Het overbruggen van de communicatiekloof

met Behaviour Driven Development/Testing

oftewel:

"Bridging the communication gap with Behaviour Driven Development/Testing"

Roy de Kleijn

Who am I ?

- Technisch Test Specialist
- Trainer test automation
- Involved in R&D activities
- Blogs:
  - http://selenium.polteq.com
  - http://www.rdekleijn.nl

Agenda

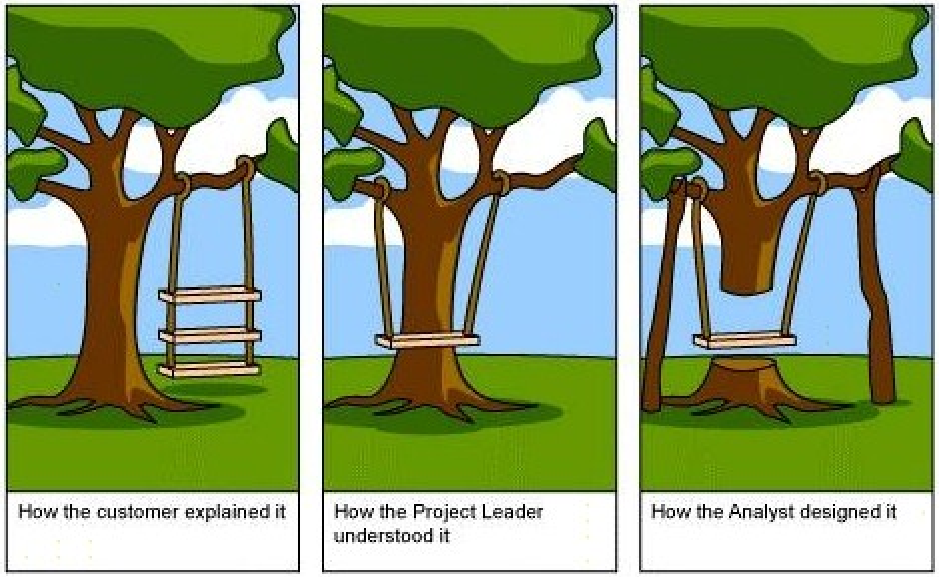
- **Why Behaviour Driven Development ?**
- What is Behaviour Driven Development ?
- Implementing Behaviour Driven Development
- Case Study

Problem

WASTE !

Problem

- Time wasted
- Lack of communication
- Every department uses another language
- Out-dated documentation
- More…


How the customer explained it
How the Project Leader understood it
How the Analyst designed it

Problem

- Developers don’t know where to start
- Unclarity on when it’s <u>done</u>
- Untraceable test scripts
- Unshareable test scripts

Agenda

- Why Behaviour Driven Development ?
- **What is Behaviour Driven Development ?**
- Implementing Behaviour Driven Development
- Case Study

