**Testing**:

Testing is to validate the software/application/module is behaving in expected manner by meeting all the business requirements.

**Manual Testing:**

It is a process which is carried out by manual testers (Quality Analyst) in order to find defects (bug) without use of any automation tools.

**Testing Techniques:**

->Unit Testing (Dev Team)

->System testing (QA Team)

->Integration Testing (QA Team)

->Acceptance Testing (QA Team)

->Black Box Testing (QA Team)

->White Box Testing (Dev Team)

**Unit Testing**: Testing the peace of the code in developers’ machine to validate whether that particular functionality is working fine is called Unity Testing.

**Note: Unit Testing is performed by the developers.**

**System Testing**: Testing the critical functionality that validates the complete and fully integrated software product. Here testers will be validating end to end system specifications /entire flow.

**Types of system Testing:**

->Functional Testing

->usability testing

->Regression Testing

->Load Testing

->Migration Testing

**Integration Testing:** Testing the software when once functionality/ module is integrated with another module. Here Testers are going to validate the integration part.

**Acceptance Testing:** Testing Technique performed to determine whether or not the software has met the requirement specifications. The main objective is to evaluate the system compliance with the business requirements and verify it has met the required Acceptance criteria to deliver it to the end users.

**Black Box Testing:** It is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. It is mainly focussing on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as behavioural testing.

**Types of Black Box testing Techniques:**

**ECP (Equivalence Class Partition)**

**Scenario: Testing the values from 1 rs to 1 crore rs**

Divide the given amount into 10 Equivalence Classes

1rs to 10 lakhs -> 3 ,36, 764, 8456,37566,362345 ,986245

10 lakhs to 20 lakhs ->12,37,445,

20 lakhs to 30 lakhs ->23,45,985,

30 lakhs to 40 lakhs ->

40 lakhs to 50 lakhs ->

50 lakhs to 60 lakhs ->

60 lakhs to 70 lakhs ->

70 lakhs to 80 lakhs ->

80 lakhs to 90 lakhs ->

90 lakhs to 1 crore ->

**BVP: Boundary Values Analysis:**

1rs to 10 lakhs -> 99 paise, 95 paise, 0.9, 0.85, 999999.8,99999,1.5,99,999.5

10 lakhs to 20 lakhs ->10,0000.50, 19,99,999, 19,99,999.5,

20 lakhs to 30 lakhs ->

30 lakhs to 40 lakhs ->

40 lakhs to 50 lakhs ->

50 lakhs to 60 lakhs ->

60 lakhs to 70 lakhs ->

70 lakhs to 80 lakhs ->

80 lakhs to 90 lakhs ->

90 lakhs to 1 crore ->

**White Box Testing:** Testing performed by dev team who have knowledge on internal functionality of the software / application. Testers will have a limited knowledge on this.

| Black Box Testing | White Box Testing |
| --- | --- |
| It is a way of software testing in which the internal structure or the program or the code is hidden and nothing is known about it. | It is a way of testing the software in which the tester has knowledge about the internal structure or the code or the program of the software. |
| It is mostly done by software testers. | It is mostly done by software developers. |
| No knowledge of implementation is needed. | Knowledge of implementation is required. |
| It can be referred as outer or external software testing. | It is the inner or the internal software testing. |
| It is functional test of the software. | It is structural test of the software. |
| This testing can be initiated on the basis of requirement specifications document. | This type of testing of software is started after detail design document. |
| No knowledge of programming is required. | It is mandatory to have knowledge of programming. |
| It is the behavior testing of the software. | It is the logic testing of the software. |
| It is applicable to the higher levels of testing of software. | It is generally applicable to the lower levels of software testing. |
| It is also called closed testing. | It is also called as clear box testing. |
| It is least time consuming. | It is most time consuming. |
| It is not suitable or preferred for algorithm testing. | It is suitable for algorithm testing. |
| Can be done by trial and error ways and methods. | Data domains along with inner or internal boundaries can be better tested. |
| **Example:** search something on google by using keywords | **Example:** by input to check and verify loops |
| **Types of Black Box Testing:**    * A. Functional Testing * B. Non-functional testing * C. Regression Testing | **Types of White Box Testing:**    * A. Path Testing * B. Loop Testing * C. Condition testing |

