Week 7
Modules:7

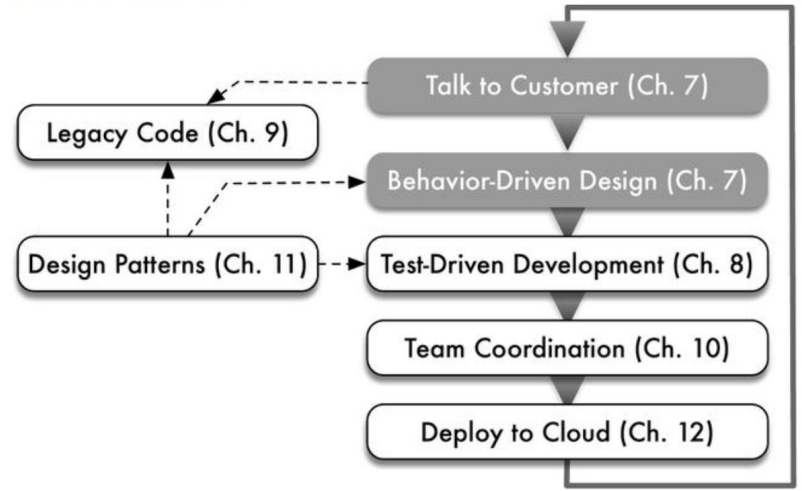
Behavior Driven Design

Topics:

- User Stories
- Cucumber & Capybara
- Pivotal Tracker

Agile & Behavior Driven Design (BDD)

- **Agile**: Working closely with stakeholders, maintaining a working prototype while deploying new features, continuously refining requirements
- **BDD**: asks questions about the behavior of an application before and during development so that the stakeholders are less likely to miscommunicate.
- Tests *behavior*, not implementation: **Validation**, not just **Verification**
  - Even if implementation changes, behavior stays the same
- One way to test is Cucumber/Capybara



Functional Requirements: User stories

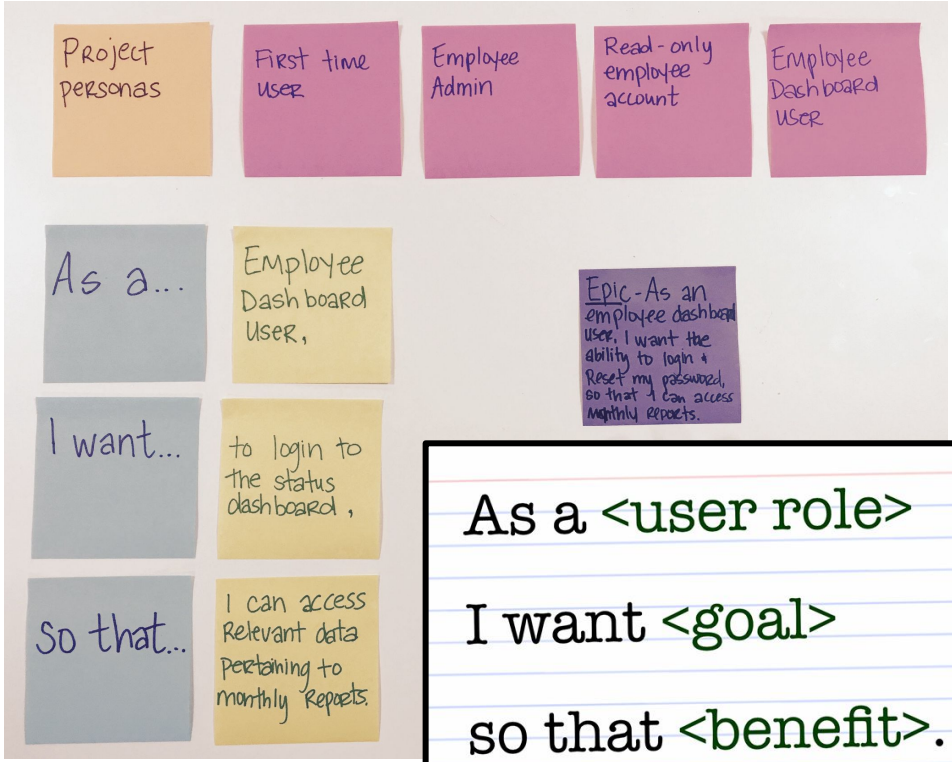
User stories capture app behavior, user needs, and expected usage of the app

