EXIT CRITERIA

TDD vs BDD

|  |  |  |
| --- | --- | --- |
| S.No | TDD | BDD |
| 1 | Stands for Test Driven Development | Stands for Behavior Driven Development |
| 2 | The process starts by writing a test case | The process starts by writing a scenario in simple english so that product owner can understand it too. |
| 3 | Collaboration is required only between the developers | Collaboration is required between all the stakeholders. |
| 4 | Some of the tools which support TDD are: JUnit, TestNG, NUnit, etc. | Some of the tools which support BDD are SpecFlow, Cucumber, MSpec, etc. |
| 5 | Tests in TDD can only be understood by people with programming knowledge | Tests in BDD can be understood by any person including the ones without any programming knowledge. |
| 6 | Not much involvement of product owner. | Product owner centric, have concept of acceptance tesing. |
| 7 | Transparency factor is missing | Transparency between user expectations and developer tests. |

**Scenario:** Login check

**Given** I am on the login page

**And** I click on the "Login" button

**Then** I get an error to enter username

Software development life cycle

Requirement- design- development- testing – deployment – maintenance



<https://www.softwaretestinghelp.com/software-development-life-cycle-sdlc/>

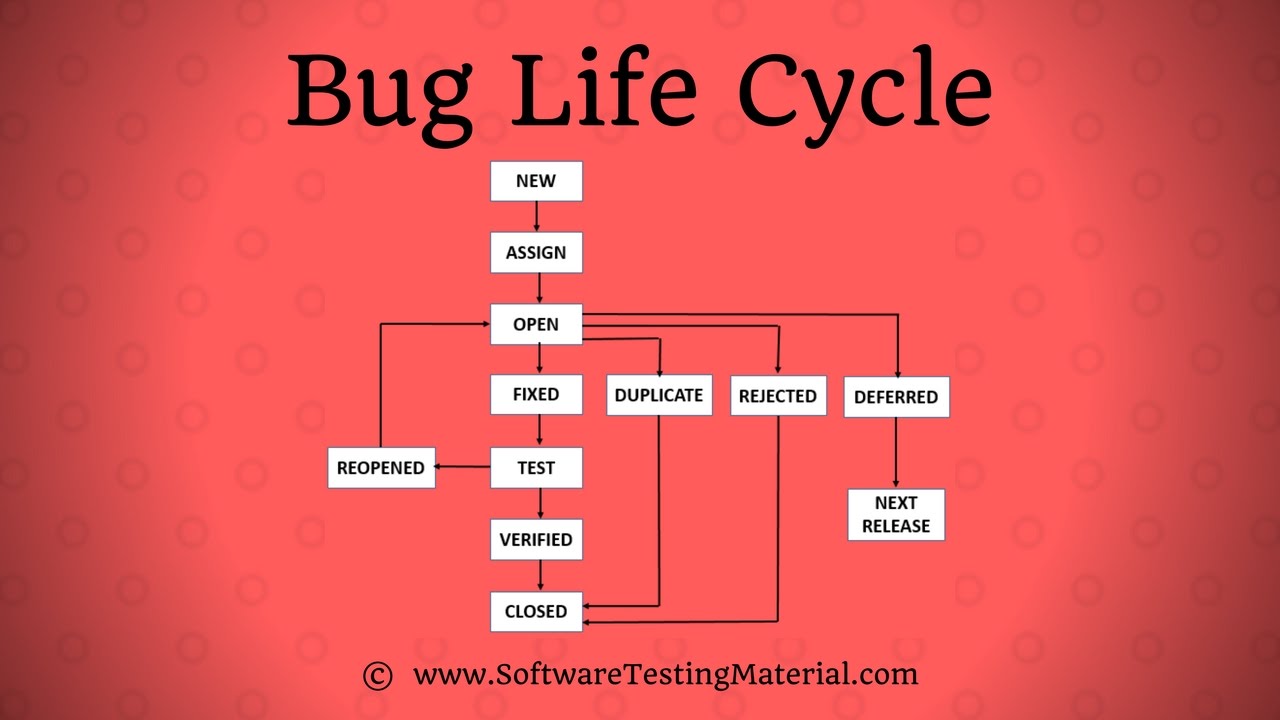
Software test life cycle

Requirement – test planning – test case development- enviromant setup-test execution – test closure



