Cucumber

Feature File

Background

Scenarios

Scenario Outline

Step Def: When feature is ran, console generates a code snippet

Class:

Inside the step def: calling our page objects/functions

3 different ways to drive data from feature file.

1. useing regular expression

Giver user is on the login page as "www.google.com"

When user enters username as "abc"and password as "123"

Then user should be able to login to "www.google.com".

2. Data Table

Given user is on the [login page]

|Login Page URL|

When user enters [username] and [password]

|Login Page URL| Username |

Then user should be able to [login]

3. Scenario Outline

Giver user is on the <login page>

When user enters <username> and <password>

Then user should be able to <login>

Scenario Outline
Example:

Login Page URL	Username	Password	Home Page
skjdfhskjh	sdfsdfsdfsd	fwewfweewff	wfewfwef

Tag
Hook

4 scenarios in one feature file

2 for Regression
2 for Smoke

Using tag we can run specific scenarion/tests

Hook

[BeforeScenario]

[AfterScenario]

Maven:

Setting up Environment:

you need to have

1. JDK
2. Eclipse
3. Selenium Jars

Create a java project

Create a lib folder

copy the jar files in the lib folder

Jar files: Selenium 2.56.0

add those jar files into the build path

Use a Build Management Tool

Maven

Dependen

Windows - Preferences - Java - Installed JREs

- Execution Environments - select jdk version (1.8) - apply and Close

Why Maven is being used:

1. Dependency Management
2. For Controlling varsions (Git - Anik will go over)
3. for CI/CD (Jenkins) (dont worry, anik will handle)

agileTech

Question: When you write something on a text box and you get

some autosuggestions, how do you choose your desired text

In other words:

how do you hover your mouse on a web element and select

desired

1. Action Class

Example: hovering over mouse

Drag and drop

press enter (or any key)

right click

```
Actions action= new Actions(driver);
```

```
action.moveToElement(driver.findelementbyXpath);

action.build().perform();
```

Sprint 0

6 stories

Created in Jira

Week 1

Backlog Grooming Meeting:

BA describes the stories.

Everyone in the team (DEVs and Testers) the make themselves clear. Ask questions and feedback.

Week 2

Sprint Planning Meeting

1. We estimate the stories
2. We Assign the stories to members

1 = couple of hours

3 = 1 day

5 = 2-3 days

8 = 5 days

13 = whole sprint

20 = Beyond 1 sprint

Story = 5+ 3 = 8

6 members

4 devs

2 testers

each tester will be assigned with 2 devs

Sprint 1

Daily standup

BAcklog grooming

Sprint Planning Meeting

Sprint Review Meeting

Retrospective meeting

sprint 2

The requirements are broken down into small pieces

Small Pieces: User Stories

Story

Task

Bug

BA collect requirements from client (BRD)

BRD/Requirements are broken down into Stories

Sprint Review Meeting:

Reviewing the product

Demo whatever is developed

feature

Participants: Product owner, BA, Devs. testes, client

Sprint Retro

Reviewing the process

only the team members

Regression Test

Smoke Test

UAT Test

Agile

Sprints

Exit criteria is met

Release

1. all user stories are completed and signed off by the QA

2. Met exit criteria

Exit Criteria:

No major bugs

1 medium bugs

2-3 minor bugs

UAT:

User acceptance Testing

Participants:

stakeholders (team, SM, BA)

end users (client and users)

UAT Script:

Difference between Test case and UAT script

Known Issues

Backlog

3 months release, 6 months release

Backlog:

100 bugs in the backlog

Environments

Test environment (Devs are pushing the build after completion of each story)

Preprod (stories are deployed into preprod)

Prod (after 6 sprints)

DRY Principal

D = do not

R = repeat

Y = yourself

Why we use OOP:

for handling any change in the code/application

Selenium:

Select class

1 visible text

2 index

3 value

Wait

Implicit

\

Explicit

thread.sleep

Explicit

Fluent

thread.sleep

Locators:

Id

Name

Xpath

CSS selector

link text

partial link text

class

tagname

//a[contains(text(),'Products')]

/html/body/div[1]/div[3]/form/div[1]/div[1]/div[3]/center/input[1]