- Written as a couple sentences on 3x5 index cards (or **Pivotal**) normal english without code/tech
- All stakeholders brainstorm and prioritize features
- Can consider different personas
- Must be Testable, Small enough to implement in one iteration and Have business value
-

Connextra Format: is how we write user stories

Feature name

As a [kind of stakeholder],
So that [I can achieve some goal],
I want to [do some task].*



SMART User Stories

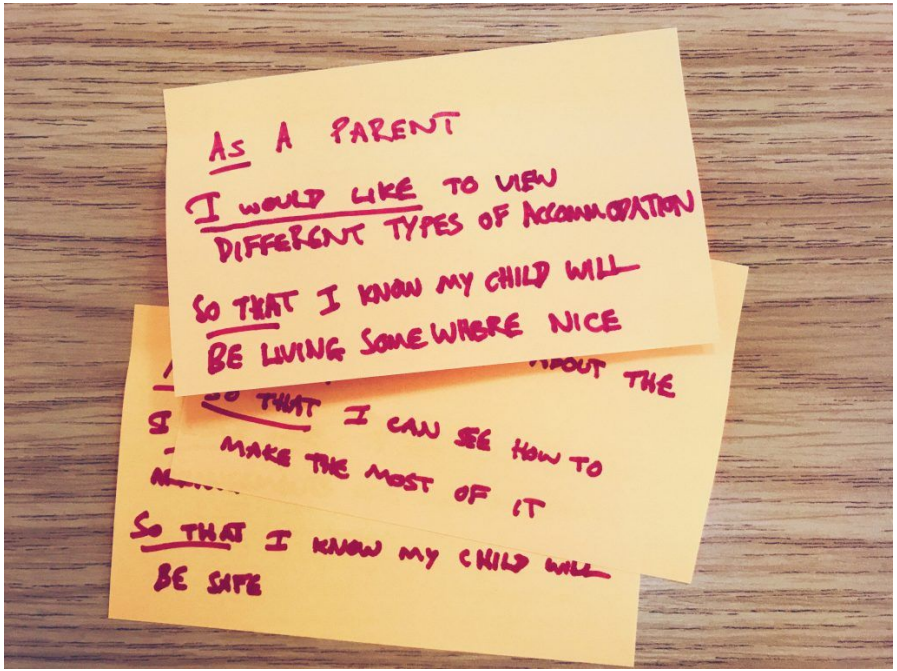
User stories should be SMART

- **S**pecific
- **M**easurable
- **A**chievable
- **R**elevant
- **T**imeboxed

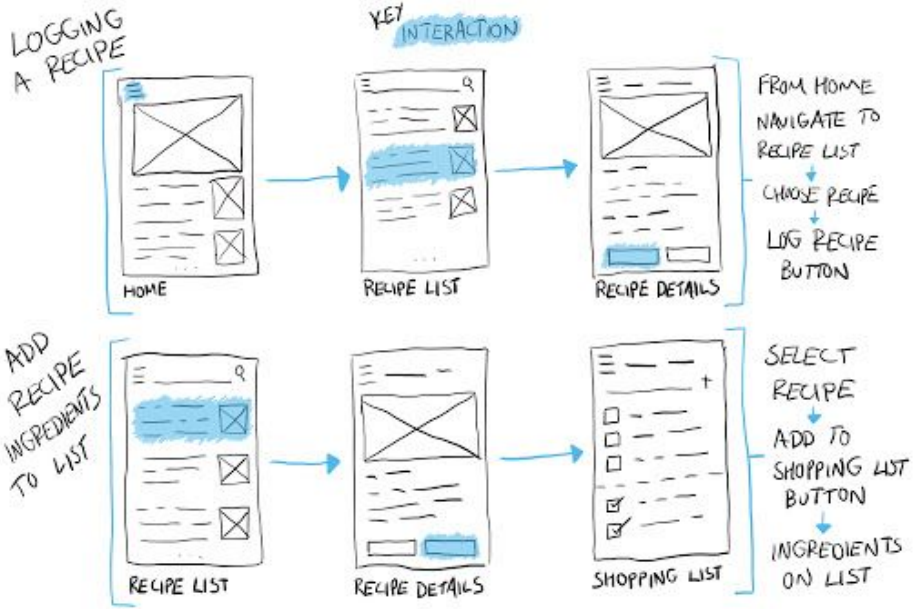
Example:
"resource providers can add resources to the database"

