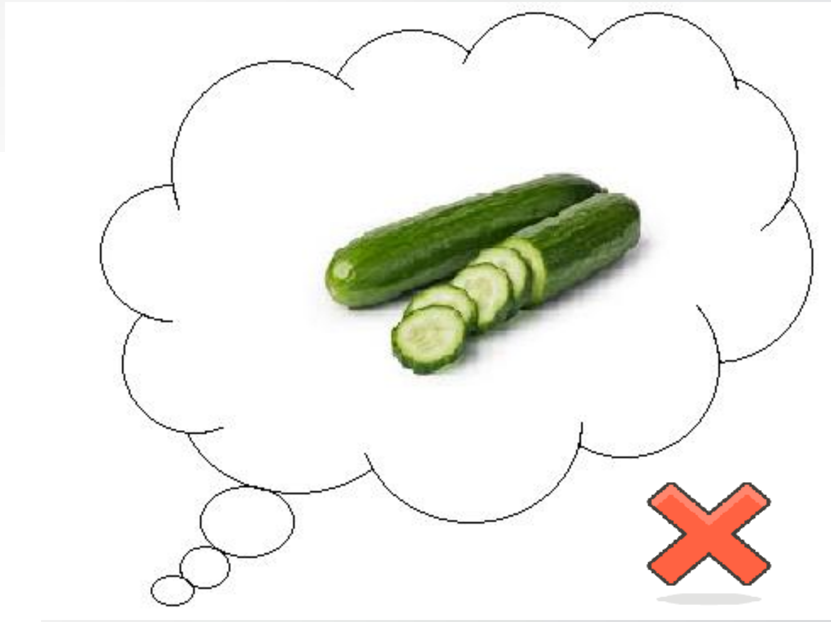
By Mithilesh Singh

Cucumber

It is a testing framework which supports behaviour driven development. It lets define application behaviour in plain meaningful English text using single grammar defined by

the language called "Gherkin".



BDD

Behavioral Driven Development (BDD) is a software development approach. It differs by being written in a shared language, which improves communication between tech and non-tech teams and stakeholders.

Why we use BDD

- Let’s assume there is a requirement from a client for an E Commerce website to increase the sales of the product with implementing some new features on the website, the only challenge of the development team is to convert the client idea in to something that actually delivers the benefits to client, The origin idea is awesome but the only challenge here is that the person who is developing the idea is not the same person is not the same person who has that idea. If the person who has the idea happens to be a talented software developer. Then we might be in luck the idea could be turned into working software without even needing to be explained to anyone else. Now the idea needs to be communicated and has to travel from the business owner(client) to the development team.

Note: Cucumber helps to improve communication between technical and non-technical members in the same project

Gherkin Keywords

* Feature.
* Scenario.
* Given.
* When.
* Then.
* And.
* Background

* Feature	: Each Gherkin file begins with a Feature keyword. Feature defines the logical test functionality you will test in this feature file
* Background	: Background keyword is used to define steps that are common to all the tests in the feature file.
* Given	: It is nothing but precondition
* When	: keyword defines the test action that will be executed.
* Then	: verification of the output with expected result
* And	: keyword is used to add conditions to your steps.
* But	: Use to give negative type comments.

Difference between testNg explanation and Cucumber explanation of a scenario

TestNg :

```
@Test
Public void testAdminUserCanUpdateUserAccPswd(){
// Create Users
User userAdmin = new User(UserRole.ADMIN, userName, Password);
Use admin user to update another user password
String message = userAdmin.updatePassword(user,user_new_password);
//Verify password changed
 Assert.assertEquals(message,"password changed successfully");
Assert. assertEquals(user.getPassword(), user_new_password);
}
```

Cucumber:

```
Feature   : Update password
Scenario  : Admin user can update the user password
Given     : I am in the HR system with an admin account
When      : I have updated password of the another user
Then      : I receive a message for updating password successfully
And       : User's password is updated to the new password.
```

Options available in cucumber:

dryRun	: true : checks if all the feature file steps have created step false: default condition
Features	: set: the path of the feature file.
glue	: set: the path of step definition file
tags	: instruct: what tags in the feature file should be executed.
monochrome	: true: display the console output in much readable way
format	: set: different report format which we can use. E.g. html, json.
A.	"**Pretty**" -- print the gherkin source with additional colors **format**={"pretty"}.
B. **HTML**	: this will help to generate HTML reports at the location mentioned in the formatter itself. **format =** {"html:Folder_Name"}.
C. **Json**	: this report contains all the information from the gherkin source in json **format**={"json:Folder_Name/cucumber.json"}.