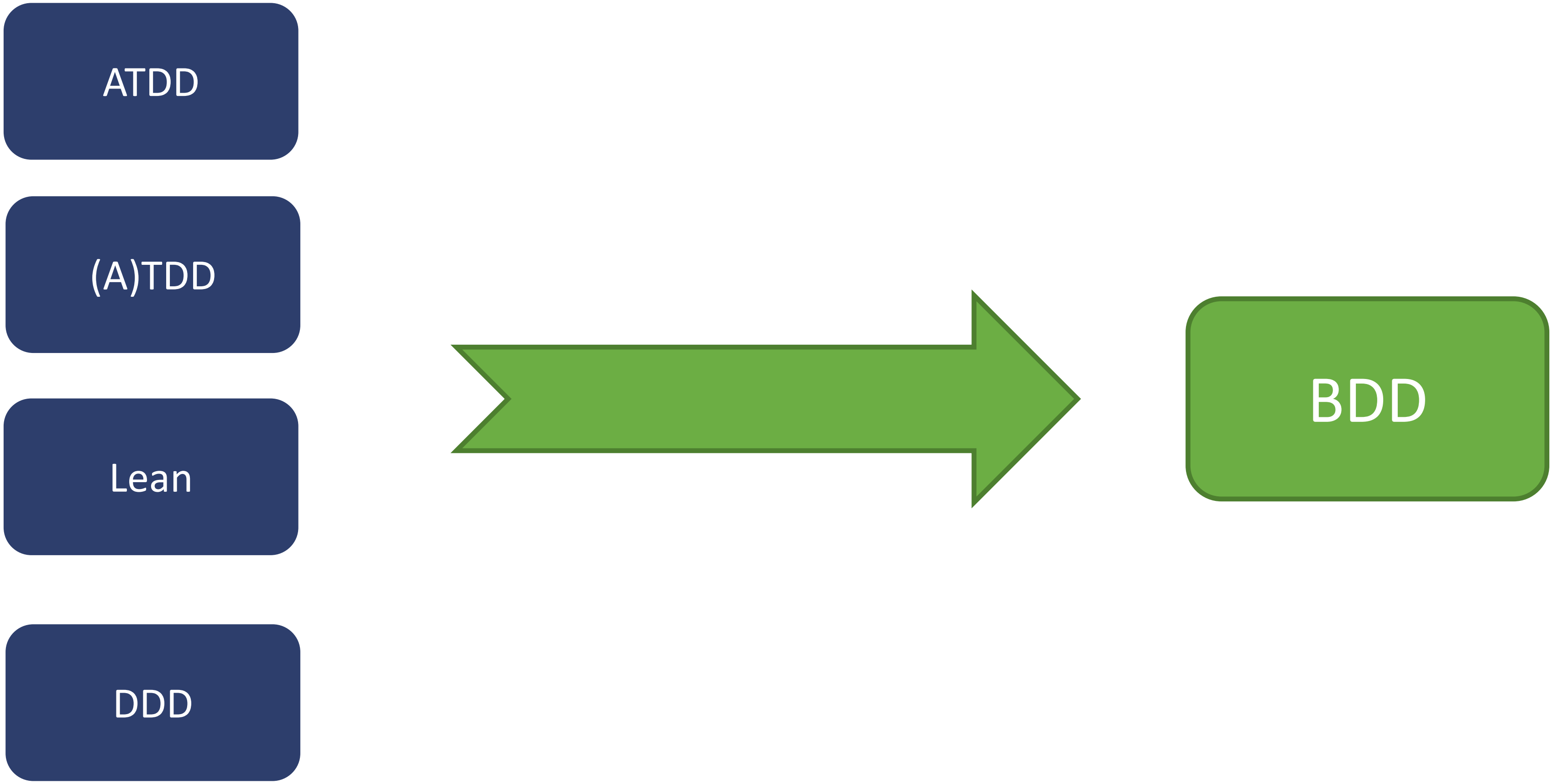
BDD != tooling

Definition

Dan North (founder):

> “BDD is a second-generation, outside–in, pull-based, multiple-stakeholder, multiple-scale, high-automation, agile methodology. It describes a cycle of interactions with well-defined outputs, resulting in the delivery of working, tested software that matters.”

Second-generation



ATDD

(A)TDD

Lean

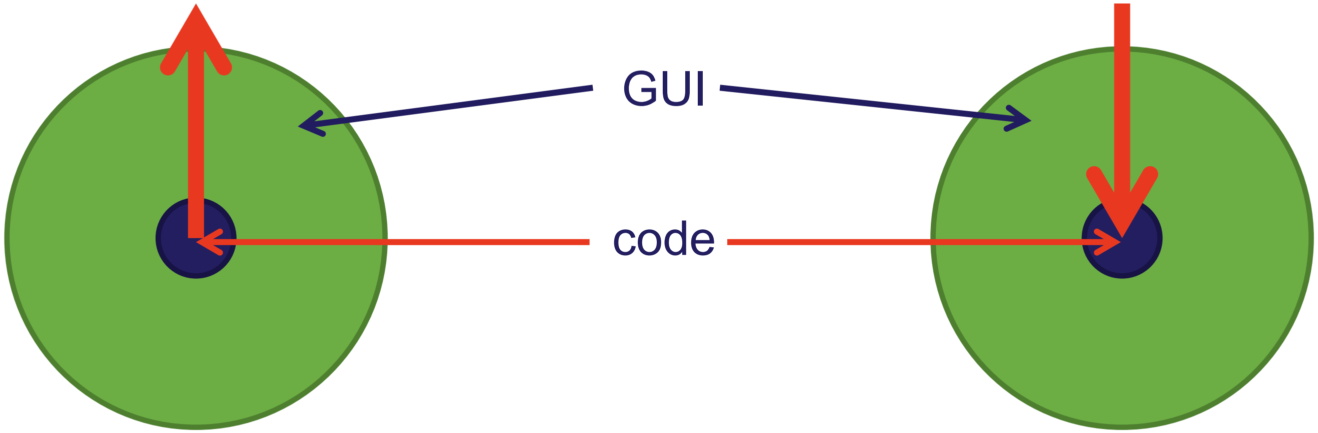
DDD

BDD

Outside-in vs. Inside-out

- Inside-out (unit testing)
  - Based on technical requirements
  - Small testable units
  - Tightly coupled
  - Fast

