EN
English

Software Architecture

Acceptance tests

Escuela de Ingeniería Informática de Oviedo

2019-20

Jose Emilio Labra Gayo
Pablo González
Irene Cid
Hugo Lebredo

Acceptance tests and BDD

- Tests that can be run in front of the client
  - If the tests pass, the product is accepted
- Behaviour-Driven Development (BDD)
  - Variant of TDD
    - Acceptance test driven development
  - Behaviour = User Stories
  - Also known as: *Specification by example*
  - Goal: Executable specifications
- Some tools:
  - cucumber, jBehave, concordion

BDD - User Stories

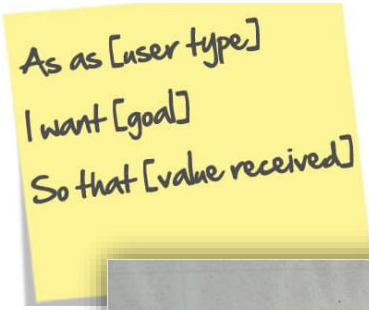
- Simple
- Readable by domain experts (business people)
- Approved by domain experts
- More advisable characteristics:
  - Independents (with no strong relationships)
  - Negotiable (with no specific details)
  - Valuable for the customer
  - Estimatable (to add them to Sprints)
  - Small (or consider division)
  - Testable (automatic tests)

User story structure

Feature: *Title (one line describing the story)*

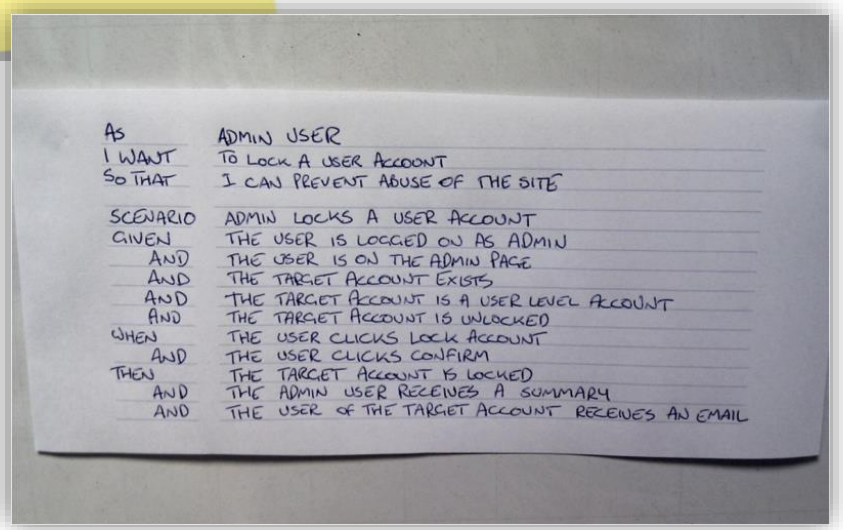
The following structure is recommended:

As a [role]
I want [feature]
So that [benefit]



Scenarios

Given [Context]
And [Some more context]
when [Event]
then [Outcome]
And [Another outcome]