**-----------------------------------------------------------------------------------------------------------**

**Smoke Testing**: **Testing the critical functionality of the software / application at the initial phases until the application becomes stable.**

**V0.0 -> introducing the login functionality**

**V0.1 ->UI looks and modules displayed**

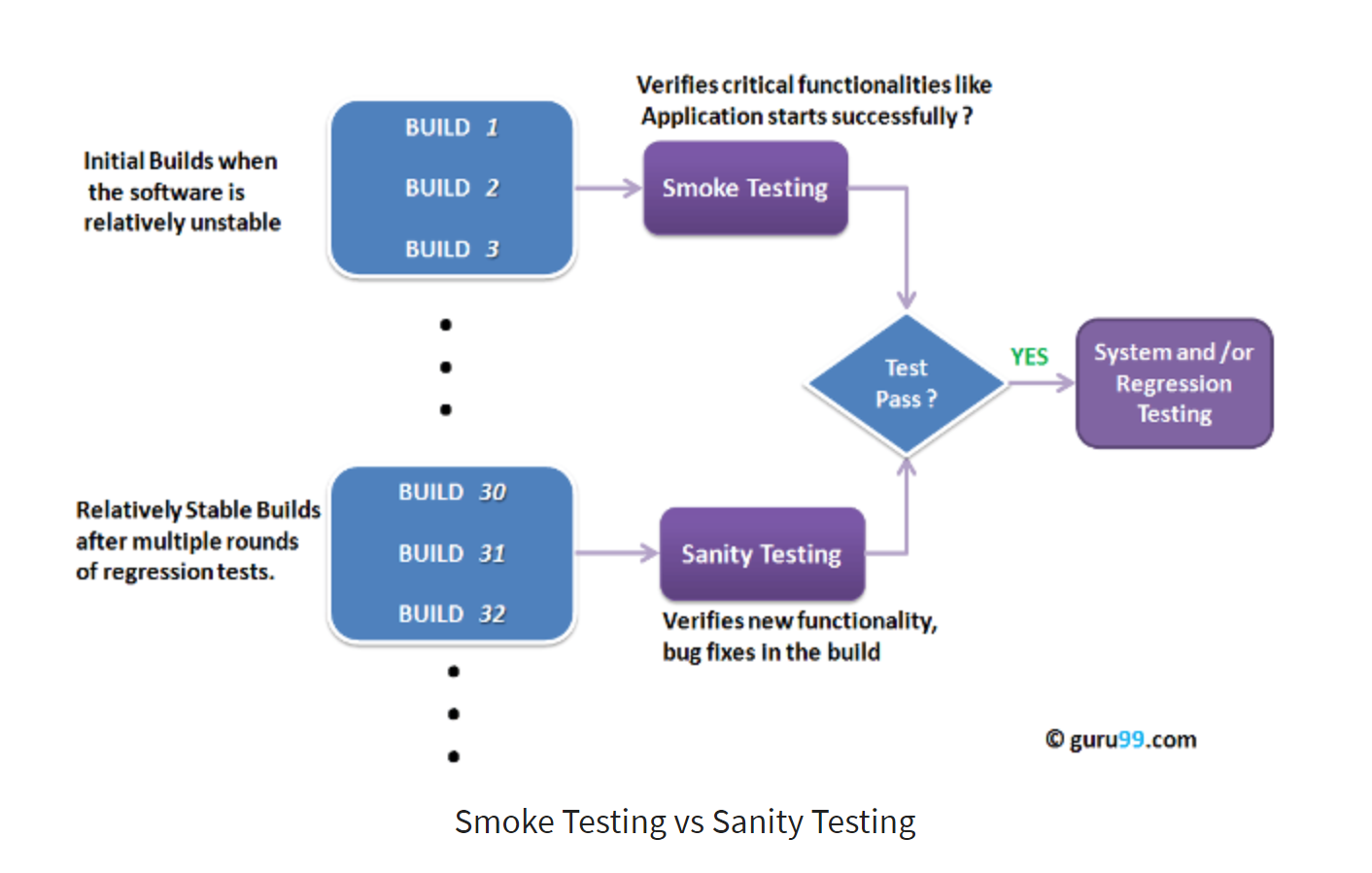
**V0.3->**

**V1.0->**

**Sanity Testing:** Testing the new functionality which is introduced in an application is called Sanity Testing. Sanity testing ensures that build is successful.

