



Exploiting the potential of Behavior Driven Development (BDD) using Selenium-cucumber automation framework

```
...fier_ob.  
... object to mirror  
..._mod.mirror_object  
...operation == "MIRROR_X":  
...mirror_mod.use_x = True  
...mirror_mod.use_y = False  
...mirror_mod.use_z = False  
...operation == "MIRROR_Y":  
...mirror_mod.use_x = False  
...mirror_mod.use_y = True  
...mirror_mod.use_z = False  
...operation == "MIRROR_Z":  
...mirror_mod.use_x = False  
...mirror_mod.use_y = False  
...mirror_mod.use_z = True  
  
...selection at the end -add  
..._ob.select= 1  
...fier_ob.select=1  
...context.scene.objects.active  
...("Selected" + str(modifier...  
...mirror_ob.select = 0  
...= bpy.context.selected_object  
...data.objects[one.name].select  
  
...print("please select exactly  
  
... OPERATOR CLASSES ----  
  
...types.Operator):  
... X mirror to the selected  
...mirror_mirror_x"
```

The fast paced IT software development environment is creating challenges for the software developers to create an ecosystem, which can support rapid prototyping, design, agile development and testing. It also put pressure on the quality assurance community to create an integrated automation tool, which can translate to all stages of software development. Automation testing is the key factor in reducing the lead time in new product deployment. However, there are several challenges in the deployment of automation framework as several different test automation tools and scripting languages are available in the market. The IT companies need to invest in training and building resources to utilize these resources effectively and adopt latest industry practices such as behavior-driven development (BDD) and behavior-driven test (BDT).

Introduction

The fast paced IT software development environment is creating challenges for the software developers to create an ecosystem, which can support rapid prototyping, design, agile development and testing. It also put pressure on the quality assurance community to create an integrated automation tool, which can translate to all stages of software development. Automation testing is the key factor in reducing the lead time in new product deployment. However, there are several challenges in the deployment of automation framework as several different test automation tools and scripting languages are available in the market. The IT companies need to invest in training and building resources to utilize these resources effectively and adopt latest industry practices such as behavior-driven development (BDD) and behavior-driven test (BDT).

