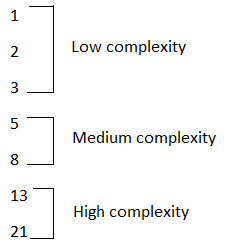
1. **What is effort in agile?**

* What you have invest to get task done. It can be measured either in units of time (e.g. Man hours) or in more abstract units (e.g. effort points)

1. **On what basis you will give story points?**

* The story point is a quantitative measure of complexity in implementing user story.

Story points are always given in terms of Fibonacci series.



1. **Differences between Assert & Verify and diff between Soft Assert & Hard Assert?**

* **Assert:** If the assert condition is true then the program control will execute the next test step but if the condition is false, the execution will stop and further test step will not be executed.  
  whereas,  
  **Verify:** There won’t be any halt in the test execution even though the verify condition is true or false.

Example: -

When an assert statement fails then all the other test case after that won't execute. This is a problem, but again its good practice to use try catch block to avoid this situation.

Verify is used in less critical things. Cases where we can move forward even if the other test cases fails.

* **Hard Assert** – Hard Assert throws an *AssertException* immediately when an assert statement fails and test suite continues with next *@Test*

The disadvantage of Hard Assert – It marks method as fail if assert condition gets failed and the remaining statements inside the method will be aborted.

To overcome this, we need to use Soft Assert. Let’s see what is Soft Assert.

**Soft Assert** – Soft Assert collects errors during *@Test*. Soft Assert does not throw an exception when an assert fails and would continue with the next step after the assert statement.

If there is any exception and you want to throw it then you need to use *assertAll()* method as a last statement in the @Test and test suite again continue with next *@Test* as it is.

We need to create an object to use Soft Assert which is not needed in Hard Assert.

1. **What are the roles in agile and scrum consists of how many teams?**

* Product owner, Srum master and team members (including developers, testers, database handlers, support engineers).
* Team size is: 7 +/- 2

1. **What are desired capabilities and what is the use?**

* It is a class in org.openqa.selenium.remote.DesiredCapabilities package.
* It gives facility to set the properties of browser. Such as to set BrowserName, Platform, Version of Browser.
* Mostly DesiredCapabilities class used when do we used Selenium Grid.
* We have to execute multiple TestCases on multiple Systems with different browser with Different version and Different Operating System.

Example:

WebDriver driver;

String baseUrl, nodeUrl;

baseUrl = "https://www.facebook.com";

nodeUrl = "http://192.168.10.21:5568/wd/hub";

DesiredCapabilities capability = DesiredCapabilities.firefox();

capability.setBrowserName("firefox");

capability.setPlatform(Platform.WIN8\_1);

driver = new RemoteWebDriver(new URL(nodeUrl), capability);

driver.manage().window().maximize();

driver.manage().timeouts().implicitlyWait(2, TimeUnit.MINUTES);

1. **How do we get text of a link?**

* If getText() returns an empty String, try the innerHTML attribute:  
  String text = element.getAttribute("innerHTML");

1. **What is fluent wait in selenium and what is the use?**

* **Implicit Wait:**

An implicit wait is to tell WebDriver to poll the DOM for a certain amount of time when trying to

find an element or elements if they are not immediately available. The default setting is 0. Once set,

the implicit wait is set for the life of the WebDriver object instance.

WebDriver driver = new FirefoxDriver();

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

driver.get("http://somedomain/url\_that\_delays\_loading");

WebElement myDynamicElement = driver.findElement(By.id("myDynamicElement"));

**When to use:** Not recommended

* **Explicit wait:**

An explicit wait is code you define to wait for a certain condition to occur before proceeding further in the code. WebDriverWait by default calls the ExpectedCondition every 500 milliseconds until it returns successfully.