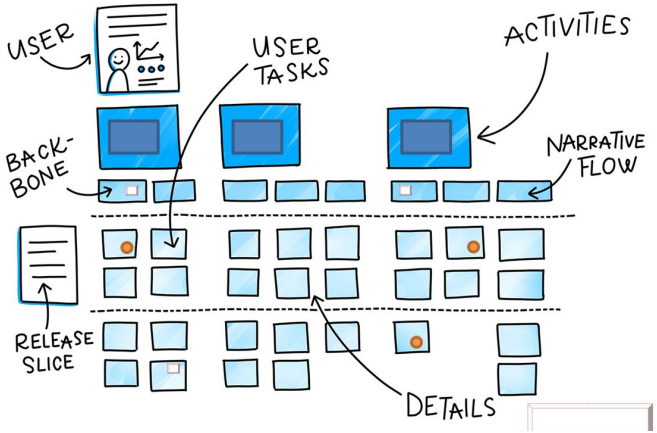
vs

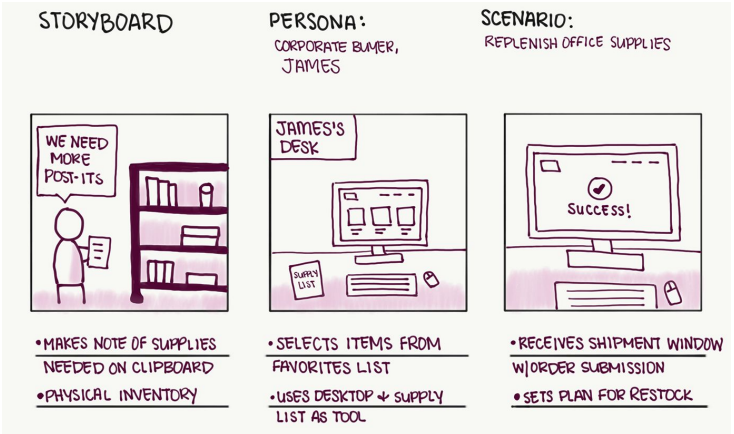
"**as a** *resource provider*, **so that** *my resource can be added to the approval queue for site admins to look at*, **I want to** *make a POST request to the Innovation Resources API with the minimum requirements for a resource to be added*"



