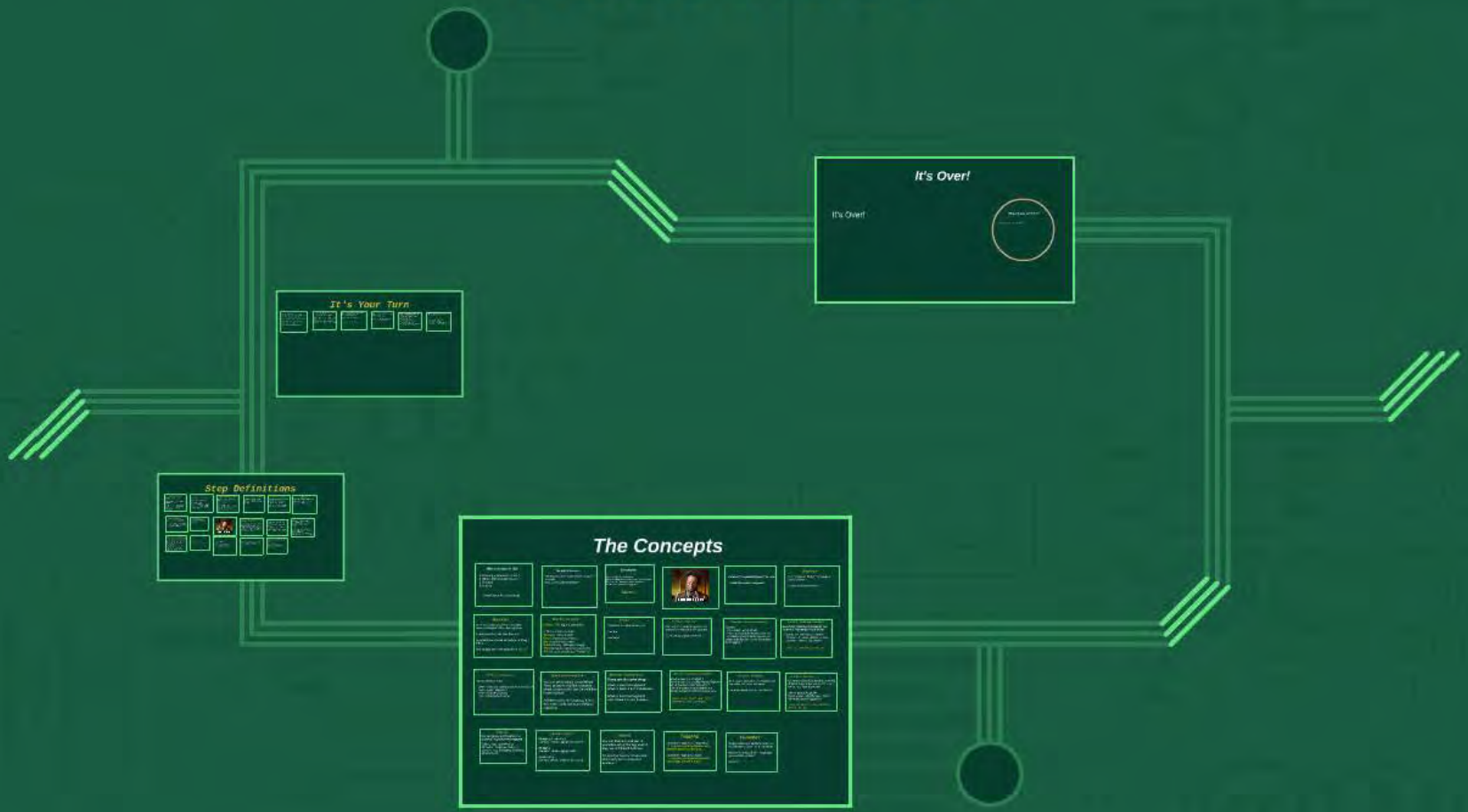
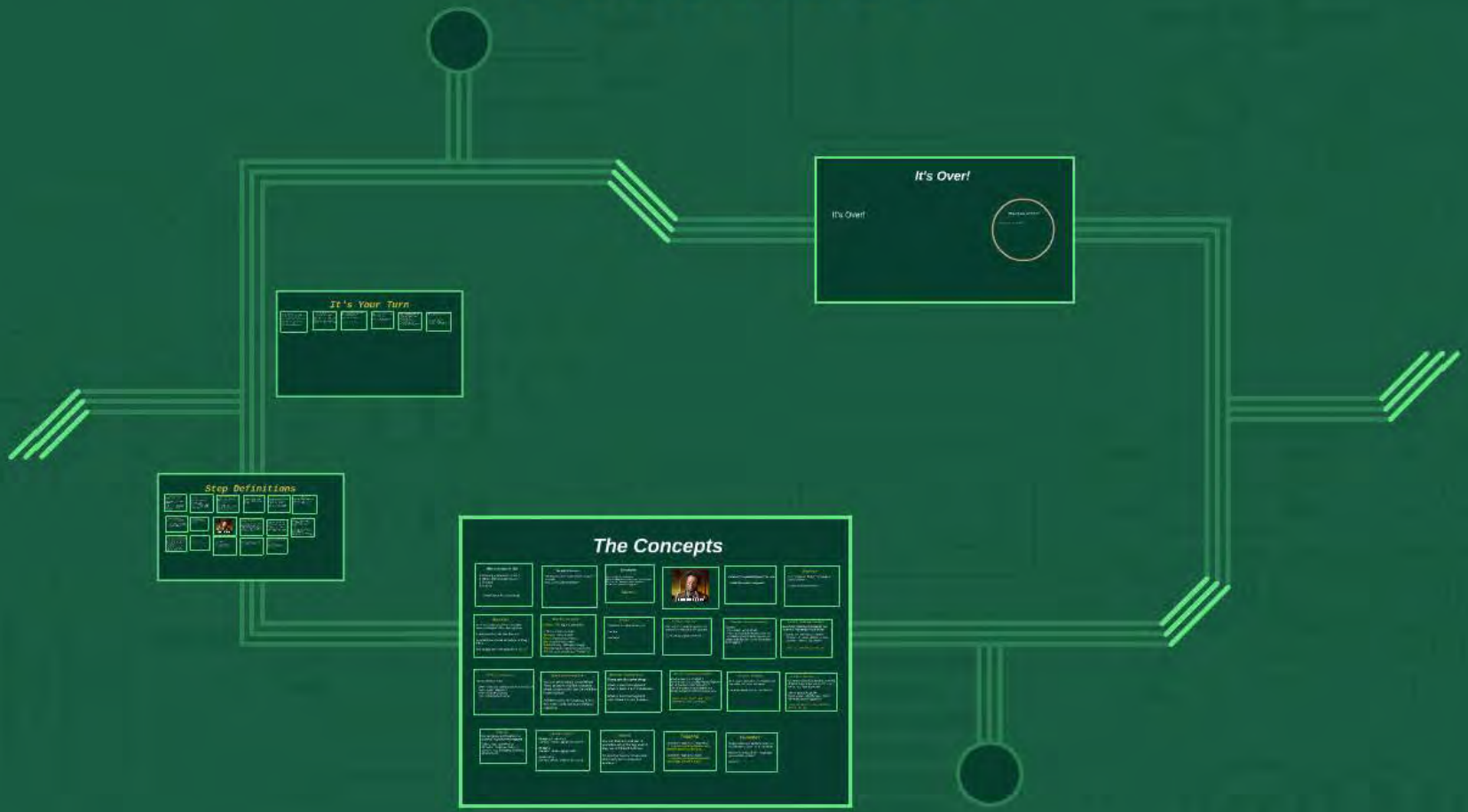
CS1699 - Lecture 12 - Automated Testing of BDD with Cucumber