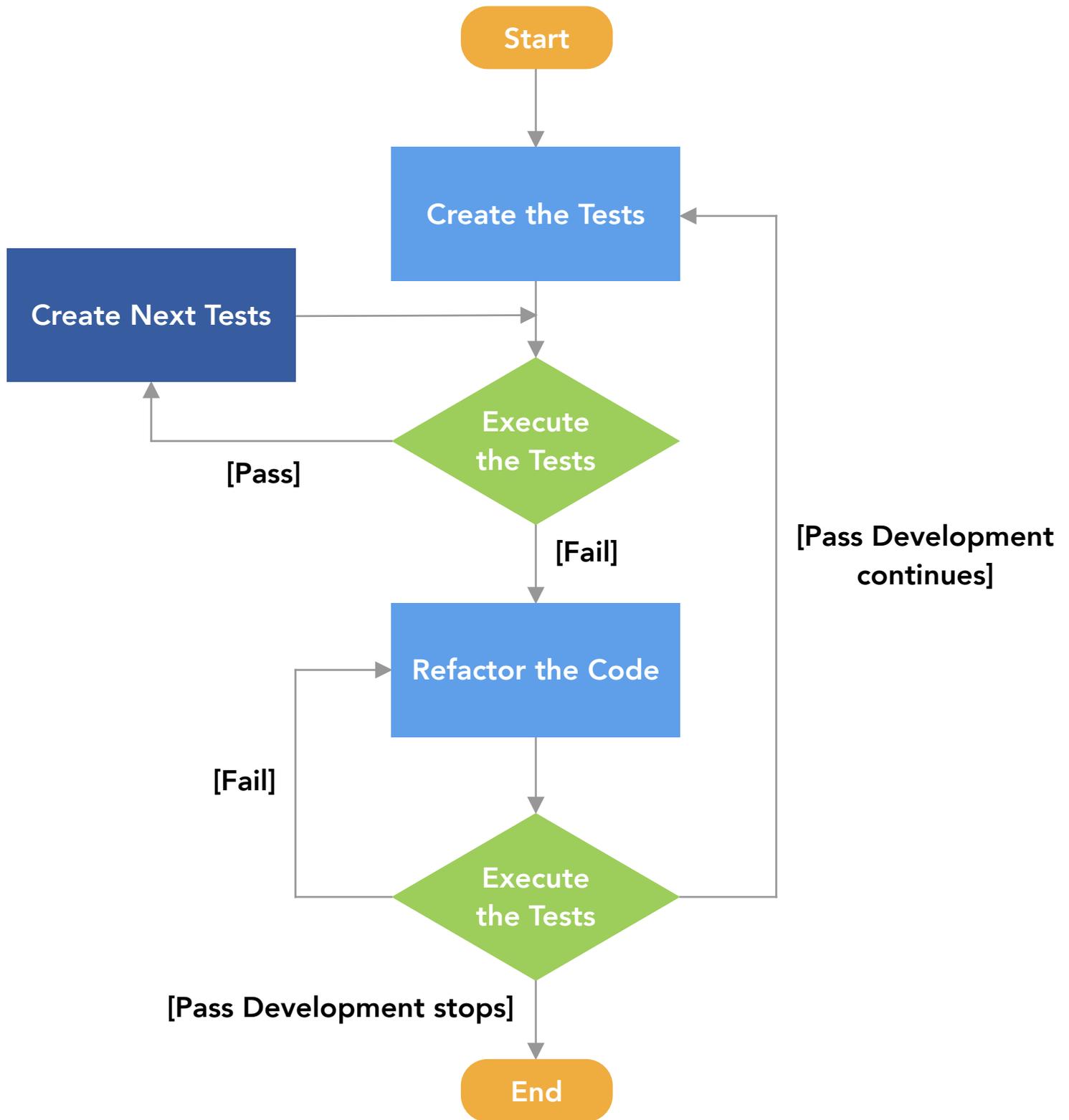
Introduction to Behavior Driven Development

Behavior driven development is a software paradigm rooted in the principle of test-driven development. The behavioral-driven development is facilitated by a simple language, which shows the development algorithm in simple English-like sentences. It is agile software development technique, which tries to combine the business needs and software development techniques.

BDD Process

Behavior Driven development addresses various cost-driven and behavior-driven IT challenges in a holistic manner. It ensures that all the stakeholders are involved in the software development process and a flexible platform is created, facilitating seamless communication between different. The BDD process creates a single point of interaction between all the stakeholders. Following process is adopted in BDD development:

- The user requirements are divided into smaller scenarios, which are compiled in a feature file.
- These scenarios are again rewritten in the Gherkin language. The test scenarios should include the business situation, data to be used and the acceptance criteria
- These scenarios or features should meet user acceptance criteria. For this, they are sent to end-users wherein they are evaluated by the end users, which gives them opportunity to validate the scenarios according to their needs
- The technical implementation of each scenario is created; however, it is kept abstracted from the users
- The test engineers create test scripts according to the business requirements
- In behavior driven development, the test scripts are bounded with the test scenarios to ensure that a single point remains point of contact



Using Cucumber and Selenium webdriver for BDD

The cucumber tool is extensively used in behavior driven development (BDD) for writing the acceptance tests. With this tool, the functional validation scripts can be written in an easy to read and understandable format. The cucumber feature file is a useful document, which can be used by business analysts, developers and end-users. With the cucumber tool, the test scripts are written in the 'Given, When and Then' structure. In this format, Given statements specifies the pre-conditions of the test case, When specifies the current conditions and Then statements specifies the outcomes under the given conditions.

In this format, the statements are written in simple language, eliminating the technical complexities of automation scripts. The selenium webdriver is a very commonly used API (Application Program Interface), in which the code can directly communicate with the web browser without the need of any manual intervention.

In order to write test cases using Cucumber Selenium Webdriver Framework, BDD steps are written in feature files. The testers need to generate a corresponding Step Definition File wherein all the functions are declared.