UI Requirements: Low-Fidelity (Lo-Fi) interfaces, Storyboards







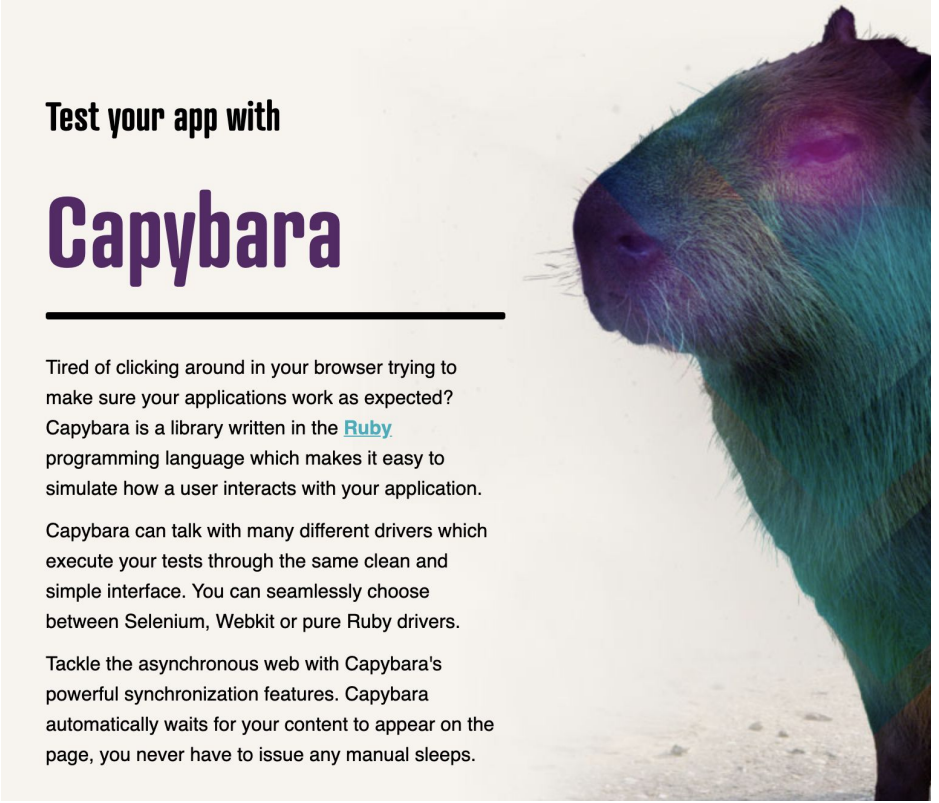
User Stories in Rails (coming soon: step definitions)

```
Feature name
  As a [kind of stakeholder],
  So that [I can achieve some goal],
  I want to [do some task]
```

```
Feature: Add a movie to RottenPotatoes
  As a movie fan
  So that I can share a movie with other movie fans
  I want to add a movie to RottenPotatoes database
Scenario: Add a movie
  Given I am on the RottenPotatoes home page
  When I follow "Add new movie"
  Then I should be on the Create New Movie page
  When I fill in "Title" with "Hamilton"
  And I select "PG-13" from "Rating"
  And I select "July 4, 2020" as the "Released On" date
  And I press "Save Changes"
  Then I should be on the RottenPotatoes home page
  And I should see "Hamilton"
```

Cucumber & Capybara

- User Stories to acceptance tests ⇒ Cucumber and Capybara
- **Cucumber** is a framework for writing user scenarios and turning them into acceptance tests.
  - Used alongside **Capybara**, the framework used to simulate the user's browser as they navigate through your webapp
- Cucumber steps:
  - Setup preconditions: **Given ...**
  - Action to test: **When ...**
  - Check postconditions: **Then ...**



Features, Scenarios and Step Definitions

- A **feature** is the new app behavior that we want to implement, typically embodied in a user story.
- **Scenarios** are the different ways a feature can be exercised.Specify and test the different behaviors, such as happy/sad paths
  - **Happy Path:** scenarios where the user and app both act exactly as you'd expect it to.
  - **Sad Path:** scenarios where the user and app don't act as you'd expect them to (malformed input, external service fails, edge/failure cases)