Notes

- 1. Cucumber options we always keep under the @CucumberOptions annotation under the test runner class inside our framework.
  E.g.

```
@CucumberOptions(
    features = "src/test/resources/features",
    glue = "com/demo/qa/ui/steps",
    tags = "not @ignore"
    dryRun= true)
```

Components of the Cucumber framework:

1. Feature file

Scenario outline	: verify updating user password feature.
Given	: I am in the HR system with "<account_type>"account
And	: There is another user with "<old_password>" password
When	: I Update password of the user to "<new_password>"
Then	: I got the message "<message>"
And	: the user password should be "<final_password>"

2. Step definition file:

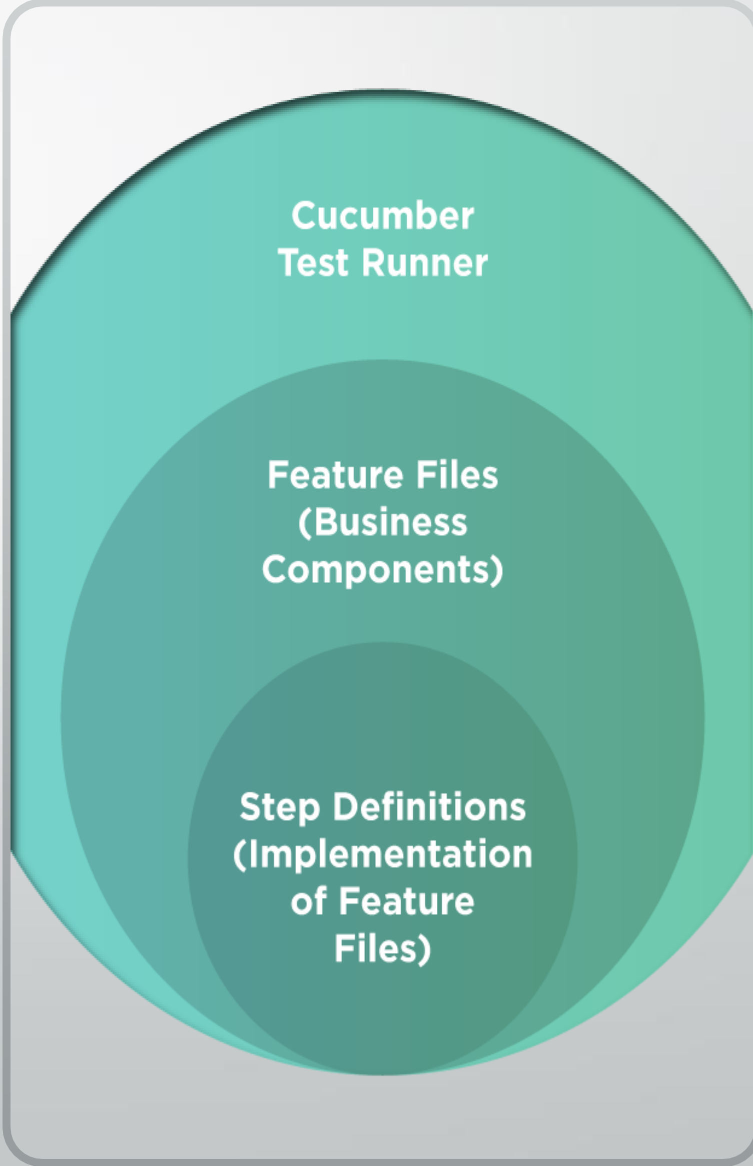
It is a small piece of code with a pattern attached to it.

