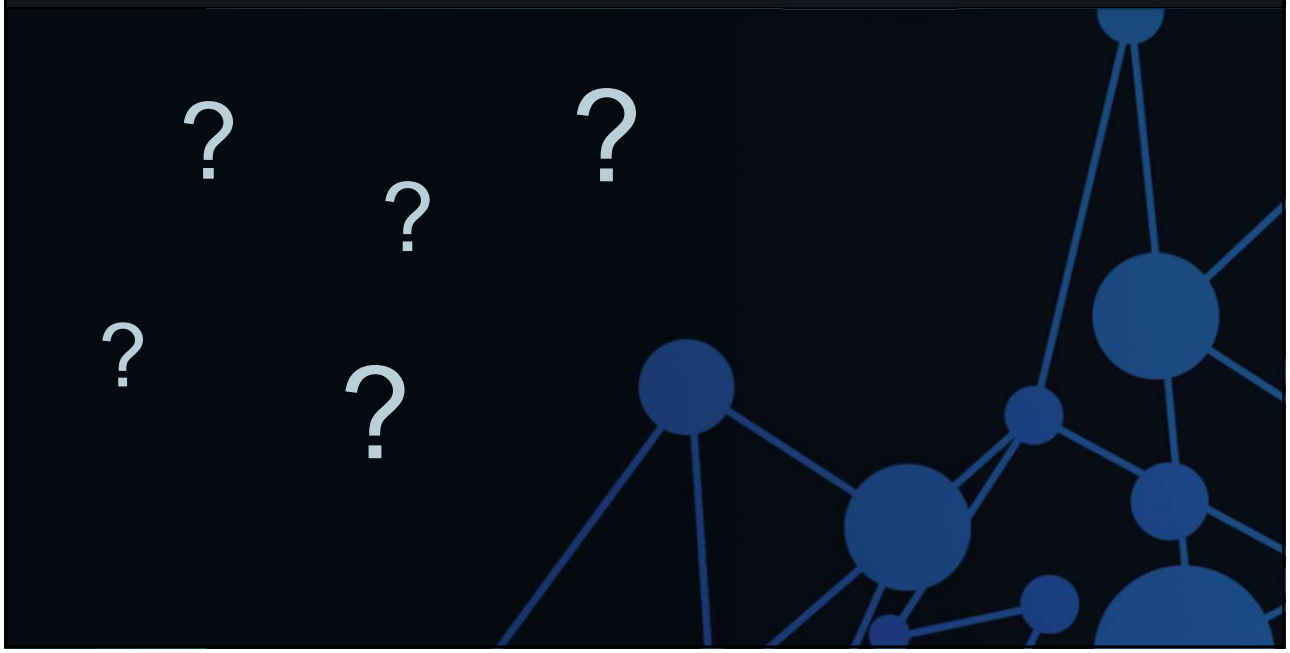
From ONAP CPS Project:

- https://gerrit.onap.org/r/admin/repos/cps
  - CpsDataPersistenceServiceIntegrationSpec.groovy

13 / 15

13

Questions & Answers



14

Thanks

thank you!