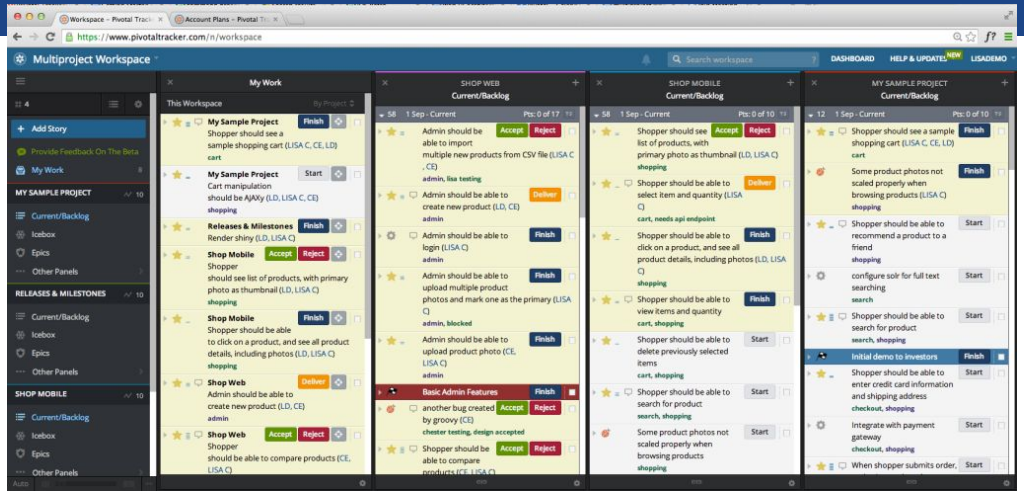
```
Feature: Write comments
  As a blog reader
  I want to be able to write a comment
  So that I tell the author my opinion / feedback

Background: Login with author user
  Given that the author adds a new post with title "Post to comment"


Scenario: Leave a comment with all info filled in
  Given that I select the post
  When I add a new comment with name, email  and  body
  Then I will see the comment on the blog


Scenario: Leave a comment with name field not filled in
  Given that I select the post
  When I add a new comment with email and body
  Then I will see the message "ERROR: please fill the required fields (name, email)."


Scenario:  Leave a comment with email field not filled in
  Given that I select the post
  When I add a new comment with name and body
  Then I will see the message "ERROR: please fill the required fields (name, email)."


Scenario: Leave a comment without body
  Given that I select the post
  When I add a new comment with name and email
  Then I will see an empty comment on the blog
```

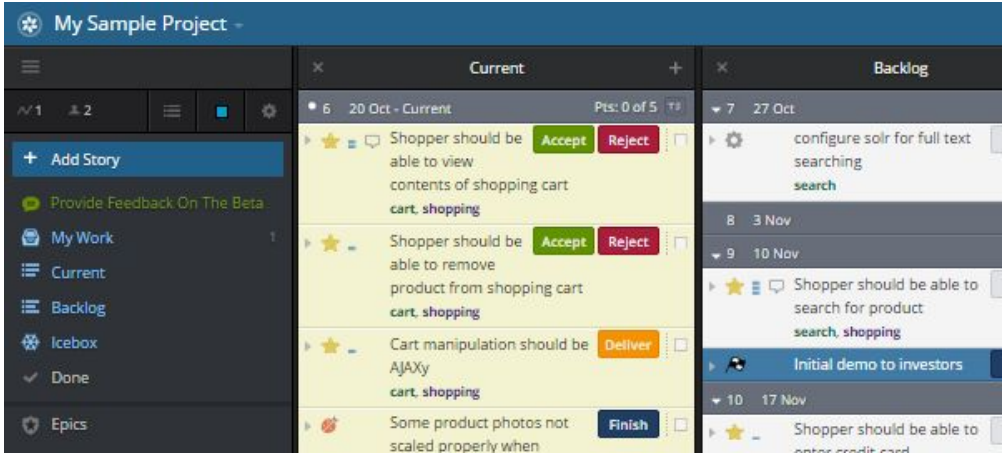
Example: Feature -> Scenarios -> Step Definitions

```
Feature: “As an e-commerce website user, so that I can aggregate all items I
want to buy in one place to look at later, I can add items of interest to my
cart.”

Scenario:
    Given I am on the detail page for the novel “Ready Player One,”
    When I click the “Add to Cart” button,
    Then the novel “Ready Player One” will be in my cart.

Step definitions:
    <regex matcher> do navigate_to_page(“/novels/id/1247234/detail”) end
    <regex matcher> do click_on_button(:name => “Add to Cart”) end
    <regex matcher> do assert user.cart.include? Novel.where(:title => “Ready
Player One”).first end
```

Productivity & Pivotal

- **Idea1**: Assign similar workloads to developers, (!= #user stories)
- **Idea2:** Assign a Point to each user story to represent its difficulty
- **Point Value:** represents expected number of coding hours
  - completely up to the team, and will likely differ across teams,
  - everyone on the team should be in rough agreement on how much a “point” is worth.
  - more points should represent not only more effort, but more uncertainty.
- **Planning Poker:** estimate points
- **Spike**
- **Velocity**