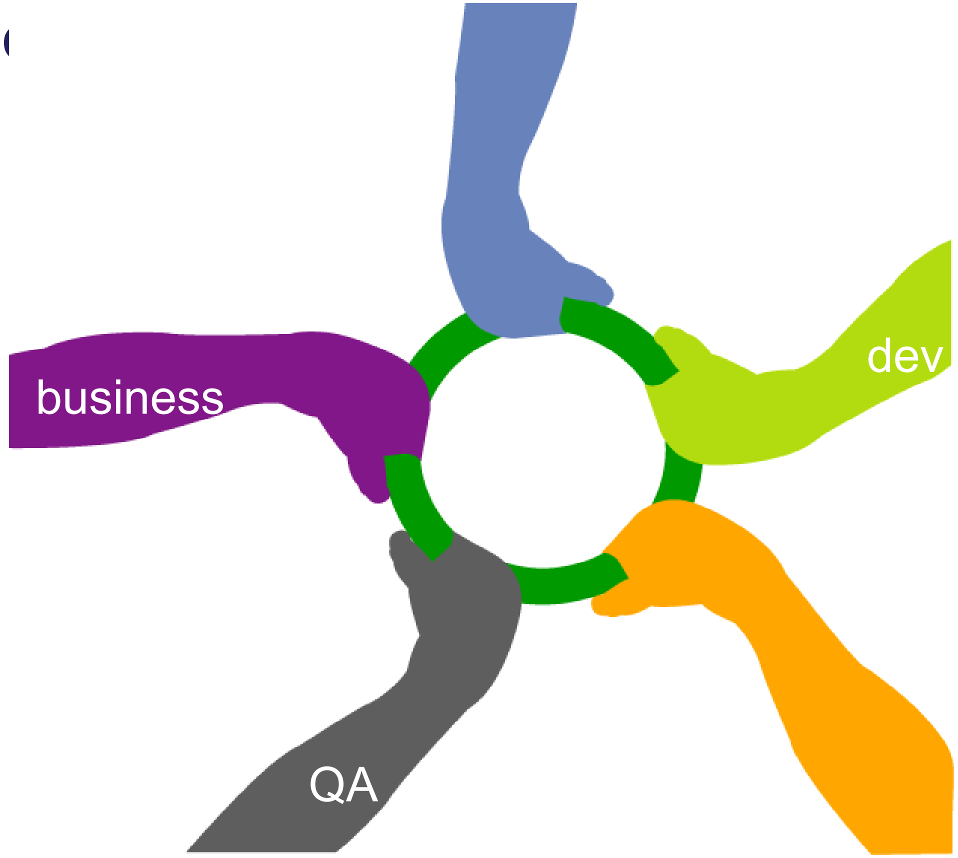
- Outside-in (acceptance testing)
  - Based on business requirements
  - End-user behaviour
  - Loosely coupled
  - Slow



"building it right / building the right product"