The Feature file contains the implementation process of Test Cases. All the steps abide by the format of Given, When and Then statements.

Scenario: Test Case to login into Email

Given - User navigate to the email homepage

When - Click on the email button

When - Enter email as test@gmail.com and Password as "testpassword" and click login button

Then - Successfully login into the mailbox

.

With reference to the feature file, a separate step definition class needs to be created to define functions declared in the feature file.

```
import cucumber.api.java.en.Given
public class StepDef {
    @Given("^Navigate to the loginpage$")
    public void loginPageNavigation() throws Throwable{
        // write technical code for the above code
    }
}
```

As per the file, each step of the feature file is technically coded. Each step is bound with the function declaration so that when we run the feature file, the bindings are automatically generated.

In the next step, we create a Java class, which can host all the functions declared in feature file steps. The Selenium code will be written in these applications. These codes will be performed on the web application.

```
Package Pages;

import org.openqa.selenium.*;

public class AutomationScript
{
    static WebDriver driver = new InternetExplorerDriver ();

    public static void navigateToHomepage ()
    {
        driver.navigate (). To ("https://appurl.com");
        driver.manage () .window () .maximize ();
    }
    public static void loginMethod(String name, String pw)
    {
        driver.findElement (By.id("username")). sendKeys (name);
        driver.findElement (By.id ("password")).sendKeys (pw);
        driver.findElement (By.id ("signin")).click ();
    }
}
```

The navigate functions is used to access any web URL. The selenium functions can be called using Step Definition file.

After running this code, the web browser will launch and navigate to the gmail homepage and enter the email and password.

Advantages of Selenium-cucumber automation framework

There are several benefits of the above mentioned selenium-cucumber framework, which are discussed below:

- In this framework, the feature file hides the coding, which makes it easier for the non-technical people to understand the code.
- As cucumber framework maps each and every step, the same code can be reused if similar step has to be performed
- If the test data changes, the tester need to make changes in only on the feature file. The selenium test scripts can be used as it is.
- Since cucumber and Selenium are open source tools, they can operate in a large variety of environment



About Test Triangle

Originally founded in 2012, Test Triangle has become a leader in IT consultancy services providing services in application testing, DevOps, RPA, Custom software development, mobile app development, Atlassian consultancy, niche IT staff augmentation and training in advanced technologies. Test Triangle is headquartered in Ireland; but it also has branch offices in London, United Kingdom, and Hyderabad, India. We have exponentially grown to become a team of 200+ members providing services in different verticals such as Banking & Finance, Utilities, Pharma, Retail, IT & Education etc.

Test Triangle's R&D department has created a propriety platform, Test Outsourcing Dashboard [TOD] which can be used to manage software testing lifecycle using collaboration tools like email, live chat, video conferencing. We have also launched a self- service testing platform (the premium version will be released as SaaS solution), which can provide a project overview and real-time updates of the software development lifecycle.

Over the years, we have established the reputation of being a 'trusted partner in IT consulting'. Test triangle is an agile software company, which constantly strives to exceed the expectations of its clients. We adopt the software testing and software application lifecycle to meet the customer's demand in an efficient and reliable manner. With a global workforce, we have proved ourselves in delivering tight-deadline projects.

We are proud to declare ourselves a client of Enterprise Ireland and European commission.



For inquiry please contact: inquiry@testtriangle.com

Ireland - HQ

Suite 12, Plaza 212 Blanchardstown Corporate Park,
Ballycoolen, Dublin, D15 W535

UK

4th floor, 86-90 Paul Street, London, EC2A 4NE

India

1-98/9/3, Plot No.3, Flat No.102, Jaihind Enclave,
Madhapur, Hyderabad 500 081

**Sales
Phone
Number**

ROI Hotline

+353 1 9685077

UK Hotline

+44 (0) 2071933020

India Hotline

+44 (0) 2071933020
+91 40 49510533



facebook.com/TestTriangle



linkedin.com/company/test-triangle



twitter.com/testtriangle



youtube.com/user/TestTriangle