Walkthrough: Set Up

Feature: user login

Scenario: user can log in with correct user/password

Scenario: user sees reset password prompt on incorrect login attempt

Scenario: user locked out after three failed login attempts

```
# Let's work on the last scenario


Feature: user login

  Scenario: user can log in with ...
     < Given - When - Then >

  Scenario: user sees reset ...
     < Given _ When _ Then >

  Scenario: user locked out after three
failed login attempts
```

Walkthrough: (Cucumber) Scenario Definitions

```
# Let's work on the last scenario

. . .
Scenario: user locked out after three failed login attempts

  Given that an account with username "cs169student" and password "pg&e" exists,
  And that I have unsuccessfully tried to login to my "cs169" account "2" times
     already,
  When I try to log in with an incorrect password for username "cs169student",
  Then I should no longer see the login prompt,
  And I should be locked out of my account.
```

What's wrong with this test?

Common Pitfall for Sad Path testing

- Avoid loosely testing against a negative condition
- When your test only asserts that you no longer see an element on the returned page, you're saying that ANYTHING else is okay, such as
  - Some other unexpected page, so long as it doesn't have the specified element
  - A page with no content or unexpected content
  - Errors (e.g. some rails error page)
- As long as the back-end is updated correctly (user is locked out) and the login prompt disappears, anything goes ==> common place for big bugs to creep in

```
. . .
Scenario: user locked out after three failed login
attempts

  Given that an account with username "cs169student"
and password "pg&e" exists,
  And that I have unsuccessfully tried to login to
my "cs169" account "2" times
      already,
  When I try to log in with an incorrect password
for username "cs169student",
  Then I should no longer see the login prompt,
  And I should be locked out of my account
  And I should be locked out of my account.
```

Common Pitfall for Happy Path testing

- Avoid loosely testing against a positive condition
- When your test only asserts that you should see an element on the returned page, what if it appears multiple times?
  - A user named Emma adding the book “Emma” to her shopping cart
  - If the shopping cart doesn’t add the book successfully, the test would still pass if Emma’s account name was somewhere
- Use Capybara’s ‘*within*’ helper with CSS selectors to specify HTML elements in your scenario

```
. . .
Scenario: user adds book to shopping cart

  Given that an account with name “Emma” exists,

  And that I have successfully logged in to my
“Emma” account,

  When I add the book “Emma” to my shopping cart,

  And I go to the shopping cart page,

  Then I should see ”Emma” within
”div#shopping_cart”.
```

Walkthrough: Step Definitions

The actual ruby code you want to be executed (when there's a matching step)

```
. . . # under features
Given that an account with username “cs169student” and
  password “pg&e” exists,
. . .
```

```
# under step_definitions
Given /^(?:|that )an account with username “cs169student” and
password “pg&e” exists$/ do

    User.create(:username => “cs169student”, :password => “pg&e”)

end
```

What's wrong with this step definition?

DRY Step Definitions

- Try to generalize step definitions whenever possible.
- One simple strategy is by including capture groups in the matcher and assigning these to variables

```
# before
Given /^(?:|that )an account with username “cs169student” and password “pg&e” exists$/ do

    User.create(:username => “cs169student”, :password => “pg&e”)

end
```

```
# now
Given /^(?:|that )an account with username “([\S]+)” and password “([\S]+)” exists$/ do |user|, |pass|
          User.create(:username => user, :password => pass)

end
```

Advanced Cucumber

Background

What if all of the scenarios for a given feature require a common subset of preconditions?

```
Feature: Multiple site support
  Only blog owners can post to a blog, except administrators,
  who can post to all blogs.

  Background:
    Given a global administrator named "Greg"
    And a blog named "Greg's anti-tax rants"
    And a customer named "Dr. Bill"
    And a blog named "Expensive Therapy" owned by "Dr. Bill"
```

```
Scenario: Dr. Bill posts to his own blog
  Given I am logged in as Dr. Bill
  When I try to post to "Expensive Therapy"
  Then I should see "Your article was published."

Scenario: Dr. Bill tries to post to somebody else's blog, and fails
  Given I am logged in as Dr. Bill
  When I try to post to "Greg's anti-tax rants"
  Then I should see "Hey! That's not your blog!"

Scenario: Greg posts to a client's blog
  Given I am logged in as Greg
  When I try to post to "Expensive Therapy"
  Then I should see "Your article was published."
```

