Smoke & Sanity Testing

Smoke Testing also called as Positive Testing, Build verification Testing, Sanity Testing

Now, let’s assume, we have a 1 Sprint which has a duration of 10 days and Release duration is 20 days and we need to test 3 modules A, B and C. Now, let's say we completed testing for module A and B, now on 7th day we just started testing for module C, now there is a blocker defect and because of this defect we are unable to proceed further. Now, TE will inform the developer to fix it ASAP else the release will get postponed. Once developers fix this issue, we need to start testing again from functional testing. If we go with this flow, it will take 27 days to release the build to the customer instead of 20 days. Customer will not accept this issue as it will affect his business. Hence to avoid these kinds of issues we should do Smoke testing in the beginning stage itself.

So, remember Smoke Testing is the first testing done by Test Engineers. Followed by Functional Testing, Integration Testing, System Testing, Regression Testing etc.

What is Smoke Testing?

It is High level testing conducted before doing rigorous testing or other types of testing to ensure that there is no blocker or critical defects that affects the customer business workflow.

Q) Why Smoke Testing is necessary?

- Because it helps in early detection of Blocker and critical bugs
- It verifies that the latest build or release is stable enough for further testing. If the build passes the smoke test, it indicates that the basic components are intact and functioning as expected.
- It ensures basic functionalities of the software is working fine
- It saves time and resources as we detect the blocker and critical bugs in the early stage itself.

Q) When to do Smoke Testing?

- Whenever new build is released
- Before conducting rigorous testing i,e,. Functional, Integration, Regression, End to end testing etc.

What is Sanity Testing?

It is also a kind of Smoke testing but it is conducted usually when we find a bug, then once retesting is done, we do sanity testing, meaning we check the application in depth and verify all dataflows if it is working correctly or not. This is called Sanity Testing.

It is basically a focused and narrow verification process performed on a specific functionality or a small set of functionalities within a software application. The primary purpose of sanity testing is to quickly check whether a particular aspect of the software is working correctly after changes, updates, or bug fixes.

Q) Why is Sanity Testing necessary?

- It helps to quickly validate specific/key functionalities of a software after some changes are done
- It helps to save time and resource as we focus on critical functionalities
- It acts like a build verification where it ensures if the current build is stable enough for further testing or not.

Regression Testing

Testing the unchanged/old feature of an application to make sure that changes like adding a feature, modifying a feature, deleting a feature or fixing a defect is not introducing any defects in the changes or old feature is called Regression Testing.

Regression Testing is also called as **Release Candidate Testing**

Note:

1. First always test for newly added feature or any other changes made

2. Then test old module(nothing but regression testing)

Types of Regression Testing

1. Unit Regression Testing

2. Regional Regression Testing

3. Full Regression Testing

Unit Regression Testing – Testing only the changes or modifications done by the developer is called Unit Regression Testing.

Example : First name, Last name, In sign up country dropdown, FAQ section

Regional Regression Testing – Testing the changes and impacted areas of an application is called Regional Regression Testing.

Example : 1) Gmail - Assume an attachment file is being added in the compose module of gmail application, this will impact other modules such as Inbox, sent items etc.

2) Facebook – Assume features of FB – sign up, login, photo, like, share, logout. Now, if you add an additional feature upload gift in the photo module, this will have an impact on others such as share. So, we need to do Regional Regression Testing in this case.

Q) How do we know which all areas got impacted?

A) By doing an Impact Analysis meeting – Here we interact with Sr TE, Developers, BA, customers who have very good product knowledge, gather the information about the impacted areas and consolidate and document the impacted areas, this process is called an “Impact Analysis meeting”.

Full Regression Testing – Testing the changes and all the remaining features of an application is called Full Regression Testing.

Example: Let’s say there is an issue in one of the cell of MS Excel, since the cell is the root of the product here, we will have to make changes to the entire application page.

Q) When we go for Full Regression Testing?

- When changes are more, do not spend time doing impact analysis meeting, test the software by doing Full Regression Testing.
- When changes are made to the Root of the Product, then we need to test the entire product by doing Full Regression Testing.

Q) Drawbacks of Regression Testing?

- Time taken is more
- Manpower is more
- No consistency in Testing

To overcome the drawbacks of Regression Testing, companies go for Automation.

Regression Testing acts like a bridge between manual and Automation testing, when there are more number of Regression test cases, then we convert those manual test cases into automation script by using tools like Selenium/QTP.

Q) What is Retesting?

A) When TE finds a bug, he asks the Developer to fix the bug. The developer fixes the bug and gives it back to us, again as TE we need to retest if the bug is fixed or not. This is called Retesting. It is mandatory testing and high priority when compared to Regression Testing.

Importance of Regression Testing:

- Regression testing helps to detect bugs introduced by new code changes.
- Regression Testing ensures that existing features or functionalities work as expected.
- It ensures that bug fixes or modifications made does not impact other functionalities
- By conducting Regression Testing we can ensure that overall integrity of the software is achieved.
- In an Agile environment, where changes are frequent, regression testing helps team to adapt to continuous modifications.