

## Runner & Mapper Code Comparison

- Hardcoded Class path leads poor maintainability
- All Objects initialization instead of feature specific Object because of one to many mapping
- False Test Alarms during parallel test execution due to Test Data Conflicts among test scripts
- Additional Runner file for every feature along with step definition file

```
@CucumberOptions(  
    features = {"src/test/resources/com/sbux/ui/feature/ui_functional_test.feature"},  
    glue = {"com.sbux.ui.steps"},  
    tags = {"@SmokeTest"},  
    strict = true,  
    monochrome = true,  
    dryRun=false,  
    plugin = {  
        "pretty"  
    }  
)  
  
public class UiFunctionalTest extends TestInitializer {}
```

*O to M Runner (Test Case & Test Code Mapper) → Cucumber + TestNg + Java*

```
@RunWith(Cucumber.class)  
@CucumberOptions(  
    features = {"src/test/resources/com/sbux/ui/feature/ui_functional_test.feature"},  
    glue = {"com.sbux.ui.steps"},  
    tags = {"@SmokeTest"},  
    strict = true,  
    monochrome = true,  
    dryRun=false,  
    plugin = {  
        "com.sbux.ras.base.BaseSteps",  
        "io.qameta.allure.cucumber4jvm.AllureCucumber4Jvm",  
        "pretty"  
    }  
)  
  
public class UiFunctionalTest{}
```

*O to M Runner (Test Case & Test Code Mapper) → Cucumber + Junit + Java*

- Hardcoded Class path leads poor maintainability
- All Objects initialization instead of feature specific Object because of one to many mapping
- False Test Alarms during parallel test execution due to Test Data Conflicts among test scripts
- Additional Runner file for every feature along with step definition file

```
@RunWith(Cucumber::class)  
@CucumberOptions(  
    features = ["src/test/resources/cucumber/features"],  
    tags = ["not @ignored"],  
    strict =true,  
    plugin = ["pretty"]  
)  
  
class RunCucumberTest
```

*O to M Runner (Test Case & Test Code Mapper) → Cucumber + Junit + Kotlin*

- Better maintainability
- All Objects initialization with one to many mapping, along with feature specific Objects initialization with one to one mapping
- No False Test Alarms during parallel test execution due to Test Data Conflicts among test scripts
- Step definition file itself acts as Runner file

```
public class HomePageFunctionalTest extends StoryMapper {  
  
    @Override  
    public InjectableStepsFactory stepsFactory() {  
        return new InstanceStepsFactory(configuration(),  
            new BaseSteps(),  
            new HomePageSteps());  
    }  
}
```

*O to M, O to O Runner (Test Case & Test Code Mapper) → J-Behave + Junit + Java*

## Step Definition Code Comparison

```
@Given("^I open the browser")
public void openBrowser() throws Exception {
    driver.get(envParamBean.getWebappUrl());
}

@When("^I enter \"([^\"]*)\"$")
public void enterUser(String string) {
    logger.info("enter username");
}

@And("^I enter password \"([^\"]*)\"$")
public void enterPassword(String string) {
    logger.info("enter pass");
}
```

Step Definitions (Data Injection) → Cucumber + TestNg + Java

- Data Injection by regular expressions leads poor readability and high maintenance
- Additional code needed for parameters data Type conversion

```
@Given("I open the browser")
public void openBrowser() throws Exception {
    driver.get(envParamBean.getWebappUrl());
}

@Then("I enter {string}")
public void enterUser(String string) {
    logger.info("enter username");
}

@Then("I enter password {string}")
public void i_enterPassword(String string) {
    logger.info("enter pass");
}

@Then("I click login button")
public void clickLogin() {
    // navigateToCartPage();
    logger.info("click signin");
    Assert.assertTrue("Test Validation Failed", true);
}
```