(Pics from Gherkin reference docs)

Tables

What if you want to pass a list of values or objects to a step? Can use tables.

```
Given the following users exist:
  | name   | email              | twitter         |
  | Aslak  | aslak@cucumber.io  | @aslak_hellesoy |
  | Julien | julien@cucumber.io | @jbpros         |
  | Matt   | matt@cucumber.io   | @mattwynne      |
```

```
Given /^the following users exist:$/ do |table|
    table.hashes.each do |acct|
        User.create(. . .)
    end
end
```

Instance Variables

What if you need some value(s) to persist between steps?

```
# Handy for things like preserving credentials and service results
# e.g. some JSON to be used by various parts of your app

    Given /^some step$/ do
        @state = . . .
    end
    …
    Given /^some other step$/ do
        expect(@state).to be_valid
    end
```

Nested Steps

What if you have some composite step that can be performed by executing some pre-existing step defs?

```
Given /^I have unsuccessfully made two login attempts$/ do
    steps %Q{
        When I make an unsuccessful login attempt
        And I make an unsuccessful login attempt
    }
endc
```

XPath

- Another powerful tool to use with Cucumber is xpath (although you can select elements with pure CSS, too. Read the docs)
- Can filter out parts of returned page in HTML DOM by element type and/or CSS attributes/identifiers
- Also helps mitigate the sad path pitfall by allowing for more specificity
  - Rather than testing "I do not see . . ." (anywhere on the page), you can test whether the HTML element with id [id] is present within [some known parent element]
  - Can combine this with other positive checks to better define expected behavior

```
. . .

div_to_test = page.find(:xpath, './/div[contains(@id, "bar")]')

. . .
```

Exercises

Please go into your assigned break-out rooms and work on the following exercises for the duration your TA allocated. The rest of the section will focus on working on these practice questions.

Q1. Exercise: User Stories

User stories should be SMART
- Specific
- Measurable
- Achievable
- Relevant
- Timeboxed

1. Discuss whether the following user story is **SMART**:

   - "The UI should be intuitive."
   - "The login UI should be so intuitive that 80% of customers can log-in within twenty seconds."
   - "As an e-commerce website user, so that I can go to cart and checkout my purchase seamlessly, the pre-purchase login UI should be so intuitive that 80% of customers can log-in within twenty seconds."

Q2. Exercise Answer the following questions

1. How can you quantitatively measure the productivity of a team?
2. Name a tool that you can use to prioritize and keep track of user stories?
3. What are some examples of actions that Capybara can do to simulate actions of real users using your app?
4. Name a technique that can help you uncover the real value of a user story.

Q3. Quiz Review: User Story

Which of the following elements, if any, is NOT an element that a user story should capture?

A) A task that a particular stakeholder wants to accomplish
B) The role of a stakeholder
C) The business reason why the task is important to the stakeholder
D) The approximate effort (points) required to code the functionality
E) All of the above should be captured by a user story

Q4. Quiz Review: User Story

What are the five most relevant characteristics of a good user story?
Mark all that apply.

I. Achievable
II. Relevant
III. Specific
IV. Measurable
V. Timeboxed
VI. Estimable
VII. Independent
VIII. Negotiable
IX. Small
X. Testable
XI. Valuable

Q5. Quiz Review: User Story

Which of the following is true about user stories? (i) they should describe how the application is expected to be used (ii) they should have business value (iii) they do not need to be testable (iv) they should be implemented across multiple iterations of the Agile lifecycle

A. i only

B. i and ii

C. i and iv

D. i, iii, and iv

Q6. Quiz Review: Imperative vs Declarative Scenarios

Which statements are TRUE regarding imperative vs. declarative scenarios?