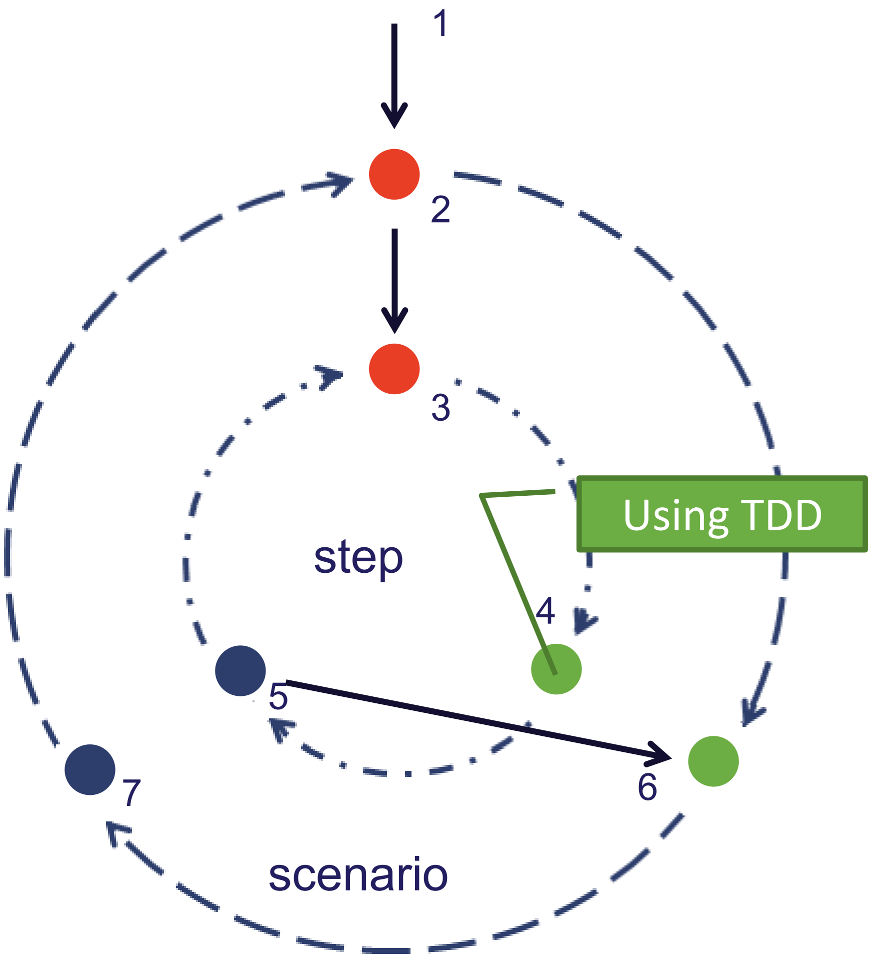
Other buzzwords

- Pull-based
  - Produce only that which is ready to be used
- Multiple-stakeholder
  - Everybody should be inv
- Multiple-scale
  - Enterprise
  - Consumer
  - Legacy
- High-automation
  - Specific syntax



BDD process

Write story with related scenarios



1. For each scenario
2. Run the scenario – it fails
3. Define the first step – it fails
4. Implement application code to let the step pass
5. Refactor the code and repeat step 4 & 5 for each step, until 6
6. The scenario passes
7. Refactor application code

Working software product/feature

Agenda

- Why Behaviour Driven Development ?
- What is Behaviour Driven Development ?
- **Implementing Behaviour Driven Development**
- Case Study

How to get there…

- Organize a meeting/workshop
  - Think of a proper name

- Timing
  - Before planning meeting

Collaboration between…

- All <u>relevant</u> stakeholders
  - Business
    - Ability to create common understanding
  - Business analyst
    - Projectteam is ‘forced’ to use these documentation
  - Developer
    - They get clear examples
    - They know where they are building towards
  - Tester
    - They get specifications with clear examples
    - They get small testable chunks of work
  - And so on…
    - Legal, marketing, …

Stories

- Describes business value
  - Not being executed
  - Documentation purpose

Syntax:

```
Narrative:
In order to <business value>
as a <role>,
I want <goal/desire>
```

Scenarios

- Describes acceptance criteria
- Act as automated tests

Syntax:

```
Title of scenario:
Given <precondition>
When <action>
And <follow-up action>
Then <postcondition>
```

Examples

- Repeat the same scenario with different data

Syntax:

```
Examples:
|username|password|
|Bob|qwerty|
|Alice|ytrewq|
```

Example 1/2

Calculator functionality

Narrative:

In order to make proper calculations

as a mathematician,

I want to perform mathematical operations

Example 2/2

```
Scenario: calculating with two number
Given I open the calculator
When I enter <input1> into the calculator
And I press <button>
And I enter <input2> into the calculator
And I press the result button
Then the result <result> is displayed

Examples:
|input1|button|input2|result|
|10|+|20|30|
|15|*|2|30|
|30|/|2|15|
```

Tools

- .Net – Nbehave, SpecFlow
- Java – Jbehave, easyb
- C – Cspec
- PHP – Behat, PHPSpec
- Ruby – Rspec, Cucumber
- And many more …

