**Task #5**

## ****What is Behavior Driven Testing?****

Behavior Driven Testing (BDT) is an uncommon term in software testing/ development compared to Behavior Driven Development (BDD). Tests in Behavior driven testing are most often focused on the behavior of users rather than the technical functions of the software. BDT is the best choice when you want to show your business point of view and the requirements of the product. To spread out the product ideas, BTD uses very easily understandable language to make sure everyone in the team does not miss anything, even they are non-technical guys. Hence people who are in charge for business analysis and product management could join actively the testing process. Behavior Driven Testing is gaining momentum as lots of organizations are looking at it as a solution to their automation and collaboration challenges. BDT framework has been successfully implemented by various QA teams in various domains. Furthermore, it is the premise for implementing [Data-driven testing](https://docs.katalon.com/katalon-studio/docs/test-case-variables.html" \t "https://katalon.com/sa/behavior-driven-testing/_blank) much more successfully and easily.

## ****Why Behavior Driven Testing originated?****

Since the time test automation started, it has evolved tremendously with new concepts, framework design, and tools. The low usability, rigidity and high maintenance cost has been the reason of evolution of automation frameworks. Evolution of Test frameworks: **Core frameworks** > **Record and Playback** > **Data-driven** > **Keyword driven** > **Hybrid**> **Behaviour driven testing.** One of the common downsides when using these frameworks is that It is hard for non-technical people like Business analysts, management people to read and understand the test cases. But Behaviour driven testing can handle it. It has helped to fill the communication gap by writing the test cases in natural language. It is a relatively new agile software development approach that focuses on communication; encouraging collaboration between developers, QA and business participants; to help bridge the Business-IT alignment gap. The scenarios are written to build up a clear understanding of the desired behavior and this happens through discussion with stakeholders.

## ****Advantages of Behavior Driven Testing****

* **Collaboration:** It builds closer relationships among business Analysts, QA, and Dev through the Behaviour Clarification Reviews. It’s also helpful for communicating the requirements effectively. Verified by BAs and testers to make sure of the acceptance criteria.
* **Easy Review and Feedback:** By writing the test cases in natural language which is human-readable, no development skills will be required. Business Analysts can actively participate in the process of creating the automated test cases, review and give their feedback to improve them.
* **Right Focus:** BDT helps to focus on the requirements and the business point of view rather than focusing too much on testing the technical implementation.
* **Greater ROI:** It has been observed that the behavior of the software stays for longer than the implementation. So, behavior driven tests are easier to modify and hence results in greater ROI. Also, the BDT tools are open source which further reduces the investment.
* **No longer costly to maintain:** With BDT code redundancy can be minimized due to ‘step reusability’ feature. Hence, these tests are less costly to maintain.

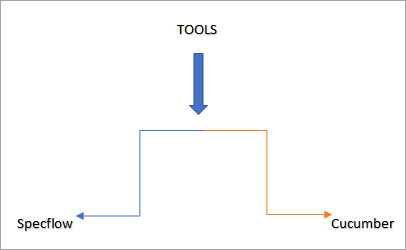
## ****How Behavior Driven Testing implemented?****

There is a plethora of open source BDT frameworks available in many programming languages and supporting different platforms. For instance, Cucumber (preferably with Ruby), JBehave for Java, NBehave and specflow for .Net, Behat for PHP and Twist for Groovy etc. The BDT layer is implemented on top of customized framework so it’s about building up layer upon layer only. The behavior driven tests are given- when-then style steps written in natural language. Of course, the implementation of these steps has to be done in a programming language of one’s choice. Below is the outline of Behavior Driven Testing process:

* Business Analyst writes a formal User story.
* Tester writes the scenarios based on the user story.
* Scenarios get reviewed by the business team.
* Automation engineer writes the step definitions for the scenario steps.
* Scenarios get automated by the QA team.
* Run the automated scenarios and fix if any bugs exist.
* Automated tests run as regression tests.
* User accepts the software (Acceptance and Quality Criteria met)

### **How to Implement the BDD Approach?**

Test scenarios should be written in plain language with a detailed description of the test, how to test the application and the application behavior which can be understandable by all.

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2019/07/BDD-approach.png)

In this tutorial, we will focus on [Cucumber](https://cucumber.io/" \t "https://www.softwaretestinghelp.com/bdd-framework/_blank) – a software tool for BDD and will learn to practically implement it using its language i.e. Gherkin.

## **Cucumber – A BDD Framework Tool**

Cucumber is a Behavior Driven Development (BDD) framework tool to write test cases.

### **Given – When – Then Approach**

* ****Given:**** Some given context (Preconditions).
* ****When:****Some Action is performed (Actions).
* ****Then:****Particular outcome/consequence after the above step (Results)..