Step Definitions (Data Injection) → Cucumber + Junit + Java

```
val alreadyHadThisManyCukes = 1
Given("^I have (\\d+) cukes in my belly$") { cukes: Long ->
    assertEquals(1, alreadyHadThisManyCukes)
    assertEquals(42L, cukes)
    val belly = Belly()
    belly.eat(cukes)
}

Given("something that is skipped") { }

When("^I wait (\\d+) hours$") { arg1: Int ->
    assertEquals(arg1, arg1)
}

When("A statement with a simple match", { assertTrue(true) })

val localInt = 1
Then("A statement with a scoped argument", { assertEquals(2, localInt + 1) })

val localState = "hello"
Then("I really have {int} cukes in my belly") { i: Int ->
    assertEquals(42, i)
    assertEquals("hello", localState)
}

Then("I will give you {int} and {float} and {word} and {int}") { a: Int, b: Float, c: String, d: Int ->
    assertEquals(1, a)
    assertEquals(2.2f, b)
    assertEquals("three", c)
    assertEquals(4, d)
}

@When("I login with $USER user")
@Step
public void login(@Named("USER") String user) {
    homePageHelper = new HomePageHelper(driver);
    homePageHelper.loginApplication();
}

@Given("I open the application")
@Step
public void openApp() throws Exception {
    driver.get(envParam.getWebappUrl());
}

@Then("I verify home page")
@Step
public void verify() throws Exception {
    logger.info("Verification");
    Thread.sleep(10000);
}
```

Step Definitions (Data Injection) → Cucumber + Junit + Kotlin

Step Definitions (Data Injection) → J-Behave + Junit + Java

- Data Injection by regular expressions leads poor readability and high maintenance
- Additional code needed for parameters data Type conversion
- Direct Data Injection with high readability and low maintenance
- Auto conversion of parameters for all data Types

## Console Output Comparison

```
07-23-2020 09:52:20.494-INFO-(WebDriverClient.java:22)-WebDriver initialized successfully
07-23-2020 09:52:22.893-INFO-(HomePageSteps.java:53)-enter username
07-23-2020 09:52:22.896-INFO-(HomePageSteps.java:58)-enter pass
07-23-2020 09:52:22.897-INFO-(HomePageSteps.java:64)-click signin
07-23-2020 09:52:22.910-INFO-(PropertiesUtil.java:30)-Properties file closed successfully
07-23-2020 09:52:24.227-INFO-(WebDriverClient.java:29)-WebDriver closed successfully
```

```
1 Scenarios (1 passed)
6 Steps (6 passed)
0m13.917s
```

```
PASSED: feature(Test allure English)
Runs Cucumber Feature
```

```
=====
Default test
Tests run: 1, Failures: 0, Skips: 0
=====
```

Console Output (Debugging Purpose) → Cucumber + TestNg + Java

Feature: CucumberJava

@SmokeTest

Scenario Outline: Login functionality exists # src/test/resources/com/sbux/ui/feature/ui\_functional\_test.feature:4

Given I open the browser  
Then I enter "test@yopmail.com"  
Then I enter password "naukri"  
Then I click login button

Examples:

Console Output (Debugging Purpose) → Cucumber + Junit + Java

@SmokeTest

Scenario Outline: Login functionality exists # src/test/resources/com/sbux/ui/feature/ui\_functional\_test.feature:12

07-23-2020 11:02:23.874-INFO-(PropertiesUtil.java:24)-Successfully loaded properties file:data/product.properties

Given I open the browser # HomePageSteps.openBrowser()  
07-23-2020 11:02:25.576-INFO-(HomePageSteps.java:45)-enter username  
Then I enter "test@yopmail.com" # HomePageSteps.enterUser(String)  
07-23-2020 11:02:25.580-INFO-(HomePageSteps.java:50)-enter pass  
Then I enter password "naukri" # HomePageSteps.i\_enterPassword(String)  
07-23-2020 11:02:25.581-INFO-(HomePageSteps.java:56)-click signin  
Then I click login button # HomePageSteps.clickLogin()  
07-23-2020 11:02:25.586-INFO-(PropertiesUtil.java:30)-Properties file closed successfully