5 steps implementation

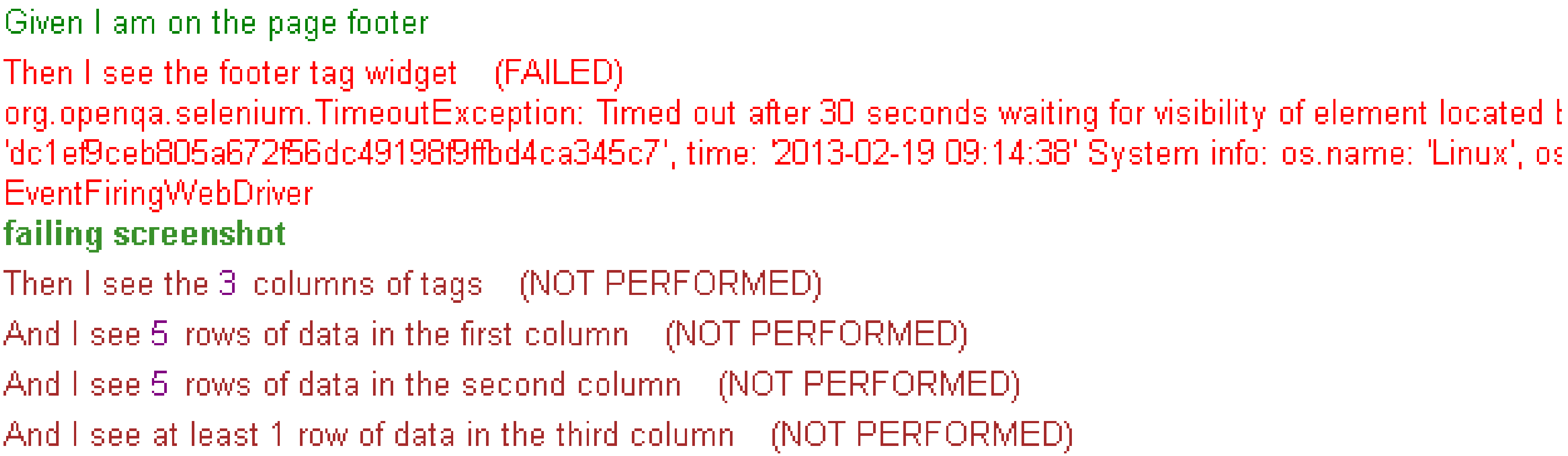
1. Write a story
   - In plain text
2. Map steps to code
3. Configure Framework
4. Run Stories
   - Manually, continuous integration, periodic job
5. View reports
   - Living documentation

Configuration

- How to load stories
- Class-file location
- Error strategy
- Reporting format

Reporting

- Green / red notation
- Different formats (HTML, TXT, XML)
  - Automatically generated
  - Customizable
- Living documentation when done right



```
Given I am on the page footer
Then I see the footer tag widget    (FAILED)
org.openqa.selenium.TimeoutException: Timed out after 30 seconds waiting for visibility of element located b
'dc1ef9ceb805a672f56dc49198f9ffbd4ca345c7', time: '2013-02-19 09:14:38' System info: os.name: 'Linux', os
EventFiringWebDriver
failing screenshot
Then I see the 3  columns of tags    (NOT PERFORMED)
And I see 5  rows of data in the first column    (NOT PERFORMED)
And I see 5  rows of data in the second column    (NOT PERFORMED)
And I see at least 1 row of data in the third column    (NOT PERFORMED)
```

Pros / cons

- Pros
  - JED – Just Enough Documentation
  - Know when it’s done
  - Ubiquitous language
  - Living documentation
  - Easy to improve code quality

- Cons
  - Extra mapping layer
  - Too structured

Agenda

- Why Behaviour Driven Development ?
- What is Behaviour Driven Development ?
- Implementing Behaviour Driven Development
- **Case Study**

Case Study – Spil Games

- Company founded in 2001
- 350+ employees world wide
- 45 portals in 19 languages
  - Casual games
  - Social games
  - Real time multiplayer games
  - Mobile games

>13.000.000.000

Minutes of Play/Month

11.000.000.000

Gameplays in 2011

200.000.000

Monthly Active Users (MAUs)

