Negative Testing

Negative Testing is a software testing type used to check the software application for unexpected input data and conditions. Unexpected data or conditions can be anything from wrong data type to strong hacking attack. The purpose of negative testing is to prevent the software application from crashing due to negative inputs and improve the quality and stability.
By just doing positive testing we can only make sure our system is working in normal conditions. We have to make sure that our system can handle unexpected conditions to ensure a 100% fault-free system.

INTEGRATION TESTING is defined as a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated

REGRESSION TESTING is defined as a type of software testing to confirm that a recent program or code change has not adversely affected existing features.

Regression Testing is nothing but a full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine.

Need of Regression Testing

The **Need of Regression Testing** mainly arises whenever there is requirement to change the code and we need to test whether the modified code affects the other part of software application or not. Moreover, regression testing is needed, when a new feature is added to the software application and for defect fixing as well as performance issue fixing.

How to do Regression Testing

In order **to do Regression Testing** process, we need to first debug the code to identify the bugs. Once the bugs are identified, required changes are made to fix it, then the regression testing is done by selecting relevant test cases from the test suite that covers both modified and affected parts of the code.

Software maintenance is an activity which includes enhancements, error corrections, optimization and deletion of existing features. These modifications may cause the system to work incorrectly. Therefore, Regression Testing becomes necessary. Regression Testing can be carried out using the following techniques:

Selecting test cases for regression testing

It was found from industry data that a good number of the defects reported by customers were due to last minute bug fixes creating side effects and hence selecting the Test Case for regression testing is an art and not that easy. Effective Regression Tests can be done by selecting the following test cases –

- Test cases which have frequent defects
- Functionalities which are more visible to the users
- Test cases which verify core features of the product
- Test cases of Functionalities which has undergone more and recent changes
- All Integration Test Cases
- All Complex Test Cases
- Boundary value test cases
- A sample of Successful test cases
- A sample of Failure test cases

• Try to enter an invalid email id in the email field	• Only valid email ids will be e email field
• Try to enter an invalid phone number in a phone number field (characters)	• The only number will be ent number field
• Upload image with size out of specified boundary	• Only images with size under boundary will be uploaded
• Upload invalid files like XML, SQL, etc. files in image upload field	• Upload only valid image for etc.

Smoke Testing is a software testing process that determines whether the deployed software build is stable or not. Smoke testing is a confirmation for QA team to proceed with further software testing. It consists of a minimal set of tests run on each build to test software functionalities. Smoke testing is also known as "Build Verification Testing" or "Confidence Testing."

When do we do smoke testing

Smoke Testing is done whenever the new functionalities of software are developed and integrated with existing build that is deployed in QA/staging environment. It ensures that all critical functionalities are working correctly or not.

In this testing method, the development team deploys the build in QA. The subsets of test cases are taken, and then testers run test cases on the build. The QA team test the application against the critical functionalities. These series of test cases are designed to expose errors that are in build. If these tests are passed, QA team continues with [Functional Testing](#).