Or

It is a java method in a class with an annotation above it.

Note: Cucumber finds the step definition file with the help of glue code in cucumber option.

3. **Runner Class:** To execute the case we need cucumber test runner class.


Cucumber Test Runner
Feature Files (Business Components)
Step Definitions (Implementation of Feature Files)

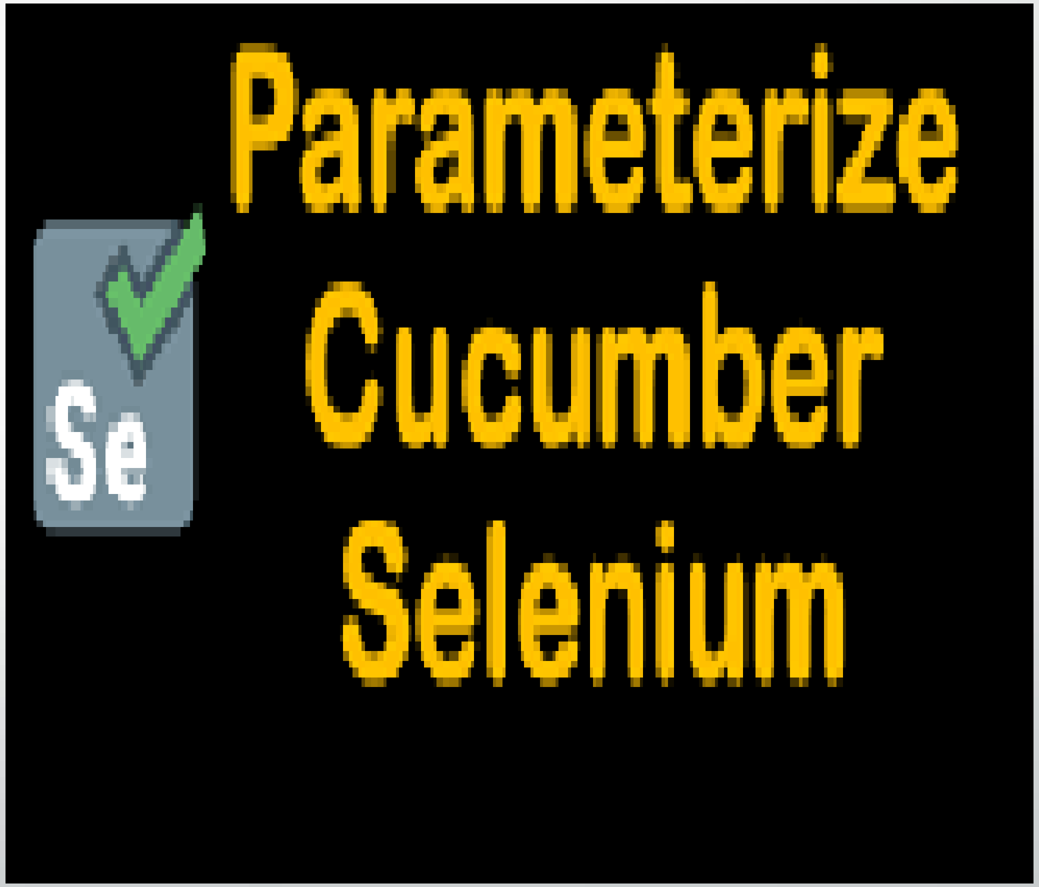
Data Driven Approach in Cucumber Framework:

A. parameterization without an example keyword.

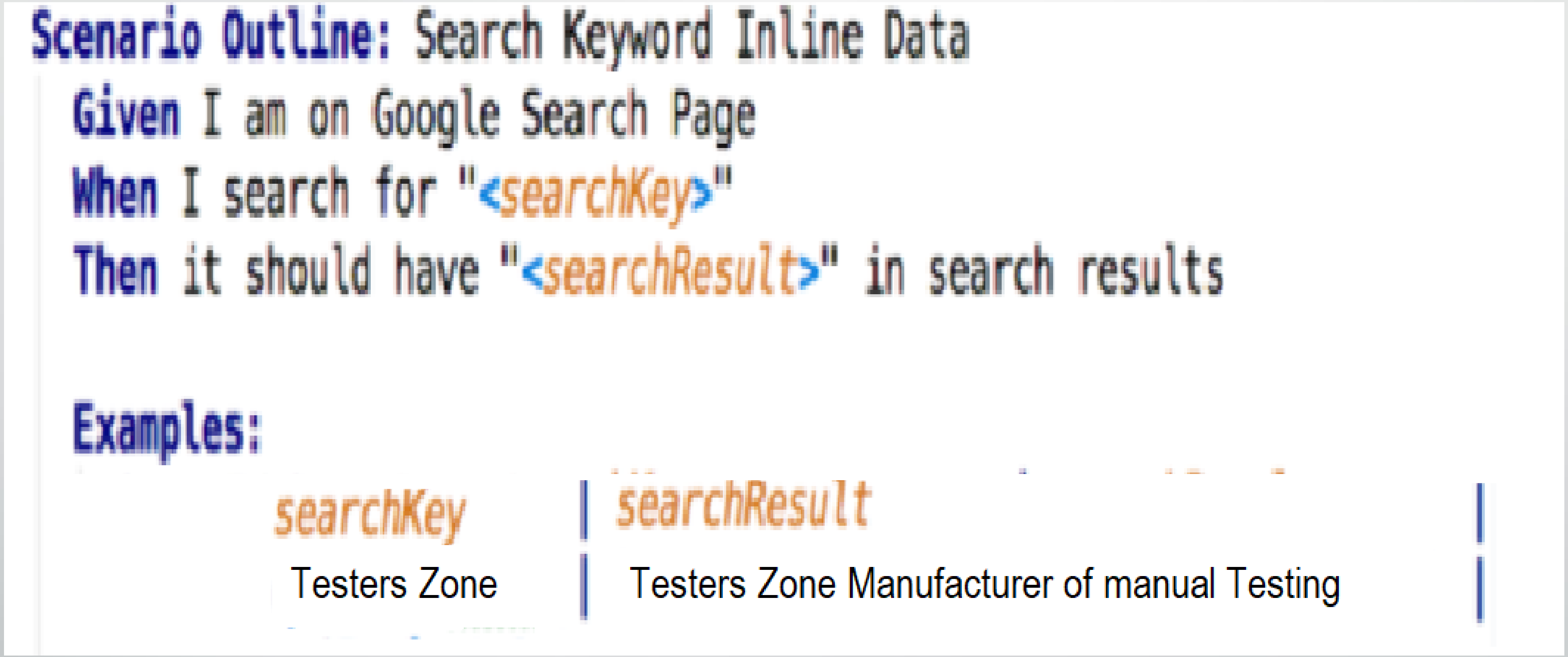
Scenario: Gmail login Test scenario

- Given User is already in login page
- When Title of login page is Gmail
- Then user enters "ABC" and "test@123"
- Then user clicks on login button
- Then user is on home page

Note: we can not test for different no of test inputs in this approach. To overcome this we have to go with the approach 2