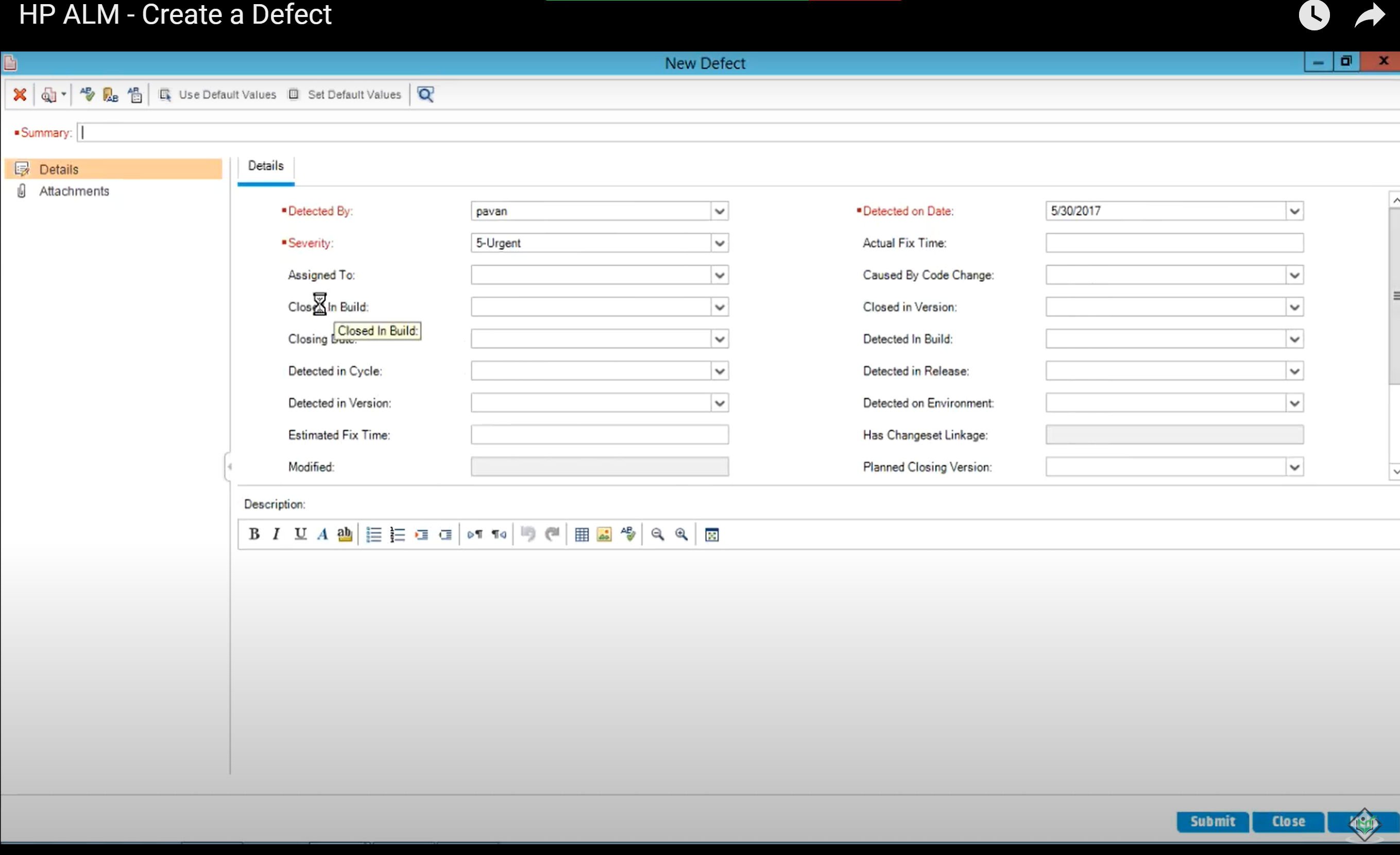
**Regression Testing**: Regression Testing ensures that existing modules or flow in an application is not getting effected by introducing the new functionality. Testing whether all existing modules / flow in an application is working as expected.

Regression Testing uses automation scripts and ensure all existing modules/flow in an application or a software is working as expected.



