Levels of testing

1. Unit testing

Testing a single module. Always performed by developers.

1. Integration testing

Two or more modules are integrated together and are tested for their integration.

Test the interface between the modules.

Approaches of IT:

Module A: Main Module

Module B: Sub Module

Top Down Approach | Bottom Up Approach

Ready Calling program Not Ready

A

A

|

Dummy code – Test Driver

|

Not Ready |

B

B

Called program Ready

Dummy code |

Test Stub |

1. System Testing

Tester tests the functional as well as nonfunctional attributes like Functional attributes, Performance, Speed, Compatibility, UI, Security, etc.

1. User Acceptance testing (UAT)

Users test the application before release.

Done in two ways:

1. Alpha testing

Done in production environment.

Customer is invited in the organization and asked to test the application.

Developers have total control in alpha testing.

1. Beta testing

Done in live environment.

Customer is given the application for some time period for testing it in his/her own environment.

Developers don’t have control in beta testing.

Once beta testing is done and customer is satisfied, then the application is released to the customer.

System Testing Strategies:

1. Functionality testing

Input field validation. E.g. if the field is to enter our name then only alphabetical data should be allowed.

Links- take to next page.

Data is entered then data should be stored into the database.

1. Smoke testing

Testing the basic functions of the application.

Helps to decide to continue actual functional testing of the module.

Test the stability of the application.

If any defect is found then the build is rejected.

1. Sanity testing

Testing any function in depth is Sanity testing.

Example:

Module A is under test – 50 defects – report – fix.

Tester used 100 test cases to test module A.

Now the developer has fixed the defects.

Fix – Retest using same 100 test cases – 0 defects.

1. Retesting

Re-executing or re-running any failed module is retesting.

Module A – test – defect – Fix – Retest – Close.

Retesting is performed to confirm if the defect is really removed from the module.

1. Regression testing

Any changes made in our application should not impact the unchanged part.

Module A, B, C, D, E, F, G, H, I, J ,K are interconnected.

Changes like:

1. Change in code (defect fixing)
2. Change in requirement (change request)
3. Environmental change.

Module A – defects – fix – retest – close.

To check which modules are interconnected with Module A, we use a process called as “Impact Analysis” – analyses the impact of the changes.

IA – modules B, C, E, G. – regression testing.

How much regression shall be performed?

Dependent on Impact analysis.

1. Compatibility testing

Is also called portability testing.

Checking is application runs in the specific OS or browser.

1. User Interface testing/ Usability testing

Testing the look N feel or appearance of the application.

UI attributes like color combinations, effects, graphics, font size, element size, spelling mistakes, grammar mistakes, navigation, images, etc.

1. Mutation testing

Changes in logic of code are made to test the code’s efficiency.

1. Defect Seeding

Defects are seeded in the application to test the tester’s efficiency.

Dev – build1 – 10 seeded defects – tester – 30 defects – report it to developer.

Developer tries to check if tester has found any of the seeded defects i.e. out 30 reported defects, how many are the seeded one. Let’s say out of 30 there are 7 seeded defects found by tester.

\*Adhoc Testing

Testing is performed if we lack in the resources we need for testing.

Ways of Adhoc testing:

Lack in time:

Monkey Testing: Any random module is tested with random inputs. If the module seems while testing it then tester put some extra efforts to test it which will save the time invested on less important module’s testing.

Buddy Testing

Lack in documentation:

Exploratory Testing: Explore the application. Try to understand the application by market survey, taking help of your colleagues, going through the previous projects, own experience or internet.

Lack in Skillset:

Pair Testing: Testers and Developers are paired together and testing is performed.

Sr. Tester and Jr./ Sr. Developer – Pair in pair testing.

\*. Performance Testing

Response time

T1

Client Server

T2

T1 – time to send the request

T2 – time by server to give response.

What is Response time? Answer: T1+T2

Throughput: no of requests per unit time.

Benchmark: target set to achieve.

Saturation point: Degradation point.

Tuning: adjustments made to achieve the required performance.

Latency: Delay