07-23-2020 11:02:25.947-ERROR-(AllureLifecycle.java:609)-Could not add attachment: no test is running

07-23-2020 11:02:27.032-INFO-(WebDriverClient.java:32)-WebDriver closed successfully

Scenario: use the API with Java8 style @90m# src/test/resources/cucumber/features/kotlin.feature:3@0m

@32mGiven @0m@32mI have @0m@32m@1m42@0m@32m cukes in my belly@0m @90m#  
io.cucumber.kotlin.LambdaStepDefinitions.<init>(LambdaStepDefinitions.kt:46)@0m  
@32mWhen @0m@32mA statement with a simple match@0m @90m# io.cucumber.kotlin.LambdaStepDefinitions.<init>(LambdaStepDefinitions.kt:59)@0m  
@32mThen @0m@32mI really have @0m@32m@1m42@0m@32m cukes in my belly@0m @90m#  
io.cucumber.kotlin.LambdaStepDefinitions.<init>(LambdaStepDefinitions.kt:65)@0m  
@32mThen @0m@32mA statement with a scoped argument@0m @90m# io.cucumber.kotlin.LambdaStepDefinitions.<init>(LambdaStepDefinitions.kt:62)@0m

Scenario: use a table

@90m# src/test/resources/cucumber/features/kotlin.feature:10@0m

@32mGiven @0m@32mthis data table:@0m @90m# io.cucumber.kotlin.LambdaStepDefinitions.<init>(LambdaStepDefinitions.kt:39)@0m  
@32mThen @0m@32mI will give you @0m@32m@1m1@0m@32m and @0m@32m@1m2.2@0m@32m and @0m@32m@1mthree@0m@32m and @0m@32m@1m4@0m @90m#  
io.cucumber.kotlin.LambdaStepDefinitions.<init>(LambdaStepDefinitions.kt:70)@0m

2 Scenarios (@32m2 passed@0m)

6 Steps (@32m6 passed@0m)

0m0.765s

Console Output (Debugging Purpose) → J:Behave + Junit + Kotlin

(BeforeStories)

07-23-2020 09:57:15.574-INFO-(TestInitializer.java:123)-Logger initialized successfully

Running story com/sbux/ras/test/home\_page\_functional\_test.story

Verify application home page functionality

@Ref Story - https://sim.starbucks.com/browse/RTEG-4230

(com/sbux/ras/test/home\_page\_functional\_test.story)

Meta:

@features rasui

Scenario: (1-MPTS) Open Home Page

Meta:

@categories bat reg

Examples:

Given I open the application  
When I login with <USER> user  
Then I verify home page

|USER|KEY|  
|VALID|Y|

Example: {USER=VALID, KEY=Y}

07-23-2020 09:57:28.049-INFO-(TestInitializer.java:94)-WebDriver initialized successfully

Given I open the application

When I login with VALID user

07-23-2020 09:57:29.256-INFO-(HomePageFunctionalTest.java:36)-Verification

Then I verify home page

Console Output (Debugging Purpose) → J:Behave + Junit + Java

## Library Comparison

### *Cucumber-jvm*

- *Reference: <https://mvnrepository.com/artifact/io.cucumber/cucumber-junit>*
- *First Release: Aug 2017*
- *Latest Release: July 2020*
- *Current Release frequency (Stability) : 10 to 15 days*
- *Total Releases: 65 in 35 months*
- *License: MIT*

### *J-Behave*

- *Reference: <https://mvnrepository.com/artifact/org.jbehave/jbehave-core>*
- *First Release: Sept 2008*
- *Latest Release: Apr 2020*
- *Current Release frequency (Stability) : 3 to 4 Months*
- *Total Releases: 160 in 139 months*
- *License: BSD*