

* Black Box Testing :-

⇒ It is a method of SW testing that examines the functionality of an application without looking into its internal structures of working.

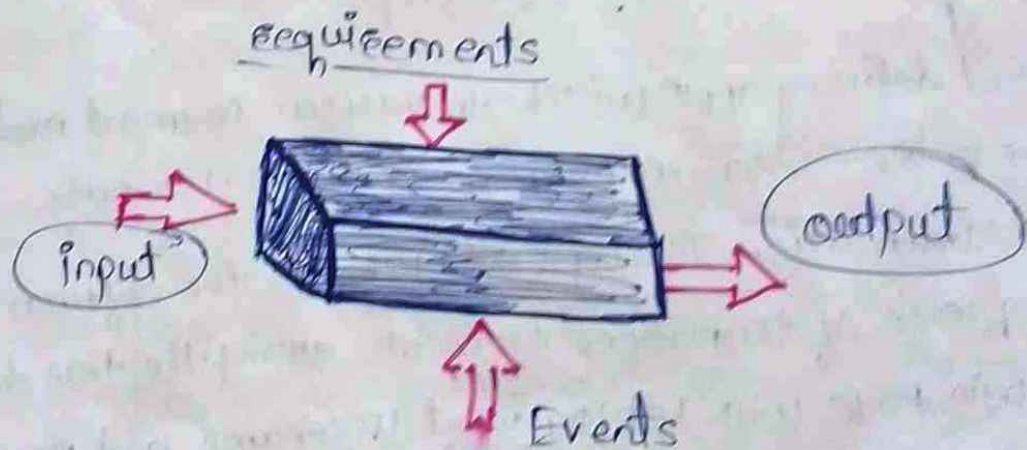
⇒ This method is applied at every level of SW testing:

Unit, Integration, System and acceptance.

⇒ Tests are based on requirement & functionality.

unit testing
done on the module
ie. smallest module so
inside the smallest module if you
want to test the code that will be
done by unit testing.
so the smallest module of an
application
will be tested by
the black box.

Whenever you combining all the
smallest element into one
element that they will do in
Integration testing. so the integr
testing done internally how the
code is interacting is able to connect
the modules.



⇒ It is carried out to test functionality of the Program
also called as "Behavioral testing".

⇒ The tester in this case
has a set of i/p values and desired results. On provided
i/p, if the o/p matches with the desired results, the
program is tested 'OK'.

⇒ In this testing method, the design & structure of code are known to tester and testing engineers and end users conduct this test on s/w.

* Black Box testing techniques

- ① Boundary value Analysis :- BVA is used to identify the flaws or errors that arise due to the limits of input data.
eg :- 1 - 100
BVA - -1, 1, 100
- ② Equivalence Partitioning :- It checks the i/p into equivalent classes. The data must be tested at least once to ensure maximum test coverage of data.
- ③ Decision table testing :-
- ④ State Transition Testing :-
- ⑤ Graph - Based Testing :-
- ⑥ Fuzz Testing Technique :-

This approach creates test cases based on various possibilities. It considers multiple test cases in a decision table format where each condition is checked & fulfilled, to pass the test & provide accurate o/p.

eg :- valid class :- 1 to 100 (any No)
Invalid class :- -1 (checking the lowest of lowest)
Invalid class :- 101 (Highest to H)

eg :- A food delivery app will check various payment modes as i/p to place the order - decision making based on the table.

- ④ State Transition Testing :- It checks the software against the sequence of transitions or events among the test data.

eg :- A login page will let you input username and password until three attempts. Each incorrect password will be sent the user to the login page. After the third attempt, the user will be sent to an error page. This state transition method considers the various states of the system & the i/p to pass only the right sequence of the testing.

HTTP-based testing:-

It is similar to a decision-based test case design approach where the relationship between links & i/p cases are considered

5) Error Guessing Technique:-

This method of designing test cases is about guessing the output & i/p to fix any errors that might be present in the system. It depends on the skill & judgment of the tester.

* API status code / HTTP [common] status codes.

- ^{200's}
① 100 (informational) ② 200 OK (success) ③ 300 (Redirection)
④ 400 (client error) ⑤ 500 server error.

200's Success
200: OK
201: Created
204: No content

400's Client errors
400: Bad request
401: Unauthorized
403: Forbidden
404: Not found
405: Method not allowed
411: Length required
414: URI too long
415: unsupported media type

500's Server errors
500: Internal server error
501: Not implemented
502: Bad gateway
503: Service unavailable
504: Gateway timeout

* CRUD :- Create Read update Delete.

Rest

SOAP

- * Representational state Transfer
- * It permits many data formats, including plain text, HTML, XML & JSON

* Require fewer resources and is lightweight

* data can be cached

* ~~Protocol style~~
Architectural style

* simple object access protocol

* only uses XML data format.