JIRA , HP ALM are the generally most used defect management tools .

**Steps need to be followed when Raising a bug/defect?**



**Fields Information:**

**Severity: Based upon the how critical functionality is getting effected (high, low, medium, critical)**

**Assigned To: Developer Name**

**Closed in Build: Flipkart -April Release /**

**Closing Build: SIT**

**Detected in Cycle: April 2021**

**Detected in Version: V11.3**

**EFT: 8 hours**

**Priority: Based upon how soon the defect need to be fixed.**

**Summary:** flight734556 – In SIT, when customer is trying to book flights, Search button is not enabled even all mandatory details are filled.

**Description:** Search button is disabled in flight booking application at “Get.Set.Travel” Page

------------start of description ----------------

**Steps to Reproduce:**

1.Login into flipkart application with valid user user name and password.

2.click on flights tab

3.Fill all the mandatory details to search for the flights.

**Below is the test data used:**

From: Hyderabad

To: Calicut

Depart On: 20 April, Tue

Traveller Class: 1 Traveller/Economy.

**\*Actual Result**: Search button is disabled and customer not is able to search for flights.

**\*Expected Result**: Search button should be enabled allowing customer to search for flights.

------------end of description--------------

**Attached a Screen Shot**: to the defect.

**Defect ID: 897235**

**Defect/Bug Life Cycle:**

**New ->Assigned ->In Progress->Fixed->Ready for Re-Test->Closed (If defect is resolved)**

**New ->Assigned ->In Progress->Fixed->Ready for Re-Test->Re-Open (if defect is not resolved)**

**New:** When created

**Assigned:** when assigned to developer

**In progress:** When Developer starts working on it

**Fixed:** When Developer provided solution (code changes had been made/included by developer)

**Ready for Retest:** When code changes are deployed to testing environment.

**Pending for clarification:** When developer need more information regarding the defect.

**Reject:** When developer rejected

**Issue to watch for:** Testing team is monitoring this defect**.**

**Deferred:** It is Defect which is accepted by development team and it will be fixed by dev team in next release /Iteration

**Closed:** When QA team re-test the defect and found that issue is resolved.

**Re-Open**: When QA team re-test the defect and found that issue is **NOT** resolved.

**Blocked**: When One defect depends on another defect

**Priority of Defect**: Based up on how **soon** the defect needs to be fixed by development team.

**Severity of Defect:** Based on how critical the **functionality** is getting effected.

**Severity High, Priority High:** Login Functionality of Application is not working or Home Page is not loading.

**Severity Low, Priority High:** company Logo is not properly displayed.

**Severity High, Priority Low:** Download statement Button is disabled in banking application.

**Severity Low, Priority Low:** spell mistakes

**Software Test Life Cycle**:

1.**Requirments Analysis**: Gather all the requirements and analyse and clear our doubts if any.

2.**Design Test cases based on test scenarios**: write test cases and get it reviewed by peer/lead

3.**Prepare test data and also test environment (SIT & UAT):** test data and test environment should be ready.

**4.Execute Test cases ->** raise the defects/bug and get it fixed by dev team

5.**Test closure**: All defects are fixed and thoroughly tested, documented in RTM / location, test data.

**Re-Testing:** Re-Testing the functionality which was already tested, observed the deviation from expected behaviour and reported to developer with bug details. After developer fixes that particular defect, we are going to re-test to make sure it is working correctly as expected.

Order by which software is released to market :

**Unit Testing -> SIT Testing -> UAT Testing ->Alpha Testing ->Beta Testing->Release the product to the Production.**

**Alpha Testing**: performed by **skilled testers / developers at the developer’s site** after acceptance testing. It ensures to identify every type of issue which was not identified before in previous rounds of testing.

**Beta Testing**: performed by group of users who are not a part of organization (Third party users). It is performed to validate whether the application is satisfying customer needs and requirements completely. Here all the business scenarios will be validated and decides whether or not to release the product.

----------------------------------------------------------------------------------------------

**Test Strategy**: High level document usually developed by project manager. It captures the approach on how we are going to test the product and satisfy business customer requirements. It contains the wire frames and business flow in an application.