B. parameterization with example keyword

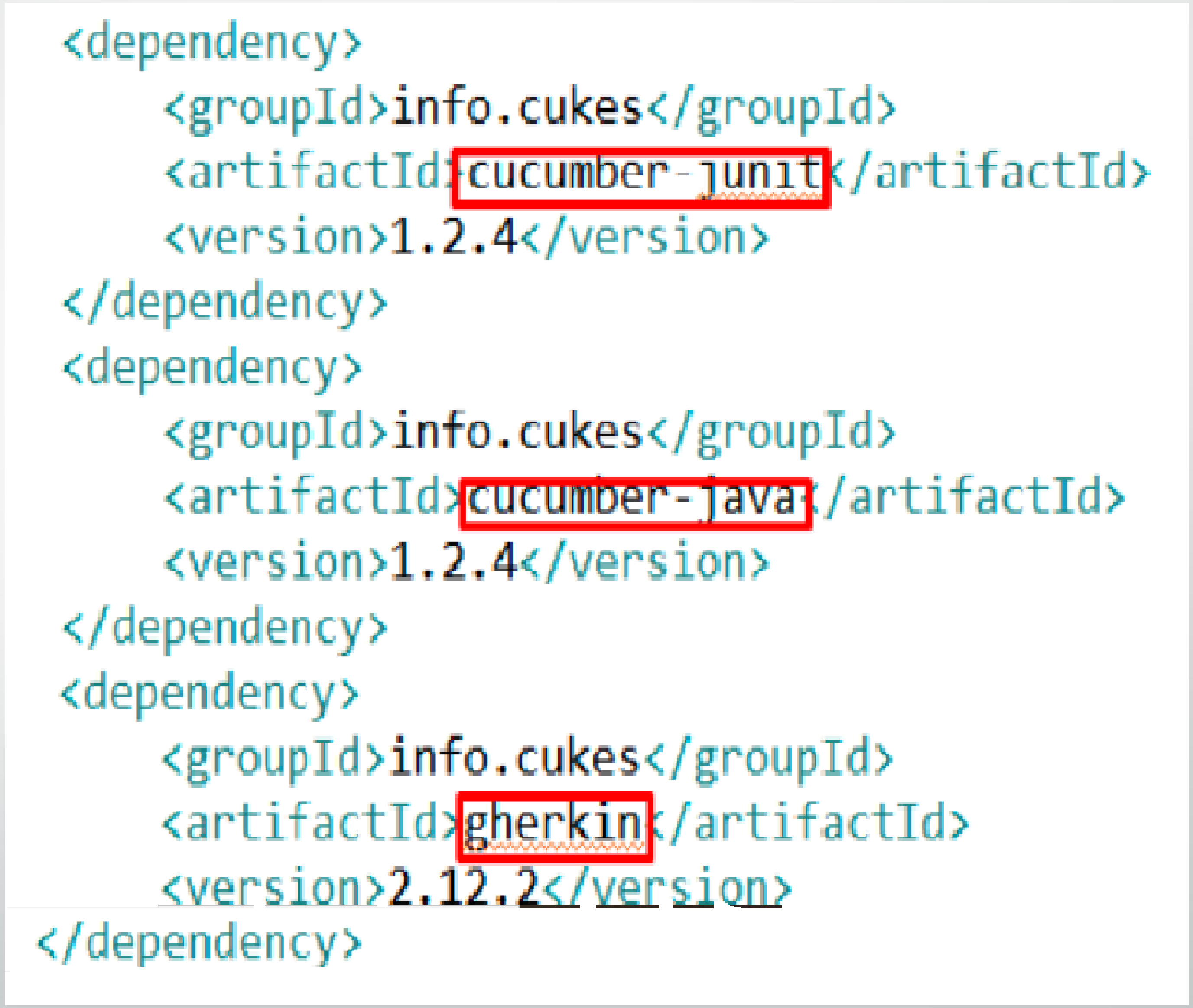


```
Scenario Outline: Search Keyword Inline Data
  Given I am on Google Search Page
  When I search for "<searchKey>"
  Then it should have "<searchResult>" in search results

  Examples:
        searchKey      | searchResult                                 |
         Testers Zone  | Testers Zone Manufacturer of manual Testing  |
```