* Required more resources and bandwidth.

* data cannot be cached

* Protocol style

* Random testing (Monkey testing) / ad-hoc testing

→ we will perform the abnormal actions on the application to check the reliability of the application.

eg. - we click the search button more times.

* Error guessing defect :- Based on expertise level we able to judge & find the defects proactively is called error guessing defect.

* Error seeding defect :- introducing the defect to find out more defects or finding the root cause analysis of some defect.

CRUD functionality (Create, Read, Update, Delete)

To perform CRUD function we have HTTP methods

- | | | | | |
|---------------|----------------|---------------------------|---------|---------------------------|
| ① Get | ② Post | ③ Put | ④ Patch | ⑤ Delete |
| ↓ | ↓ | ↓ | ↓ | ↓ |
| Read the data | To create data | It use to update the data | | It use to delete the data |

POST

① Post sends data to a particular URI & expects the resource at that URI to deal with the req. The web server at this point can decide what to do with the data in the context of specific resource.

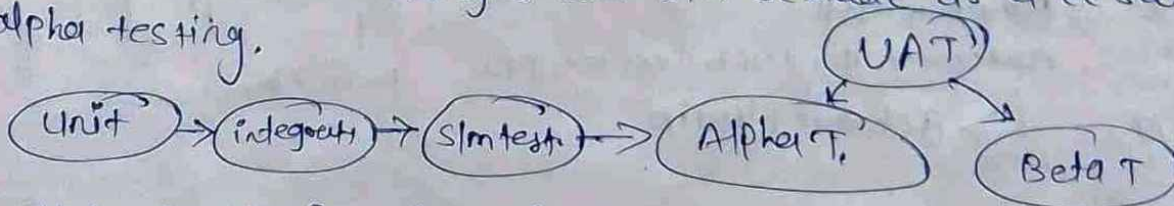
PUT

① PUT puts a file or resource at a particular URI and exactly at that URI. If there is already a file or resource at that URI, PUT changes that file or resource. If there is no resource or file there, PUT makes one.

UAT → Alpha testing & Beta testing.

↓ (internal user acceptance testing)

alpha testing is testing of an applⁿ when development is about to complete. minor changes can still be made as a result of alpha testing.



Alpha testing is a type of acceptance testing performed to identify all possible issues/bugs before releasing the product to real users.

* Beta Testing :- Beta testing is performed by real users of the software applⁿ in a real environment & can be considered as a form of external UAT.

eg:-

* Sanity Testing :- Sanity testing is a subset of regression testing, performed when we do not have enough time. It happens when minor changes in code.

* Smoke Testing :- To find whether the build is stable or not by testing whether the most critical components are working or not.
(Build verification Testing)

(Build) → Software which consists of some set of features, bug fixes & is installed on the test server, which needs to be tested for stability.

(Sanity) it is a part of Regression

↓ the bug.
Bug raise then inform to dev.

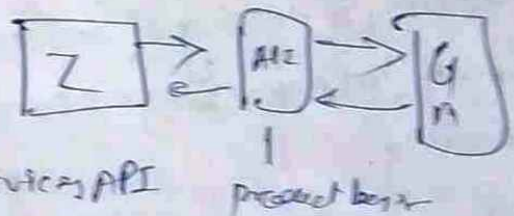
↓
Retesting for automation tool

* Postman tool / SOAP UI

use to → plugin is used to automate the web services API

Rest assured API → Chrome plugin

API - App's prog. interface



Web services → SOAP & REST

Diff. of severity & Priority?

→ If the company name is misspelled in the home page of the website, then the priority is high & severity is low to fix it.

Bug Severity

High Severity & low Priority
Status means that the bug can cause significant damage, but can be fixed at a later date.

Bug Priority

High Priority & low Severity
Status means that the bug must be fixed immediately but it does not affect the software too adversely.

* 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.

(Eg. A site maintaining the student details, on saving record if it, doesn't allow to save the record then this is high priority & high severity bug.)

* Low Severity & Low Priority.

(Eg. A minor low severity bug occurs when there is almost no impact on the functionality but it is still a valid defect.)

* Regression Testing.

The testing is done to make sure that new code changes should not have side effects on the existing functionalities.

It ensures that the old code still works once the latest code changes are done.

Comments,
Concepts,
are placed
check

What do you do when you find a bug?

- ① Analyse the issue.
- ② Identify the steps
- ③ What was scenario
- ④ Check the logs
- ⑤ Try to do Root Cause Analysis.

* What will you do when developer does not accept bug?

-
- ① Don't fight with the developer
 - ② Check with PO about the use case
 - ③ Check with BA for the requirements
 - ④ Discuss the scenario
 - ⑤ Collaborate with Team.