Prezi

CS1699 - Lecture 12 - Automated Testing of BDD with Cucumber



Prezi

The Concepts

BDD is similar to TDD

1. Develop a scenario (= test)
2. Write code to make it pass
3. Refactor
4. Repeat

(Red-Green-Refactor loop)

The key difference...

The tests / scenarios are human-readable, not code.
How can we automate these?

Example

Given a user who is disabled
When an attempt is made to login with that user
Then an error message will be displayed
And the user will not be logged in

Ideas?

Convert "Templated English" to code

... with the power of regexes!

Gherkin

This "templated English" language is called Gherkin.*

* Does anyone know why?

Gherkin

Think of Gherkin as a very formalized version of English with a few keywords.

It uses spaces or tabs, not braces { }.

Comments are allowed anywhere, and begin with a #.

Line endings terminate statements (no ;'s).

Gherkin Example

Feature: Test login functionality

This is a test scenario
Scenario: Correct login
Given a correct username
And a correct password
When the user attempts to log in
Then the login should be successful
And the user should see "Welcome!"

Gherkin

Two Parts to a Gherkin program

Feature

Scenarios

Gherkin Feature

Start with the keyword feature, then freetext until you get to the scenario.

Think of it as a giant comment.

Gherkin Feature Example

Feature:
As a system administrator
I want to be able to disable accounts
In order to prevent users who are no longer authorized to access the system from logging in

Gherkin Feature Example

Reminder: Nothing is Stopping You from Not Following A User Story!

Feature: Chicken chicken chicken Chicken. Chicken; chicken chicken. Chicken > chicken && chicken.

This is perfectly valid.

Gherkin Scenarios

Given / When / Then

Given - State you need to set up or things that have already happened
When - Execution Steps
Then - Expected outcome

Gherkin Scenarios

You can either repeat Given/When/Then, or use And or But keywords which act just as the last Given/When/Keyword used.

And/But makes the language flow a little more easily, but same thing as repeating.

Gherkin Scenarios

These are the same thing:

Given a user has logged in
Given 1 book is in the database

Given a user has logged in
And 1 book is in the database

Gherkin Scenario Example

Given a user Joe is logged in
And the user Joe is using Internet Explorer
When the user clicks "Who am I?"
Then the system should display Joe
But the system should not display Jane

Note that "And" and "But" keywords are synonyms.

Gherkin Example

A file usually describes one feature, and has some number of scenarios.

Features usually map to user stories.

Gherkin Gotchas!

Cucumber does not actually do anything different "behind the scenes" for Given, When, and Then keywords!

Then a user is logged in
Given a user clicks the logout button
When the user is logged out

This is valid, but please don't do it.

Tagging

You can group test features or scenarios together with tagging.

Just put tags, specified by @<name>, before a feature or scenario, e.g. @logging, @queues, @quickcheck.

Tagging Example

@logging @networking
Scenario: Check logging over network

@logging
Scenario: Check logging locally

@networking
Scenario: Works if network is disabled

Tagging

You can then run a subset of scenarios set by the tag, even if they are in different features.

Or you may have a "smoke test" which only runs a subset of features.

Tagging

cucumber --tags @foo --tags @bar
^- Runs any scenarios/features with EITHER tags @foo and @bar

cucumber --tags @foo, @bar
^- Runs any scenarios/features with BOTH tags @foo and @bar

Cucumber

Testers often tie in additional tools, such as Selenium or Jasmine, to Cucumber.

But how do these Gherkin-languages get executed as code?

Anyone?

Prezi

BDD is similar to TDD

1. Develop a scenario (= test)
2. Write code to make it pass
3. Refactor
4. Repeat

(Red-Green-Refactor loop)

Prezi

The key difference...

The tests / scenarios are human-readable, not code.
How can we automate these?

Prezi

Example

Given a user who is disabled
When an attempt is made to login with that user
Then an error message will be displayed
And the user will not be logged in

Ideas?

Prezi


REGULAR EXPRESSIONS
HD
ORY.COM
erator.net

Convert "Templated English" to code

... with the power of regexes!

Prezi

Gherkin

This "templated English" language is called Gherkin.*

* Does anyone know why?

Prezi

Gherkin

Think of Gherkin as a very formalized version of English with a few keywords.

It uses spaces or tabs, not braces { }.

Comments are allowed anywhere, and begin with a #.

Line endings terminate statements (no ;'s).

Prezi

Gherkin Example

```
Feature: Test login functionality

# This is a test scenario
Scenario: Correct login
Given a correct username
And a correct password
When the user attempts to log in
Then the login should be successful
And the user should see "Welcome!"
```

Prezi

Gherkin

Two Parts to a Gherkin program

Feature

Scenarios

Prezi

Gherkin Feature

Start with the keyword feature, then freetext until you get to the scenario.

