**WHAT ARE ACCEPTANCE CRITERIA?**

Acceptance Criteria are the conditions that a software product must satisfy to be accepted by a user, customer, or in the case of system level functionality, the consuming system.

Acceptance Criteria are a set of statements, each with a clear pass/fail result, that specify both functional and non-functional requirements, and are applicable at the Epic, Feature, and Story Level. Acceptance criteria constitute our “Definition of Done”, and by done I mean well done.

The GWT (Given-When-Then) format is helpful way to specify criteria:

*Given some precondition When I do some action Then I expects some result*

When writing acceptance criteria in this format, it provides a consistent structure. Additionally, it helps testers determine when to begin and end testing for that specific work item.

**Discuss the Requirements**

During the Planning Meeting in which we discuss the story about secure passwords, we ask the

Business stakeholder requesting the feature questions intended to elicit acceptance criteria:

* *“What should precondition for validate the Secure and Insecure passwords to create user account?”*

Given User should on login page with username and password text fields

When user enters passwords valid and Invalid

* *“What should happen if a user enters a secure password?”*

Then success message “Successfully user account is created” should be displayed

* *“What should happen if a user enters an insecure password and clicked on button “Create Account”?*

Then error message should be displayed

*Example:* Passwords must be at least 6 characters long and contain at least one letter, one number

* *“Can you give me examples of passwords you consider secure and insecure?”*

*Examples:*

*Secure:* “p@ssw0rd”, “@@@000dd”, “p@ss w0rd”,“p@sw0d”

*Insecure:* “password”,“p@ss3”, “passw0rd”, “p@ssword”, “@@@000”

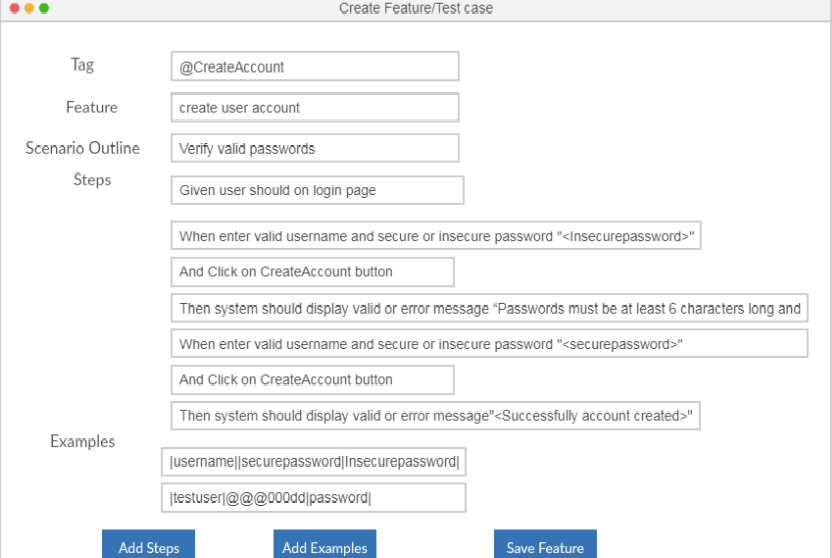
**Advantages:**

1. No confusion on test cases as review is not required as the test cases captured in acceptance criteria in the form of feature files

It will save 50 to 60 % of effort compare to the general practice of writing and reviewing test case

**Distill Tests in a Framework-Friendly Format**

We can capture the tests in a format that defined in acceptance criteria and works with our test EliteRobotics framework.



**Develop the Code (and Hook up the Tests)**

Developers test first approach (TDD) by using script less EliteRobotic automation frameowrk

1. Add the selenium script to perform actions on application by selecting action methods like Click,sendkeys..etc with display name of the object in step definitions

Example:

Allscripts.webinteractions.**Click (“pagename.Displayname”)**

Execute the acceptance tests and watch them fail. In this case, the tests will fail with error messages like

the following from EliteRobotics Framework:

Web element “username“not found

1. Capture element properties using Firepath macro for the objects by defining the **pagename.Displayname** for the implemented object

Execute the acceptance tests and implemented objects test cases should pass

**Advantages:**

1. We can finish the 90% of automation without application by using Elite Robotics framework
2. With less guidance developer also automate the test case by using EliteRobotics framework

**Elite Robotics framework**

Elite Robotics framework is developed to meet the BDD approach and it is common plat form to run all kind of test cases like Unit, web services, mobile applications and web applications, which is integrated with the tools Cucumber, Selenium Web driver, Appium, Apache http client.