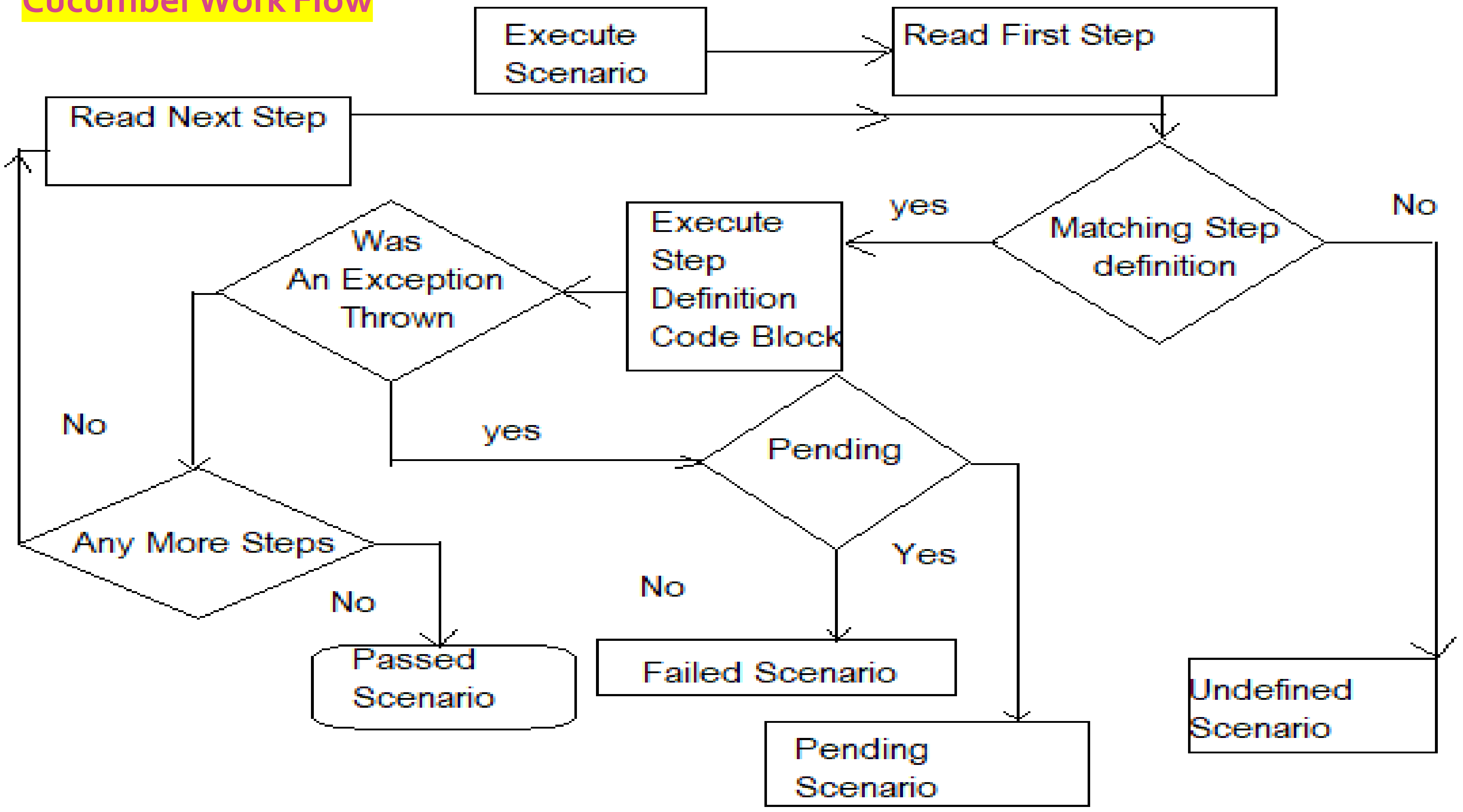
https://www.facebook.com/Testers-Zone-107916170837875/