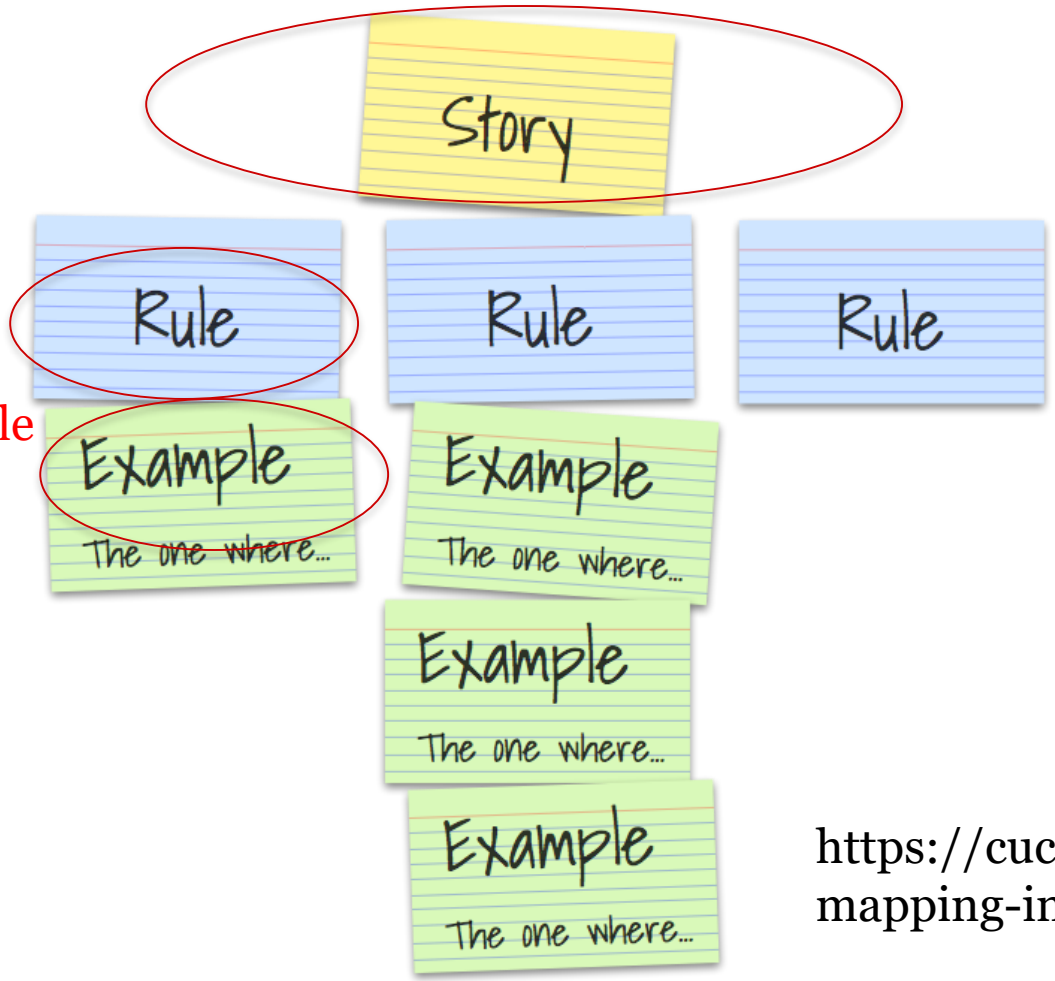


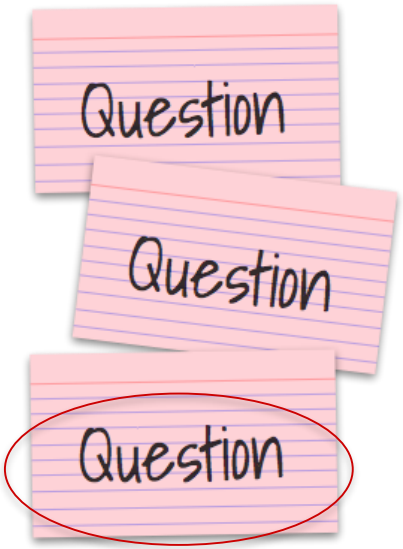
BDD – Example Mapping

Feature

Scenario

Example





Unsolved discussions

https://cucumber.io/blog/example-mapping-introduction/

BDD using Cucumber

- Cucumber = developed in Ruby (2008)
  - RSpec (Ruby), jbehave (Java)
  - Based on Gherkin
    - internal language to define user stories
  - Web: `http://cukes.info`
  - Support for multiple languages
- Java: cucumber-jvm
  - `https://github.com/cucumber/cucumber-jvm`
- Eclipse support:
  - `http://cucumber.github.io/cucumber-eclipse/`
  - `Ansi colors:`
    - http://www.mihai-nita.net/eclipse

BDD using cucumber

- Features define some functionality
  - Gherkin language
    https://cucumber.io/docs/gherkin/
  Can be used in several languages
- User stories are linked to step definitions
  - Step definitions can be run to validate user stories

BDD using cucumber

Feature: Describes a system feature

- A feature can have several scenarios
- Scenario:
  - How must the system behave in some context
    - *Given*: Prepares scenario
    - *When*: Interact with the system
    - *Then*: Checks the state
    - *Examples*: Specific data

BDD

- Step by step guide to a user story
  - Install Cucumber
  - Write a first scenario in Gherkin
  - Write steps definitions in a chosen programming language
  - Run cucumber

BDD with cucumber

- Depends on programming language/environment
  - Java/Javascript/Python/...
  - Installation: https://cucumber.io/
- React: https://github.com/Arquisoft/viade_0

  jest-cucumber: Module to define user stories in Gherkin
    - And convert them to executable tests by Jest

  ```
  $ npm install --save-dev puppeteer jest-cucumber
  ```

  jest-puppeteer. Module to run the tests in a browser

  It could be configured to use Selenium.

  ```
  $ npm install --save-dev puppeteer jest-puppeteer
  ```

BDD

- User Story example using Node.js

Feature: Registering a new user

Feature

Scenario: The user is not registered in the site
Given An unregistered user
When I fill the data in the form and press submit
Then A welcome message should be shown in the screen

Scenario

Scenario: The user is already registered in the site
Given An already registered user
When I fill the data in the form and press submit
Then An error message should be shown in the screen