Think of it as a giant comment.

Prezi

Gherkin Feature Example

Feature:

As a system administrator

I want to be able to disable accounts

In order to prevent users who are no longer authorized to access the system from logging in

Prezi

Gherkin Feature Example

Reminder: Nothing is Stopping You from Not Following A User Story!

```
Feature: Chicken chicken chicken
    Chicken.  Chicken; chicken chicken.
    Chicken > chicken && chicken.
```

This is perfectly valid.

Prezi

Gherkin Scenarios

Given / When / Then

Given - State you need to set up or things that have already happened
When - Execution Steps
Then - Expected outcome

Prezi

Gherkin Scenarios

You can either repeat Given/When/Then, or use And or But keywords which act just as the last Given/When/Keyword used.

And/But makes the language flow a little more easily, but same thing as repeating.

Prezi

Gherkin Scenarios

These are the same thing:

Given a user has logged in
Given 1 book is in the database

Given a user has logged in
And 1 book is in the database

Prezi

Gherkin Scenario Example

Given a user Joe is logged in
And the user Joe is using Internet Explorer
When the user clicks "Who am I?"
Then the system should display Joe
But the system should not display Jane

Note that "And" and "But" keywords are synonyms.

Prezi

Gherkin Example

A file usually describes one feature, and has some number of scenarios.

Features usually map to user stories.

Prezi

Gherkin Gotchas!

Cucumber does not actually do anything different "behind the scenes" for Given, When, and Then keywords!

Then a user is logged in
Given a user clicks the logout button
When the user is logged out

This is valid, but please don't do it.

Prezi

Tagging

You can group test features or scenarios together with tagging.

Just put tags, specified by @<name>, before a feature or scenario, e.g. @logging, @queues, @quickcheck.

Prezi

Tagging Example

@logging @networking
Scenario: Check logging over network

@logging
Scenario: Check logging locally

@networking
Scenario: Works if network is disabled

Prezi

Tagging

You can then run a subset of scenarios set by the tag, even if they are in different features.

Or you may have a "smoke test" which only runs a subset of features.

Prezi

Tagging

cucumber --tags @foo --tags @bar
^- *Runs any scenarios/features with EITHER tags @foo and @bar*

cucumber --tags @foo, @bar
^- *Runs any scenarios/features with BOTH tags @foo and @bar*

Prezi

Cucumber

Testers often tie in additional tools, such as Selenium or Jasmine, to Cucumber.

But how do these Gherkin-languages get executed as code?

Anyone?

Prezi

Step Definitions

Step Definitions

All of those Given, When , Then statements are called *steps*.

They are defined in code in a separate "step definitions" file.

Usually, there is a 1:1 correspondence between feature files and step definitions, plus additional ones for common steps.

Step Definitions Example

Let's say we have a feature file like so:

Feature: HelloWorld says Hello

Scenario: English hello
Given the language is set to English
When I run the helloworld program
Then the response is "Hello!"

Step Definitions File

Remember there are two parts of a Gherkin feature file, the *feature* and the *scenarios*.

Also remember that the feature part is just freetext description. There's no need to make steps for it.

Step Definitions Example

That leaves us with the scenarios. We will need to define the given, when, and then, in Java (or whatever language).

Step Definitions Example

Let's add the proper imports first.

```
import cucumber.annotation.en.Given;
import cucumber.annotation.en.Then;
import cucumber.annotation.en.When;

import static org.junit.Assert.assertEquals;
```

Whoa! JUnit! What are you doing here?

Sidenote

Remember I said BDD builds on TDD?

Cucumber builds on other testing software, e.g., JUnit or RSpec.

Step Definitions

```
public class HelloworldStepDefs {

   // Now let's define the actual steps
   // here.  Just remember that they're
   // in this step definition class.
}
```

```
private String _language;

@Given ("the language is set to (.*)$")
public void setLanguageTo(String lang) {
   _language = lang;
}
```

A Brief Refresher on Regexes (Regular Expressions)

```
(?:[a-z0-9!#$%&'*+/=?^_`{|}~-]+(?:\.[a-z0-9!#$%&'*+/=?^_`{|}~-]+)*|"(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21\x23-\x5b\x5d-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])*")@(?:(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?|\[(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?|[a-z0-9-]*[a-z0-9]:(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21-\x5a\x53-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])+)\])
```

Anyone know what this is?