Writing Stories

- Session
  - Developers / QA / product owner

Example

```
Login

Narrative:
In order to store generated content
As a user
I want to log in on the portal

Scenario: User can successfully sign in with valid credentials
Given I am on the homepage
When I login with username <username> and password <password>
Then I am logged in

Examples:
|username|password|
|supertester3|supertester3|
```

Mapping steps to code

Jbehave provides a skeleton at the first run

```
@Given("I am on the homepage")
@Pending
public void openHomePage() {

}
```

Example

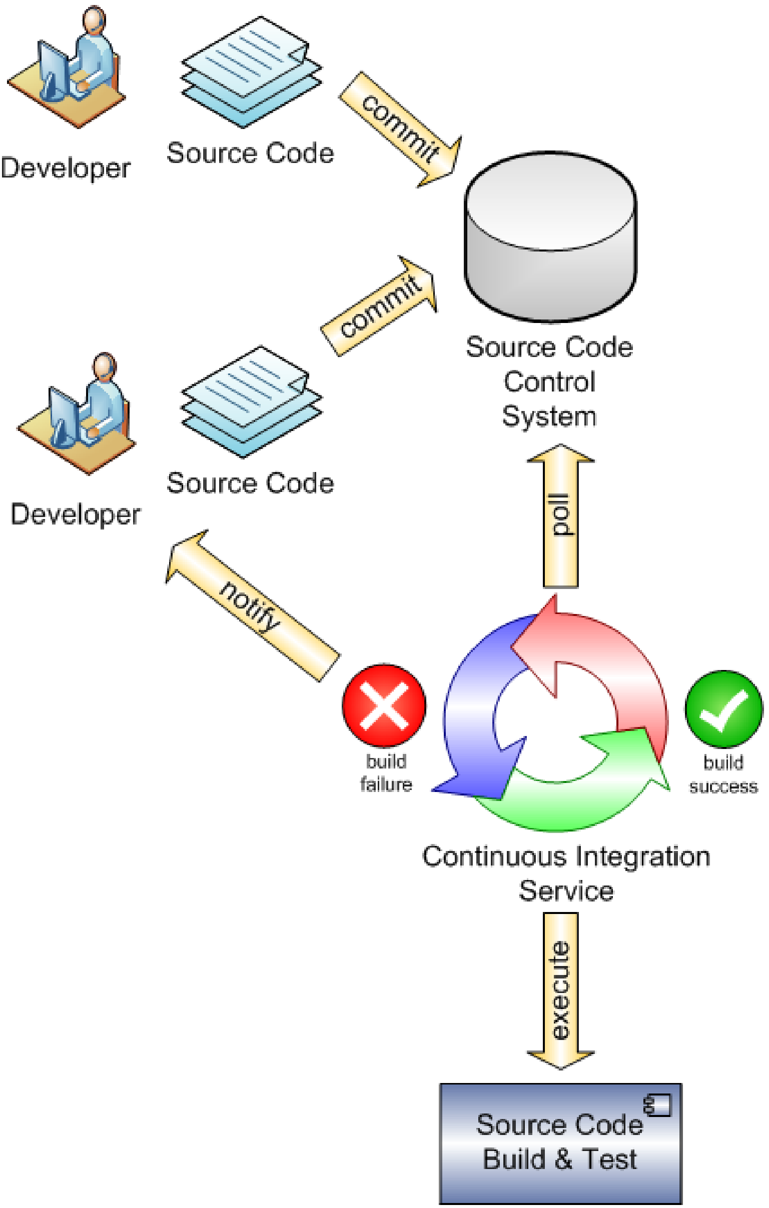
```
@Given("I am on the homepage")
public void openHomePage() {
  new HomePage(getProvider()).init();
}

@When("I login with username <username> and password <password>")
public void loginAs(@Named("username") String username,
      @Named("password") String password) {
  new LoginComponent(getProvider()).enterUsernamePassword(username,
        password).submitLoginForm();
}

@Then("I am logged in")
public void thenIAmLoggedIn() {
  assertThat(new HeaderComponent(getProvider())
  .isLoggedInContainerDisplayed(), is(equalTo(true)));
}
```