A) Imperative scenarios are OK if the details of the scenario are the focus of the test
B) Declarative scenarios should always be preferred over imperative scenarios
C) Imperative scenarios are necessary when the customer is nontechnical
D) Declarative scenarios can often re-use steps from imperative scenarios

Q7. Quiz Review: Implicit & Explicit Requirements

Which of the following is true about implicit requirements and explicit requirements?

A) You cannot write user stories for both explicit and implicit requirements
B) Implicit requirements tend to be more concise, while explicit requirements tend to be more verbose
C) Implicit requirements are the logical consequence of explicit requirements, and typically correspond to integration tests
D) Explicit requirements are usually defined with imperative scenarios and implicit requirements are usually defined with declarative scena

Q8. Quiz Review: BDD

The goal of Behavior-Driven Design (BDD) is:

(i) to verify that the application meets the specification
(ii) to validate that the design does what the customer wants
(iii) to help the customer understand the use of the application
(iv) to ask questions about the behavior of an application before and during development

A. i and ii
B. i, ii, and iii
C. i, ii, and iv
D. i, ii, iii, and iv

Q9. Quiz Review: Cucumber & Step Definition

For the Cucumber step Given RottenPotatoes contains a movie "Rambo" with rating "PG" which of the following COULD be true about its step definition?

A) It could set up this precondition by calling Movie.create
B) It could set up this precondition by calling a sequence of steps corresponding to how an end user would add this movie manually (go to New Movie page, submit form, etc.)
C) It would need at least 2 regular-expression capture groups
D) If the step appears in a Background: section, the step definition will be called only once even if there are multiple scenarios after the Background section

Q10. Quiz Review: Cucumber & Step Definitions

In terms of how Cucumber executes scenarios (features), what is the difference between Given, When, and Then steps?

A. Only When steps (representing actions to be taken) can modify application state or have side effects, for example via POST requests.
B. You must specify at least one Given step before any When or Then steps.
C. There is no difference in execution; Given, When, and Then are aliases for the same method

Q11. Quiz Review: Cucumber & Step Definitions

What is the role of the Background steps?

A. It makes the user story DRYer by factoring out common steps across all scenarios of one feature for this app.
B. It makes the user story DRYer by factoring out common steps across all scenarios for this app.
C. It is a 'holding area' for steps that have not yet been implemented.
D. It sets the environment for this app.

Q12. Quiz Review: Cucumber & Step Definitions

For the Cucumber step Given RottenPotatoes contains a movie "Rambo" with rating "PG" which of the following COULD be the first line of its step definition?

A.
```
Given /RottenPotatoes contains a movie (".*") with rating (".*")/ do
|movie,rating|
```

B.
```
 Then /RottenPotatoes contains a movie (".*") with rating (".*")/ do
|movie,rating|
```

C.
```
 When /RottenPotatoes contains a movie (".*") with rating (".*")/ do
|movie,rating|
```

D.
```
 Then /RottenPotatoes contains a movie (".*")(with rating (".*"))?/ do
|movie,rating|
```

E.
```
Then /RottenPotatoes contains a movie (".*")(with rating (".*"))?/ do
|movie,has_rating,rating|
```

Q13. Quiz Review: Cucumber & Step Definitions

For the Cucumber step Given RottenPotatoes contains a movie "Rambo" with rating "PG" which of the following COULD be true about its step definition?

A. It could set up this precondition by calling Movie.create
B. It could set up this precondition by calling a sequence of steps corresponding to how an end user would add this movie manually (go to New Movie page, submit form, etc.)
C. It would need at least 2 regular-expression capture groups
D. If the step appears in a Background: section, the step definition will be called only once even if there are multiple scenarios after the Background sectio

Q14. Quiz Review: True or False

- You need to implement all the code being tested before Cucumber will say the test passes
- A sad path Cucumber scenario can pass without having the code written need to make a happy path pass
- Cucumber matches step definitions to scenario steps using regular expressions, and Capybara pretends to be a user that interacts with the SaaS application according to these scenario
- The purpose of the Lo-Fi UI and storyboards is to debug the UI before you program it.

Attendance: tinyurl.com/cs169a-disc-7-attendance
Password: cucumber