A (or some other alphanumeric) = itself
.* = Any number of a character, or none
.+ = One or more of a character

Example:

ca*t -> matches ct, cat, or caaaat
ca+t -> matches cat or caaaat, not ct

[0-9] or d = matches any digit
^ = beginning of line (just like vi!)
$ = end of line (just like vi!)
() = how you enclose regexes

Example

(d+) matches 1 or 984, but not "foo"
(bird$) matches "I saw a bird" but not "I saw a bird flying by"
(^Ice) matches "Ice ice baby" but not "I gave my baby some ice for his teething."

What You'll Probably Use

(.*) = match any set of characters
(d+) = match any set of digits
(an?) = matches a or an (for English-ism)
(?:got a | got some) = matches either "got a" or "got some"
^ and $ = for making sure you catch an entire line

```
private String _language;

@Given ("the language is set to (.*)$")
public void setLanguageTo(String lang) {
   _language = lang;
}
```

```
private String _output;

@When ("I run the helloworld program$")
public void runHelloWorld() {
   h = new HelloWorld(_language);
   output = h.run();
}
```

```
@Then ("^the response is \"([^\"]*)\"$")
public void theResponseIs(String expected) {
   assertEquals(expected, output);
}
```

We Now Can Not Only Test The Code, but Also What the Users Expect to See

Given the language is English
When I run the 'Helloworld program'
Then the response is "Hello!"

Prezi

Step Definitions

All of those Given, When, Then statements are called *steps*.

They are defined in code in a separate "step definitions" file.

Usually, there is a 1:1 correspondence between feature files and step definitions, plus additional ones for common steps.

Prezi

Step Definitions Example

Let's say we have a feature file like so:

Feature: HelloWorld says Hello

Scenario: English hello
Given the language is set to English
When I run the helloworld program
Then the response is "Hello!"

Prezi

Step Definitions File

Remember there are two parts of a Gherkin feature file, the *feature* and the *scenarios*.

Also remember that the feature part is just freetext description. There's no need to make steps for it.

Prezi

Step Definitions Example

That leaves us with the scenarios. We will need to define the given, when, and then, in Java (or whatever language).

Prezi

Step Definitions Example

Let's add the proper imports first.

```
import cucumber.annotation.en.Given;
import cucumber.annotation.en.Then;
import cucumber.annotation.en.When;

import static org.junit.Assert.assertEquals;
```

Whoa! JUnit! What are you doing here?

Prezi

Sidenote

Remember I said BDD builds on TDD?

Cucumber builds on other testing software, e.g., JUnit or RSpec.

Prezi

Step Definitions

```
public class HelloworldStepDefs {

   // Now let's define the actual steps
   // here.  Just remember that they're
   // in this step definition class.
}
```

Prezi

```
private String _language;

@Given ("the language is set to (.*)$")
public void setLanguageTo(String lang) {
 _language = lang;
}
```

Prezi

Prezi

A Brief Refresher on Regexes (Regular Expressions)

(?:[a-z0-9!#$%&'*+/=?^_`{|}~-]+(?:.[a-z0-9!#$%&'*+/=?^_`{|}~-]+)*|"(?:[x01-x08x0bx0cx0e-x1fx21x23-x5bx5d-x7f]|[x01-x09x0bx0cx0e-x7f])*")@(?:(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?|[(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?).){3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?|[a-z0-9-]*[a-z0-9]:(?:[x01-x08x0bx0cx0e-x1fx21-x5ax53-x7f]|[x01-x09x0bx0cx0e-x7f])+)])

Anyone know what this is?