**Test plan:** Testers will take Test Strategy and start writing test plan. Prepared by test lead/manager.

Focus on what to test and what not to test and what is our scope of testing. When to test, what to test and how to test. It contains in which tools we are using for automation and what is framework that is followed and where is the test data maintained. It should always keep up to date. Exit criteria is included and risks included are also mentioned in this test plan document.

**Compatibility Testing**: Testing the functionality in different operating systems like windows 10, windows 8, windows 7, Mac OS, IOS 14, IOS 12...etc, Android version 11, 10,9 and different browsers like chrome, IE, Safari, Edge ...etc.

**API Testing**: Testing the backend services. It will ensure the application is working fine and services provided by them will be utilized across different platform. SOAP API and REST API.

Tools used to test API: Postman, SOAP UI, rest assured services for automation Rest API’s.

We can identify defects in early stages.

Below are response codes:

1. [Informational responses](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#information_responses) (100–199)

2. [Successful responses](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#successful_responses) (200–299)

3. [Redirects](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#redirection_messages) (300–399)

4. [Client errors](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#client_error_responses) (400–499)

5. [Server errors](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#server_error_responses) (500–599)

Pease visit: https://developer.mozilla.org/en-US/docs/Web/HTTP/Status

Https requests:

* GET: The resource has been fetched and is transmitted in the message body.
* HEAD: The entity headers are in the message body.
* PUT or POST: The resource describing the result of the action is transmitted in the message body.
* TRACE: The message body contains the request message as received by the server.

**Negative Testing:** we will perform negative testing to know how application behaves when invalid values are given by end customer / end user.

**ATM:** withdraw 500 Rs from ATM

Pre-condition / pre- requisite:

1.valid card should be taken.

Test cases:

1.Card should be detected.

2.Testing the pin of ATM card with invalid pin at first time (Negative)

3. Testing the pin of ATM card with invalid pin at second time (Negative)

4. Testing the pin of ATM card with invalid pin at third time (Negative)

5.users should be communicated about the blockage of card. Message should be displayed to the user. Customer should get information about this transaction via message and mail.

The messages should contain which location, which bank and ATM location.

6.correct pin has been provided to user (positive)

Expected result: in should identify my user details and proceed further to access the feature of ATM.

7. download last 10 transactions.

Expected result: user should get 10 transactions Details via a slip.

8.Test pin change functionality

Authorization of user: posting the security questions.

All responses need to be validated by machine.

9.withdraw 500rs from ATM

Transaction should process and machine should discharge 500 rs to user.

10. notification messages/mail should be triggered automatically.

->funding information should update to bank and it should reflect in account.

**Verification and Validation**

**Verification:** Are we building the product right?  
**Validation:** Are we building the right product?

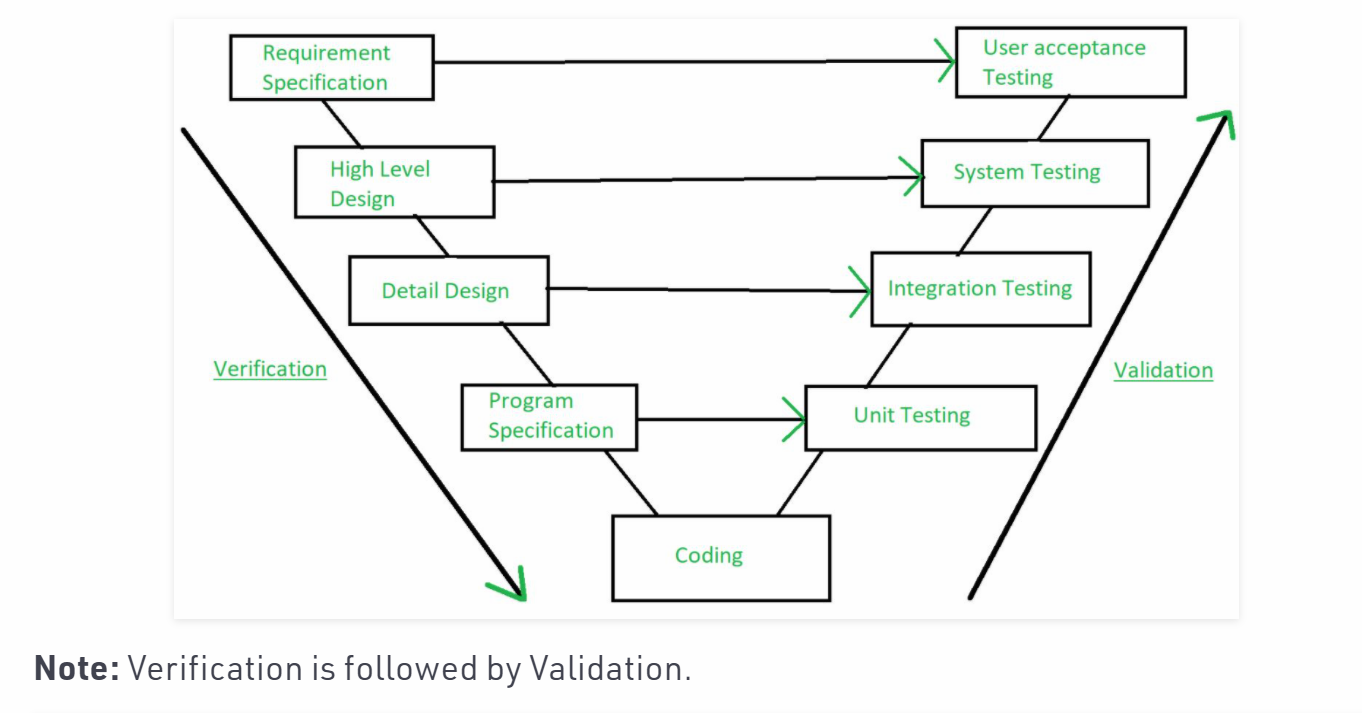
Activities involved in verification:

1. Inspections
2. Reviews
3. Walkthroughs
4. Desk-checking

**Validation:**  
Validation is the process of checking whether the software product is up to the mark or in other words product has high level requirements. It is the process of checking the validation of product i.e. it checks what we are developing is the right product. it is validation of actual and expected product.  
Validation is the **Dynamic Testing**.

Activities involved in validation:

1. Black box testing
2. White box testing
3. Unit testing
4. Integration testing



**Agile Methodology:**

Stake holders/Product owners /Business people:

Requirements to develop any product or software will be analysed at high level and start preparing Product Backlog. **Product Owner** is responsible for this.

Let’s say we are in **sprint** **52:**

Sprint duration will be from 1 week to 4 weeks.

**3week sprint:** The User stories given need to be developed and tested within 3 weeks duration.

**User Story 345**: **Big Savings Days** need to be included in flip kart home page as a banner.

Description: As a customer I need feasibility to view the **Big Savings Days** promotion on flip kart application home page as a banner

**Acceptance Criteria:**

The banner should contain the below information:

1.Sale should start on 2nd May at 12 AM.

2. clock icon should appear.

3.big saving days need to be displayed on the bag

4.duration need to be mentioned, service provider name needs to be mentioned.

5.Bank information which includes percentage of discounts/offer.

6.Stay Home, shop home should be displayed

7.Sneak peak button should be displayed and should be enabled and when user select it, should navigate to offer details page

User Story points: 0,1, 1,2,3,5,8,11,13,17 -> Follows Fibonacci series

8\*8=64 hours of time is allocated to develop and test this banner **user story 345.**

**Development time: 48 hours**

**Testing team: 16 hours**

**sprint** **52:** 8 user **stories** are taken by dev team and testing.

**3-week sprint: 21 days -> 15 are working days**

**Per day 8 hours -> so for 15 days: 15 \*8 =120 hours**

**6 developers: 6 \*120 = 720 hours ->400 hours taken into consideration**

**3 testers: 3\*120 =360 hours ->200 hours taken into consideration**

**Total 1080 hours.**

**600 hours: Take this into consideration**

**Hours utilized: 64**

Sprint Planning: usually happens at beginning day of sprint.

Sprint Planning: Accepting 9 user stories.

**Sprint grooming**: takes place at beginning day of the sprint.

BA, Dev Team and Testing Team will be involved -> Analysed all the user stories accepted in Sprint 52.

Requirement Analysis need to be completed by 2nd day-> both dev team and testing team.

3rd day of sprint: Dev team need to start developing the user story and testing team need to start writing test cases.

**Scrum Calls / Daily Stand-up meetings**:(15 minutes) Scum Master will be leading this.

Each and every member (both dev team and testing team):

1.what went well/ what we have done yesterday

2.what we are going to do today.