<https://www.guru99.com/software-testing-life-cycle.html>

**Bug life cycle:**



**Agile Vs Waterfall Method**

|  |  |
| --- | --- |
| Agile Model | Waterfall Model |
| Agile method proposes incremental and iterative approach to software design | Development of the software flows sequentially from start point to end point. |
| The agile process is broken into individual models that designers work on | The design process is not broken into an individual models |
| The customer has early and frequent opportunities to look at the product and make decision and changes to the project | The customer can only see the product at the end of the project |
| Agile model is considered unstructured compared to the waterfall model | Waterfall model are more secure because they are so plan oriented |
| Small projects can be implemented very quickly. For large projects, it is difficult to estimate the development time. | All sorts of project can be estimated and completed. |
| Error can be fixed in the middle of the project. | Only at the end, the whole product is tested. If the requirement error is found or any changes have to be made, the project has to start from the beginning |
| Documentation attends less priority than software development | Documentation is a top priority and can even use for training staff and upgrade the software with another team |
| Every iteration has its own testing phase. It allows implementing regression testing every time new functions or logic are released. | Only after the development phase, the testing phase is executed because separate parts are not fully functional. |
| Testers and developers work together | Testers work separately from developers |
| At the end of every sprint, user acceptance is performed | User acceptance is **performed** at the end of the project. |
|  |  |

What is Scrum?

Scrum is a simple framework for effective team collaboration on complex products.

### Scrum Artifacts

* [Product Backlog](https://www.scrum.org/resources/what-is-a-product-backlog)
* [Sprint Backlog](https://www.scrum.org/resources/what-is-a-sprint-backlog)
* [Increment](https://www.scrum.org/resources/what-is-an-increment)

Product backlog: contains all the requirement

Sprint backlog : Product Backlog items selected for the Sprint.

Increment: sum of all the [Product Backlog](https://www.scrum.org/resources/what-is-a-product-backlog) items completed during a Sprint.

[Daily Scrum](https://www.scrum.org/resources/what-is-a-daily-scrum): Meeting to synchronize activities and create a plan for the next 24 hours.

**Product owner**: client is not product owner

Client give his requirements to a person and that person give us the requirement so that person is a product owner

**Scrum master**: manage the scrum meetup, helping people to achieve task.

**Scrum Team**: QA, devs etc.

### Different Types Of Software Testing

Given below is the list of some common types of Software Testing:

**Functional Testing types include:**

* Unit Testing

Testing of an individual software component or module is termed as [Unit Testing](https://www.softwaretestinghelp.com/unit-testing/).

* Integration Testing

verify the combined functionality after integration of few modules

* System Testing

the entire system is tested as per the requirements.

* Sanity Testing

Testing technique which determines if a new software version is performing well enough to accept it for a major testing effort

* Smoke Testing

The result of this **testing** is used to decide if a build is stable enough to proceed with further **testing**.

* Interface Testing
* **Regression Testing**

It is done when changes to the program (e.g. new functionality) have been made.

**Retesting**

**Retesting** is **testing** of a particular bug after it has been fixed.

* Beta Testing

It is typically done by a small group of end-users.

* Acceptance Testing

It is usually performed by the customer, determine whether or not a system satisfies its acceptance criteria.

**Non-functional Testing types include:**

* Performance Testing
* Load Testing
* Stress Testing
* Volume Testing
* Security Testing
* Compatibility Testing
* Install Testing
* Recovery Testing
* Reliability Testing
* Usability Testing
* Compliance Testing
* Localization Testing

exhaustive testing

integration testing:

**Integration testing** (sometimes called **integration and testing**, abbreviated **I&T**) is the phase in [software testing](https://en.wikipedia.org/wiki/Software_testing) in which individual software modules are combined and tested as a group.

Load and stress testing

**Load testing** is a type of non-functional testing. A load test is type of software testing which is conducted to understand the behaviour of the application **under a specific expected load.**

**Stress Testing:**

[Stress testing](https://en.wikipedia.org/wiki/Stress_testing) is normally used to understand the upper limits of capacity within the system

Performance testing:

**performance testing** is in general a [testing](https://en.wikipedia.org/wiki/Software_testing) practice performed to determine how a [system](https://en.wikipedia.org/wiki/System) performs in terms of responsiveness and stability under a particular workload.

|  |  |
| --- | --- |
| Verification | Validation |
| Are we building the product right? | Are we building the right product? |
| The verifying process includes checking documents, design. | It is a dynamic mechanism of testing and validating the actual product |
| It does not involve executing the code | It always involves executing the code |
| It comes before validation | It comes after verification |

**Bug leakage :**

A defect which exists during testing yet unfound by the tester which is eventually found by the tester/end-user is also called bug leakage.

**bug release:**

A **bug release** is when a particular version of s/w is **released** with a set of known **bug**(s)/**defect**(s). These **bugs** are usually low severity and/or low priority **bugs**.