WebDriver driver = new FirefoxDriver();

driver.get("http://somedomain/url\_that\_delays\_loading");

WebElement myDynamicElement = (new WebDriverWait(driver, 10))

.until(ExpectedConditions.presenceOfElementLocated(By.id("myDynamicElement")));

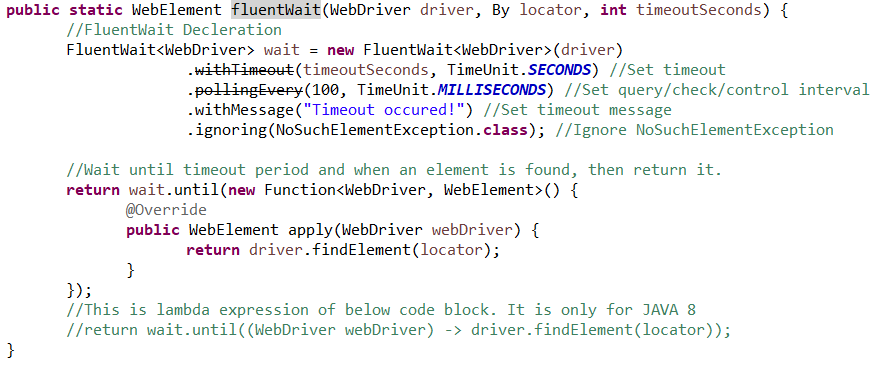
**When to use:** If element takes a long time to load. Also, used to check property of an element (presence, clickability. etc).

* **FluentWait**:

For each FluentWait instance, you can specify:

1. Frequency with which FluentWait has to check the conditions defined.
2. Ignore specific types of exception waiting such as NoSuchElementExceptions while searching for an element on the page.
3. Maximum amount of time to wait for a condition

**When to use FluentWait**: When you try to test the presence of an element that may appear after every x seconds/minutes (Just an example, this is my guess of where such a thing can be used).

****

1. **What is same origin policy in selenium?**

* First of all *“****Same Origin Policy****”* is introduced for security reason, and it ensures that content of your site will never be accessible by a script from another site. As per the policy, any code loaded within the browser can only operate within that website’s domain.
* ***Same Origin policy*** prohibits JavaScript code from accessing elements from a domain that is different from where it was launched. Example, the HTML code in www.google.com uses a JavaScript program **"testScript.js"**. The ***same origin policy*** will only allow **testScript.js** to access pages within **google.com** such as **google.com/mail, google.com/login,** or **google.com/signup**. However, it cannot access pages from different sites such as **yahoo.com/search or fbk.com** because they belong to different domains.
* This is the reason why prior to Selenium RC, testers needed to install local copies of both Selenium Core (a JavaScript program) and the web server containing the web application being tested so they would belong to the same domain.
* To avoid ***“Same Origin Policy”*** proxy injection method is used, in proxy injection mode the **Selenium Server** acts as a client configured HTTP proxy, which sits between the browser and application under test and then masks the **AUT** under a fictional URL.
* If you are referring to selenium webdriver and not selenium RC, then the answer is you don’t have to worry about same origin policy in case of webdriver since each browser has its own webdriver. This is the whole advantage of webdriver as opposed to RC i.e no selenium core injection into the browser and no middleware client server between the browser and AUT. Webdriver provides a native OS level support in controlling the browser automation.

1. **What is set speed?**

* Thread.sleep() will stop the current (java) thread for the specified amount of time. It's done only once.
* Selenium.setSpeed() will stop the execution for the specified amount of time for **every** selenium command. It is useful for demonstration purpose (you will see thing moving in your browser) or if you are using a slow web application (there are better technique to handle slow applications but that's off topic.)
* Thread.sleep(2000);

Operation 1

Thread.sleep(2000);

Operation 2

Thread.sleep(2000);

Operation 3

* selenium.setSpeed("2000");

Operation 1

Operation 2

Operation 3

1. **What is the main use of throws keyword in java?**

* **throws keyword in java**

1. throws clause is used to declare an exception, which means it works similar to the try-catch block.
2. throws is followed by exception class names.

Ex: throws Exception:

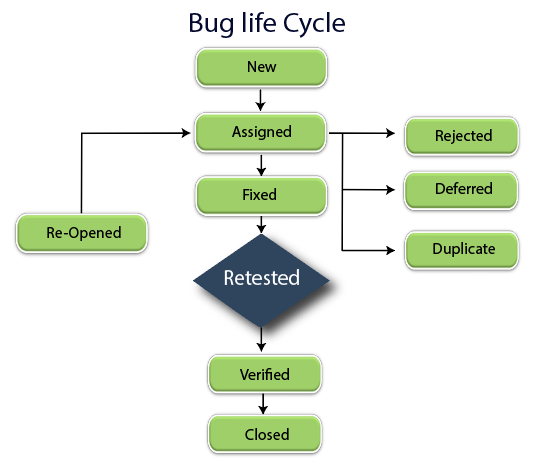
1. throws is used in method signature to declare the exceptions that can occur in the statements present in the method.
2. you can handle multiple exceptions by declaring them using throws keyword.