Run Stories

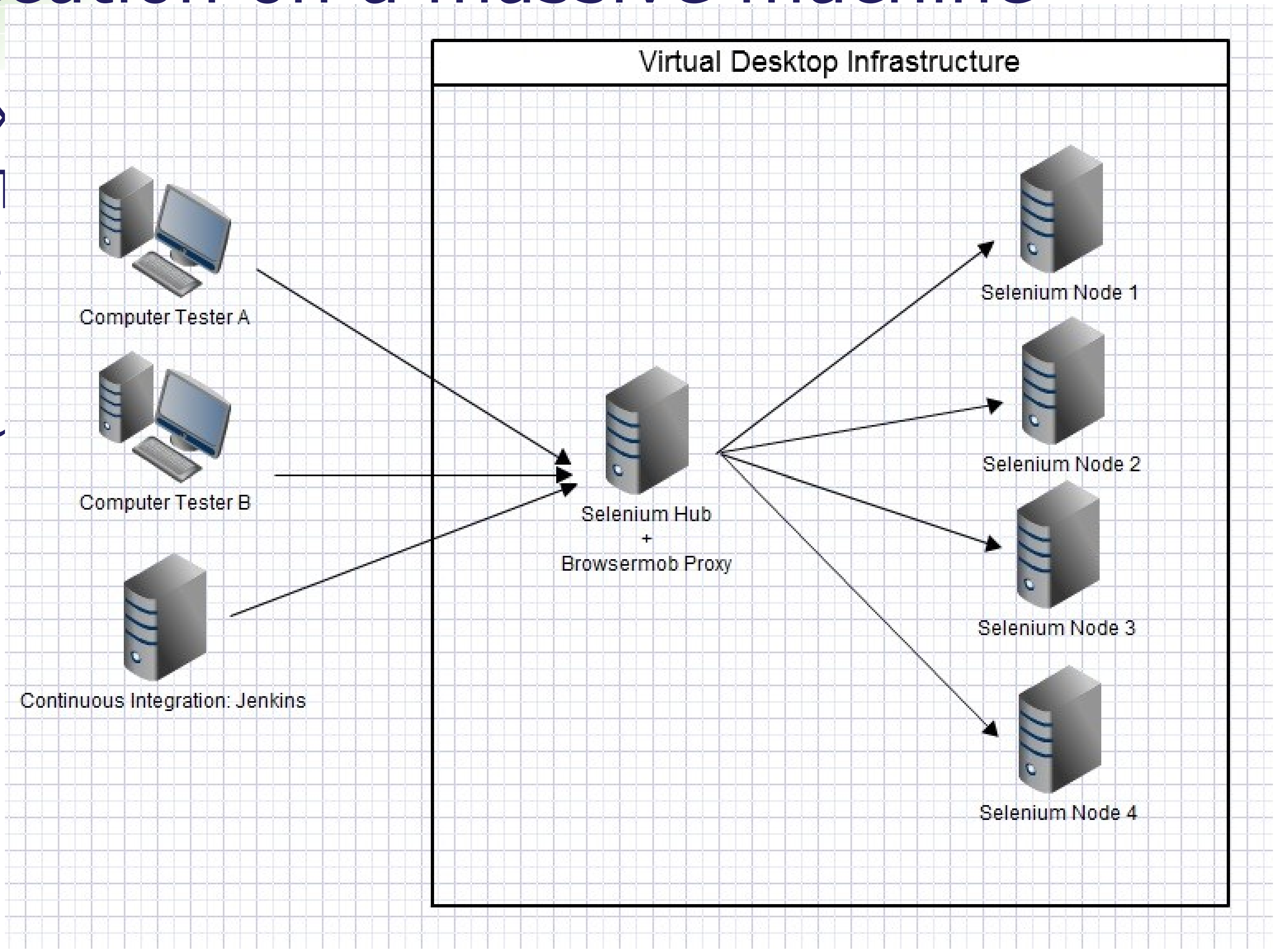
- On every release branch comn
  - Continuous integration
  - Manually