Scenario

feature/register-form.feature

BDD

feature/step-definition/register-form-steps.js

```
const {defineFeature, loadFeature}=require('jest-cucumber');
const feature = loadFeature('./feature/register-form.feature');

defineFeature(feature, test => {

  beforeEach(async () => {
    await page.goto('http://localhost:3000')
  })

  test('The user is not registered in the site', ({given,when,then}) => {

    let email;

    given('An unregistered user', () => {
      email = "newuser@test.com"
    });

    when('I fill the data in the form and press submit', async () => {
      await expect(page).toFillForm('form[name="register"]', {
        email: email,
        remail: email,
      })
      await expect(page).toClick('button', { text: 'Submit' })
    });

    then('A welcome message should be shown in the screen', async () => {
      await expect(page).toMatchElement('span', { text: 'The user '+email+' has been registered!' })
    });
  });
```

RegisterForm

```
registerUser(email){
    if (email==='alreadyregistered@test.com') //This user is already registered
        return false
    else
        return true //Everything went smooth
}


submitForm(e)
{
    e.preventDefault()
    //Add the user to the database
    if (this.registerUser(this.state.email))
        this.setState({welcomemessage:'The user '+this.state.email+' has been registered!'})
    else
        this.setState({welcomemessage:'ERROR: The user '+this.state.email+' is already registered!'})
}

render()
{
    return(
        <Form name="register" onSubmit={this.submitForm.bind(this)}>
            <Form.Control type="text" name="email"
                          placeholder="Input email"
                          aria-label="email-input"
                          onChange={this.changeEmail.bind(this)} value={this.state.email}/>
            <Form.Control type="text"
                          name="remail"
                          placeholder="Input remail"
                          aria-label="remail-input"
                          onChange={this.changeRemail.bind(this)} value={this.state.remail}/>
            <Button variant="primary" type="submit" disabled={!this.state.enabled}>Submit</Button>
            <div>
                <span hidden={this.state.welcomemessage===''}>{this.state.welcomemessage}</span>
            </div>
```

BDD

- Configuration jest-config.js

```
module.exports = {
    project: 'bigfoot',
    verbose: true,
    preset: 'jest-puppeteer',
    testRegex: 'feature/.*\\.js$',
}
```

- Configuration jest-puppeteer.config.js

```
module.exports = {
    server: {
      command: `npm start`,
      port: 3000,
      launchTimeout: 100000,
      debug: true,
    },
}
```

BDD

- Configuration package.json:

```
"scripts": {
  "start": "react-scripts start ",
  "build": "react-scripts build",
  "test": "react-scripts test --coverage",
  Run Script
  "eject": "react-scripts eject",
  "test:e2e": "jest -c  jest-config.js"
},
```

- Configuration package.json:
  - **npm run test:e2e**

BDD

- Result

```
 PASS  feature/step-definition/register-form-steps.js (7.515s)
  Registering a new user
    ✓ The user is not registered in the site (5146ms)
    ✓ The user is already registered in the site (523ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        7.919s, estimated 11s
Ran all test suites.
```

Other example cucumber + selenium + java (spring boot) from previous years:

https://github.com/arquisoft/votingSystem0

Browser-based tests

- Browser automation
  - https://cucumber.io/docs/reference/browser-automation
- Several systems
  - Selenium WebDriver - http://docs.seleniumhq.org/
  - Capybara - http://teamcapybara.github.io/capybara/
  - Watir - https://watir.com/
  - Serenity - http://serenity-bdd.info

Selenium

- Selenium IDE: Allows to record actions
  - Firefox and Chrome plugins
- Generates code to execute those actions
- Travis configuration
  - https://lkrnac.net/blog/2016/01/run-selenium-tests-on-travisci/

Another example with cucumber and selenium at:

https://github.com/arquisoft/votingSystem0

Bibliography and links

- User Story Mapping by Jeff Patton
  - **User Story Mapping: Discover the Whole Story, Build the Right Product, 1st Edition**
    https://www.amazon.com/User-Story-Mapping-Discover-Product/dp/1491904909
- Historias de Usuario
  - Scrum. Historias de Usuario (Fernando Llopis, Universidad de Alicante)
    https://fernandollopis.dlsi.ua.es/?p=39
  - User stories with Gherkin and Cucumber (Michael Williams)
    https://medium.com/@mvwi/story-writing-with-gherkin-and-cucumber-1878124c284c
  - BDDevelopment en Javascript con Cucumber (Ricardo Ahumada)
    https://www.bit.es/knowledge-center/behavior-driven-development-en-javascript-con-cucumber/
  - Cucumber 10 minutes tutorial (JS)
    → **https://docs.cucumber.io/guides/10-minute-tutorial/**
- Pruebas basadas en navegador
  - Automated UI Testing with Selenium and JavaScript
    https://itnext.io/automated-ui-testing-with-selenium-and-javascript-90bbe7ca13a3