**Testcase techniques**

Boundary Value Analysis (BVA)

Decision Table Testing

Static and dynamic testing

**Static testing** does the verification process while **Dynamic testing** does the validation process

**Static testing** is about the prevention of defects whereas **Dynamic testing** is about finding and fixing the defects.

Latent defect

**Latent Defect** is one which has been in the system for a long time; but is discovered now. i.e. a **defect** which has been there for a long time and should have been detected earlier is known as **Latent Defect**

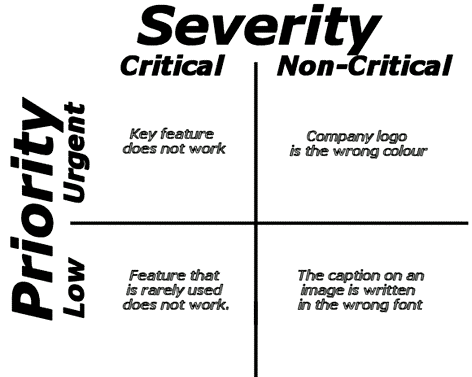
Move to a different elements after hover

Book a flight 100 times in 100 tabs using testing threads?

**Severity vs priority**

**Severity** means how severe defect is affecting the functionality.

**Priority** means how fast defect has to be fixed.



**High Priority, High Severity :-** An error which occurs on the basic functionality of the application and will not allow the user to use the system (E.g. user is not able to login to the application)

**High Priority, Low Severity bug :-** If the company name is misspelled in the home page of the website,then the priority is high and severity is low to fix it.

**High Severity, Low Priority :-** Web page not found when user clicks on a link (user&#39;s does not visit that page generally)ss

**Low Priority, Low Severity :-** Any cosmetic or spelling issues which is within a paragraph or in the report

**Logic behind amazon logo:**

The **Amazon logo** was created to represent the message that it sells everything from A to Z

## Apache POI?

**Apache POI** (Poor Obfuscation Implementation) is a Java API for reading and writing MS Office files using Java programs.

* **HSSF** (Horrible Spreadsheet Format) − It is used to read and write **xls** format of MS-Excel files.
* **XSSF** (XML Spreadsheet Format) − It is used for **xlsx** file format of MS-Excel.

**Smoke Testing:** It is the initial testing process exercised to check weather the software under test is ready/stable for further testing. It is also known as build acceptance testing.

In this we check whole system but not regressly/check app at high level.

Smoke testing is usually documented or scripted

**Sanity Testing:** check apart of an application.

It isthesubset of regression testing.

Check bug fixed working perfectly or not.

Check the main features of app at high level.

**Regression testing:** Repeated testing of an already tested program, after modification.

We do regression testing when:

* New functionalities added to the application
* Defect fixing

**Retesting:** To ensure that the defect which were found and posted in the earlier build was fixed or not in the current build.

Say, Build 1.0 is released.

Test team found some defects(say defect id 1.0.1, 1.0.2) and posted.

# Heaven Or Hell Puzzle

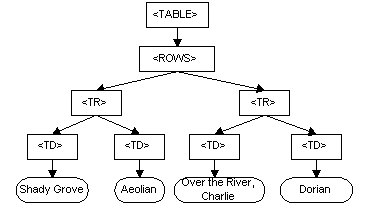
**Me saamne vale se puchta hu ki ‘bhai agr me saamne vale se puchunga to vo kisko heaven bolega’**

“If I ask the other guard about which side leads to heaven, what would he answer?”.

**Testcases of lift:**

* Verify the dimensions of the lift
* Verify the capacity of the lift in terms of the total weight
* Verify the door open and close as number of button pressed.
* Verify that lift moves to the particular floor as the button of the floor is clicked
* Verify if there is an emergency button to contact officials in case of any mishap
* Verify the performance of the floor – the time is taken to go to a floor
* Verify that in case of power failure, lift doesn’t free-fall and get halted in the particular floor
* Verify lifts working in case button to open the door is pressed before reaching the destination floor
* Observe if door open back if we put obstacle after press close button.

Document Object Model (**DOM**):



### What is Entry and Exit Criteria?

**Entry Criteria:** Entry Criteria gives the prerequisite items that must be completed before testing can begin.

Ex:

* Appropriately defined and approved requirements
* Access to sufficient and desired test data
* Setting up of test environment with all the necessary resources like tools and devices

**Exit Criteria:** Exit Criteria defines the items that must be completed before testing can be concluded

Ex:

* Ensuring all critical Test Cases are passed
* Identifying and fixing all the high-priority defects