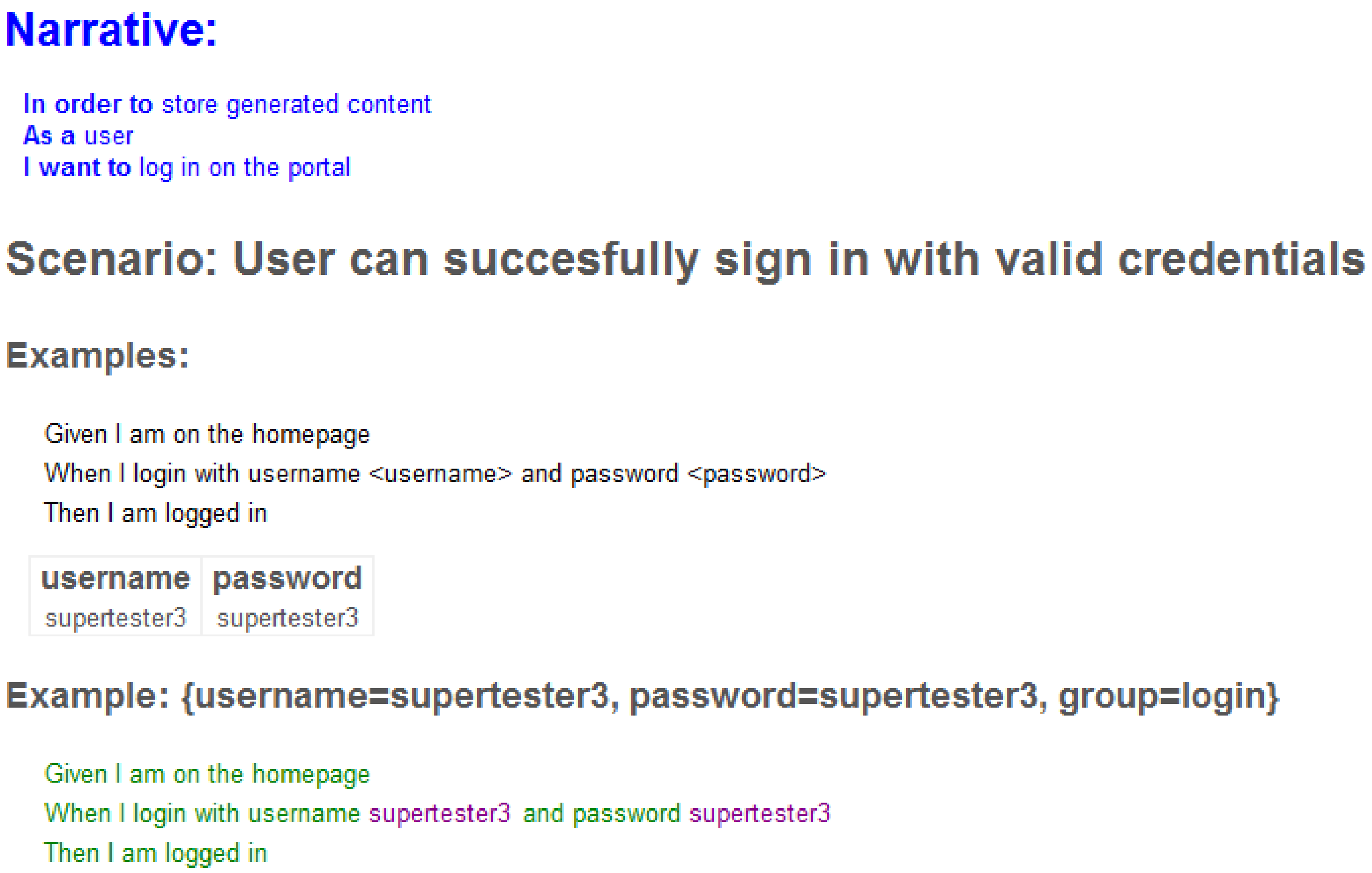
Execution on a massive machine

- 2 ›
- 1 T
- 13

- Mu



View Reports



Narrative:

In order to store generated content
As a user
I want to log in on the portal

Scenario: User can succesfully sign in with valid credentials

Examples:

Given I am on the homepage
When I login with username <username> and password <password>
Then I am logged in

username	password
supertester3	supertester3

Example: {username=supertester3, password=supertester3, group=login}

Given I am on the homepage
When I login with username supertester3 and password supertester3
Then I am logged in

Conclusion

- Better collaboration
  - It’s a conversation captured
- Executable specification and examples
  - Avoid misinterpretations
  - Reliable documentation
- Ultimate regression test suite
- Living documentation
  - Provides actual state of application

Questions

?

Thank you!

Email: roy.dekleijn@polteq.com

Twitter: https://twitter.com/TheWebTester

LinkedIn: http://www.linkedin.com/in/roydekleijn

About.me: http://about.me/roydekleijn

Website: http://selenium.polteq.com

Github: https://github.com/roydekleijn