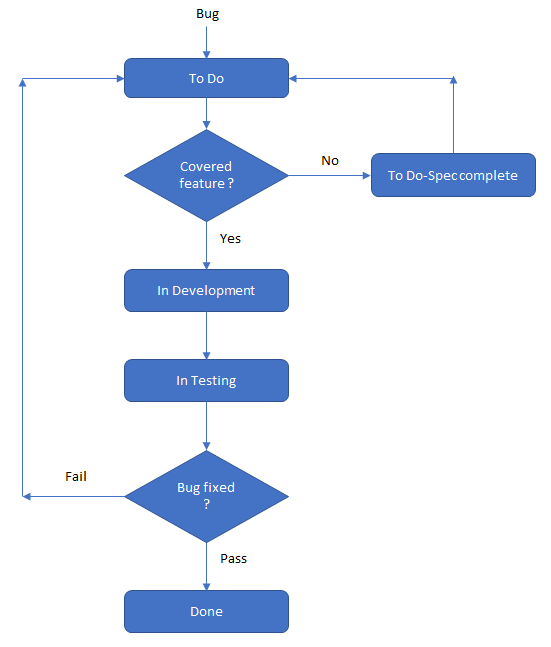
* **throw keyword in java**

1. throw keyword is used to throw an exception explicitly.
2. throw is followed by an instance of Exception class

Ex: - throw new Exception(“Error divide by zero”);

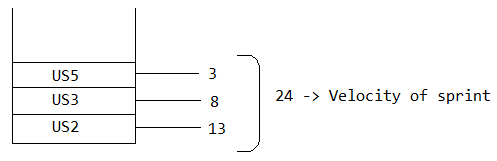
1. throw keyword is used within the method body to throw an exception
2. You can throw one exception at a time
3. **What is the defect life cycle in java?**



****

1. **What is velocity in agile and how do we measure?**

* The sum of all the story points of user stories present in sprint backlog in called as velocity of sprint.



1. **What are the different processes in agile?**

1) Scrum

2) Kanban

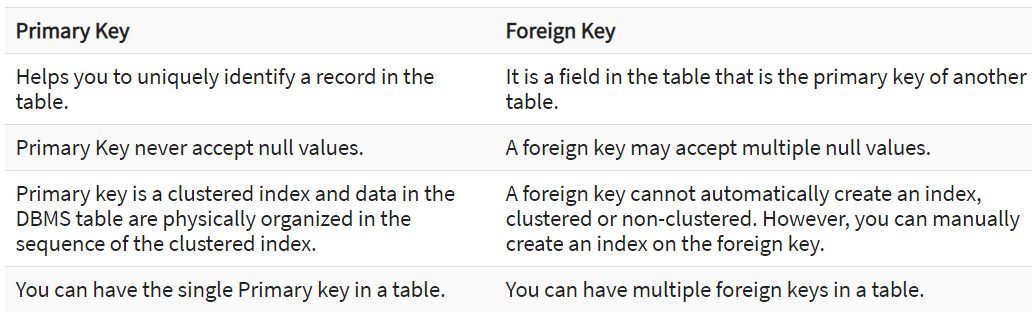
3) Extreme Programming (XP)

4) Learn Development

5) Crystal

1. **Diff between primary key and foreign key in MySQL?**





1. **Diff between truncate and delete?**

* **DELETE:**

DELETE is a DML command

DELETE you can rollback

Delete = Only Delete- so it can be rolled back

In DELETE you can write conditions using WHERE clause

Syntax – Delete from [Table] where [Condition]

* **TRUNCATE:**

TRUNCATE is a DDL command

You can't rollback in TRUNCATE, TRUNCATE removes the record permanently

Truncate = Delete+Commit -so we can't roll back

You can't use conditions(WHERE clause) in TRUNCATE

Syntax – Truncate table [Table]