3.Any Impediment’s (blockages) for the work.

**Scrum Master:**

Us 1344: dev and team

Dev:

Test:

Impediment’s:

Us 1346: good

Us 1345: good

**Testers / QA Team:**

3rd Day, 4Th Day: Writing the test cases and get it reviewed by Peer Review or QA Lead.

Test cases can be in Excel Sheet or uploaded to **JIRA or HP ALM.**

Also Update the RTM (Requirement Traceability Matrix)

5th Day -> Test data need to prepared and make the environment ready.

2nd Week or 3rd Week: -> If any user Story code is deployed into Testing Environment (SIT, UAT), Start Testing the User Story and raise defects/bugs. Get them fixed by dev team and re-test the defects and close them.

**Exit Criteria**: When all the test cases (+ve and negative) are executed and all the defects are resolved/ retested by QA Team. When integration of modules/flow is tested. When all the AC mentioned in User Story has been satisfied, then we can say user story can be delivered.

**All the test data used for testing and test cases, test scripts including screen shots are stored in common location.**

**Demo to PO: date 27 April 5 :30 pm:**

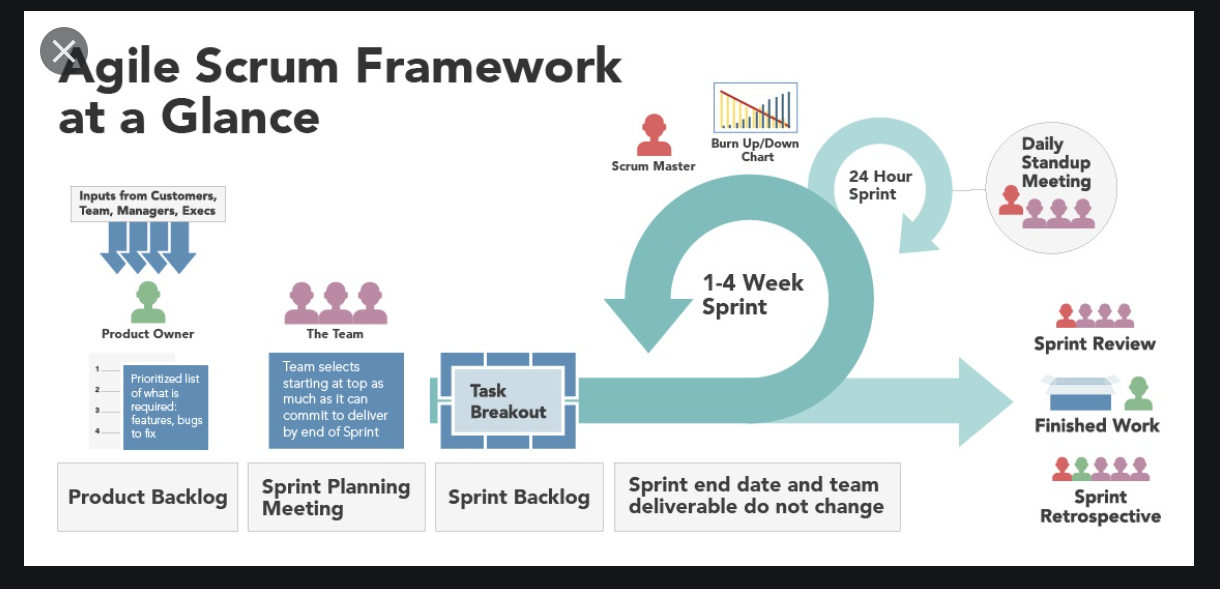
**User Story information need to be open in JIRA.**

**Sprint Retrospective Meeting:** At the end of the sprint, we will have this meeting.

**Discuss about:**

* 1. What went Well
  2. What can be improved (out of 9, we have delivered 7)
  3. Any feedback or suggestions are welcome/acceptable.

**2 user stories will be added to product backlog**



**Burnt Down Chat:**

**JIRA:**

**Testing Activities:**

1. **Analysing User Story and writing test cases -> it is strike out which indicated that this particular activity is completed.**
2. **Preparing the test data.**
3. **Executing test cases**

**-----------------------------------------------------------------------------------------------------------------------------------**