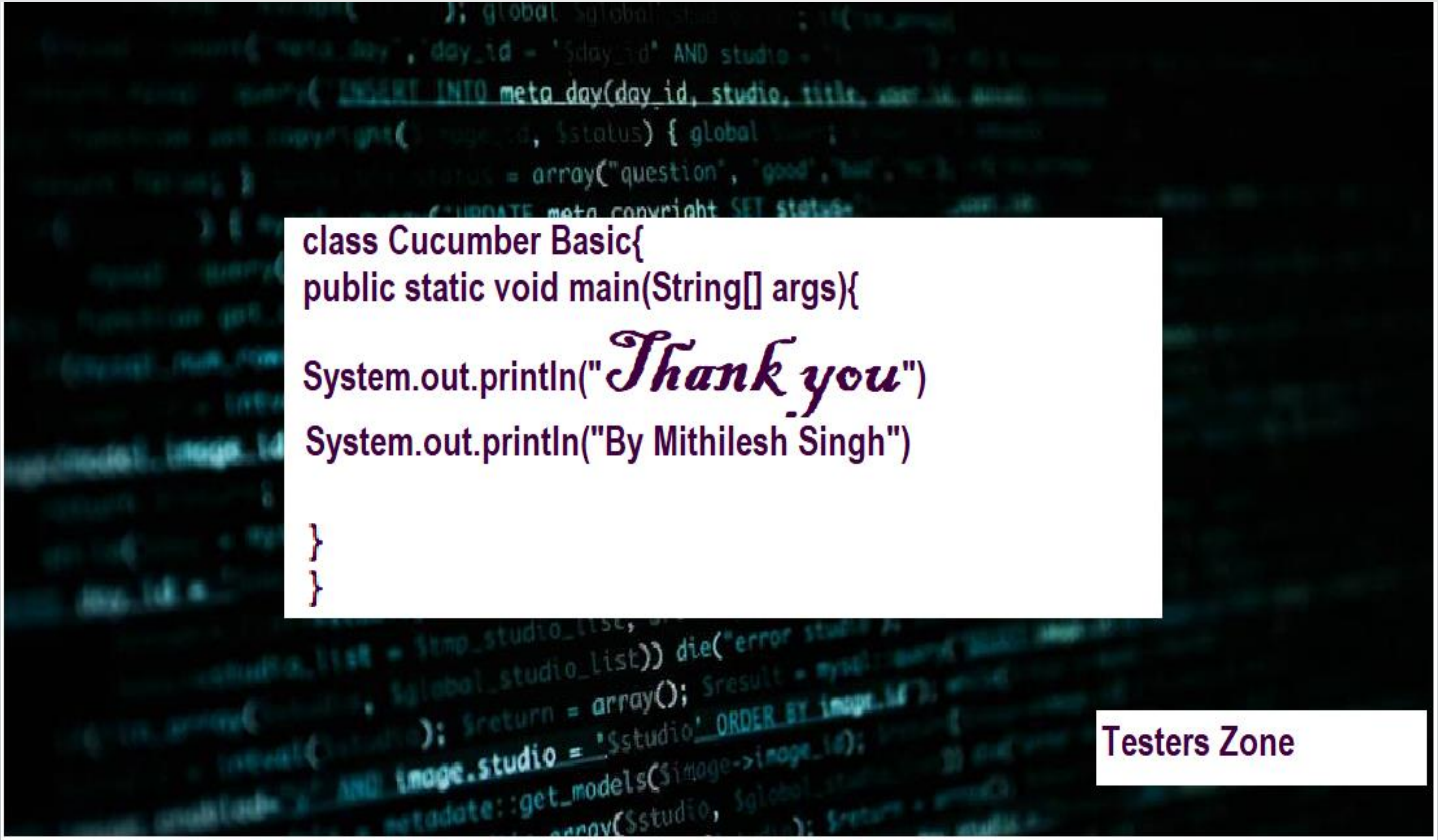
Maven Dependency for Cucumber



```
<dependency>
    <groupId>info.cukes</groupId>
    <artifactId>cucumber-junit</artifactId>
    <version>1.2.4</version>
</dependency>
<dependency>
    <groupId>info.cukes</groupId>
    <artifactId>cucumber-java</artifactId>
    <version>1.2.4</version>
</dependency>
<dependency>
    <groupId>info.cukes</groupId>
    <artifactId>gherkin</artifactId>
    <version>2.12.2</version>
</dependency>
```

Cucumber Work Flow





```
class Cucumber Basic{
public static void main(String[] args){

System.out.println("Thank you")
System.out.println("By Mithilesh Singh")


}
}
```

Testers Zone

https://www.facebook.com/Testers-Zone-107916170837875/