Prezi

A (or some other alphanumeric) = itself

.* = Any number of a character, or none

.+ = One or more of a character

Example:

ca*t -> matches ct, cat, or caaaat

ca+t -> matches cat or caaaat, not ct

Prezi

[0-9] or d = matches any digit
^ = beginning of line (just like vi!)
$ = end of line (just like vi!)
() = how you enclose regexes

Example

(d+) matches 1 or 984, but not "foo"
(bird$) matches "I saw a bird" but not "I saw a bird flying by"
(^Ice) matches "Ice ice baby" but not "I gave my baby some ice for his teething."

Prezi

What You'll Probably Use

(.*) = match any set of characters
(d+) = match any set of digits
(an?) = matches a or an (for English-ism)
(?:got a | got some) = matches either "got a" or "got some"
^ and $ = for making sure you catch an entire line

Prezi

```
private String _language;

@Given ("the language is set to (.*)$")
public void setLanguageTo(String lang) {
 _language = lang;
}
```

Prezi

```
private String _output;

@When ("I run the helloworld program$)"
public void runHelloWorld() {
   h = new HelloWorld(_language);
   output = h.run();
}
```

Prezi

```
@Then ("^the response is \"([^\"]*)\"$")
public void theResponseIs(String expected) {
   assertEquals(expected, output);
}
```

Prezi

We Now Can Not Only Test The Code, but Also What the Users Expect to See

Given the language is English
When I run the Helloworld program
Then the response is "Hello!"

Prezi

It's Your Turn

We want to test that FizzBuzz works appropriately. Let us assume that there is a FizzBuzz server running in the background that needs to be started up.

Given the FizzBuzz server is running
When I enter the value 120
Then _____

We are testing a new feature for Rent-A-Cat that informs users in real-time when a cat they requested is ready, if the user is logged in.

Given that user "X" has requested cat "Y"
And _____
When cat "Y" is ready to be picked up
Then user "X" should receive a message

Given the sequence being used is the Fibonacci
When the user enters the number 7
Then the system should return 13

@Then("^the system should return (_____)$")

Note this should only get digits, not any text

Given the sequence being used is the Triangle Numbers
When the user enters "7"
Then the value should be "21"

@Then("^the value should be \"([^\"]*)\"$")
public void theValueShouldBe(String f) {
_____
}

Given the FizzBuzz server is running
When the user enters the value "5"
Then the user should see "Buzz"

```
public class FizzBuzz {
  // Checks if server is running
  public boolean running() { ... }
  // Returns value of a particular input
  public String value(int n) { ... }
  // Prints out the FizzBuzz values from 1 to n
  public void printFizzBuzz(int n) { ... }
}
```

Given the user is in the jungle
When the user is in the village, the peaceful village
Then the lion sleeps tonight

```
public class User
  public boolean inJungle() { ... }
  public boolean inVillage() { ... }
  public boolean villageIsPeaceful() { ... }
  public boolean lionSleepsTonight() { ... }
}
```

Prezi

We want to test that FizzBuzz works appropriately. Let us assume that there is a FizzBuzz server running in the background that needs to be started up.

Given the FizzBuzz server is running
When I enter the value 120
Then _____

Prezi

We are testing a new feature for Rent-A-Cat that informs users in real-time when a cat they requested is ready, if the user is logged in.

Given that user "X" has requested cat "Y"
And _____
When cat "Y" is ready to be picked up
Then user "X" should receive a message

Prezi

Given the sequence being used is the Fibonacci
When the user enters the number 7
Then the system should return 13

@Then("^the system should return _____$")

Note this should only get digits, not any text.

Prezi

Given the sequence being used is the Triangle Numbers
When the user enters "7"
Then the value should be "21"

```
@Then("^the value should be \"([^\"]*)\"$")
public void theValueShouldBe(String f) {

  ____________________________________

}
```

Prezi

Given the FizzBuzz server is running
When the user enters the value "5"
Then the user should see "Buzz"

```
public class FizzBuzz {
   // Checks if server is running
   public boolean running() { ... }
   // Returns value of a particular input
   public String value(int n) { ... }
   // Prints out the FizzBuzz values from 1 to n
   public void printFizzBuzz(int n) { ... }
}
```

Prezi

Given the user is in the jungle
When the user is in the village, the peaceful village
Then the lion sleeps tonight

```
public class User
   public boolean inJungle() { ... }
   public boolean inVillage() { ... }
   public boolean villageIsPeaceful() { ... }
   public boolean lionSleepsTonight() { ... }
}
```

Prezi

It's Over!

It's Over!

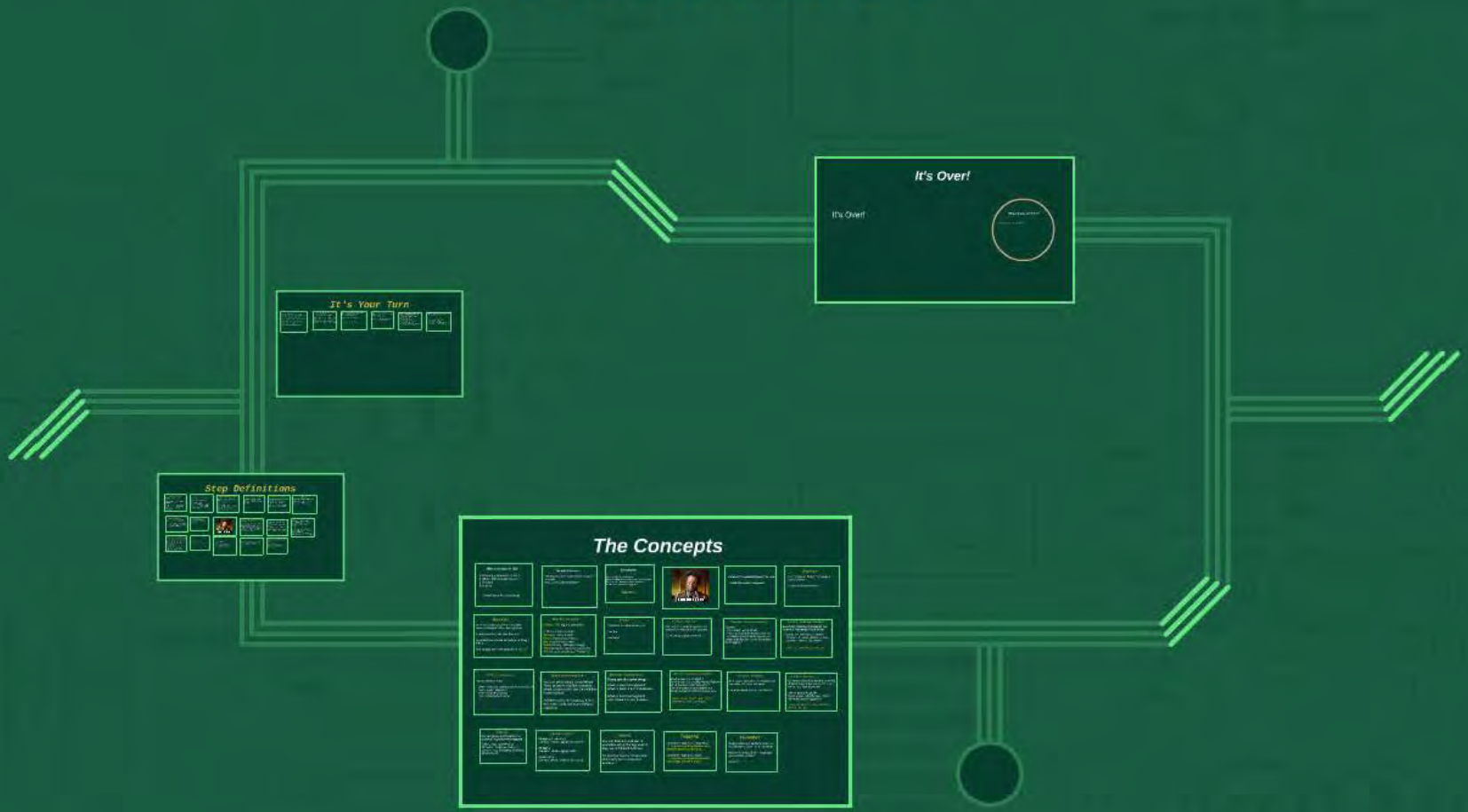
Prezi

Why are you still here?

The lecture is over! Go home!

Prezi

CS1699 - Lecture 12 - Automated Testing of BDD with Cucumber



Prezi