

Sanity , Smoke and Regression Testing

Sanity Testing

Sanity Testing as a test execution which is done to check existing/previous functionality and its impact but not thoroughly or in-depth.

The sanity test should only be done when you are running short of time, so never use this for your regular releases. Theoretically, this testing is a subset of regression testing.

Sanity testing is done at random to verify that each functionality is working as expected.

This is to verify whether the requirements are met or not, by checking all the features breadth-first.

This is not a planned testing and is done only when there's a time crunch.

This mainly includes verification of business rules, functionality.

This mostly spans over 1-2 days max.

Regression Testing:

Regression testing is done to verify that the complete system and bug fixes are working fine

This includes in-depth verification of functionality,

This mostly spans over 2-3 days .

Test cases are generally automated as test cases are required to be executed again and again and running the same test cases again and again manually is a time-consuming and tedious one too.

Test cases are re-executed to check the previous/existing functionality of the application is working fine, and the new changes have not produced any bugs.

When the defect fixed-Defects fixed as per the criticality/priority of the defect/bug

Example:

Assume login button is not working in a login page and a tester reports a bug stating that the login button is broken. Once the bug fixed by developers, tester tests it to make sure Login Button is working as per the expected result. Simultaneously, tester tests other functionality which is related to the login button.

Smoke Testing:

This testing is a normal health check-up to the build of an application before taking it to test in-depth.

This testing is conducted to ensure whether the most crucial functions of a program are working,

Smoke Testing is directly related to Build Acceptance Testing (BAT).

In BAT, we do the same testing – to verify if the build has not failed and if the system is working fine or not.

Sometimes, it happens that when a build is created, some issues get introduced and when it is delivered, the build doesn't work for the QA.

Smoke Testing is ideally performed by the QA

It is used to test the acute functionality of the software. When the developers deliver a new build to the Quality teams, smoke testing is done

Practical Rule for Sanity and Regression-

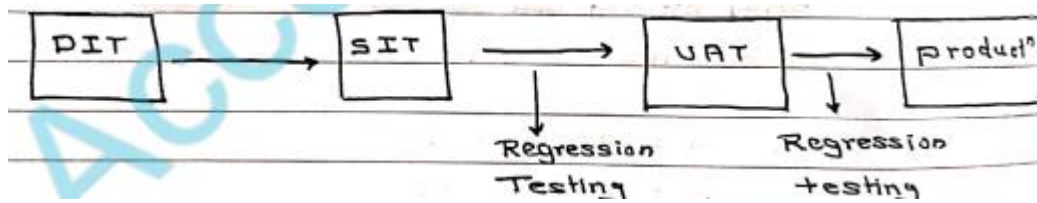
1.Requirement move from 1 server to other server then we have to perform it bcz we have to check whther any impact not happened to exitig functionality as we hvae changed code.

2.Need basis - If developement team or we think as like in current sprint there is any user story and that user story is impacting much for exiting functionality then we can perform - S/R

3.Regression - If any big requirement/module/submodule moving from 1 server to other server then its good to perform regression bcz it may impacting many modules or exiting features (Example - flight complete module implementation and need to deploy on production)

4.Sanity - If limited time to execute existing feature OR if any small feature change (Example - design change or email etc [not much impact in existing feature]) is moving to 1 server to other server OR need basis (refer point 2)

-We have 4 environments/servers so we do regression once the code move from 1 server to other to make sure everything works well.



SMOKE TESTING	SANITY TESTING
Smoke Test is done to make sure if the build we received from the development team is testable or not	Sanity Test is done during the release phase to check for the main functionalities of the application without going deeper
Smoke Testing is performed by both Developers and Testers	Sanity Testing is performed by Testers alone
Smoke Testing, build may be either stable or unstable	Sanity Testing, build is relatively stable
It is done on initial builds.	It is done on stable builds.
It is a part of basic testing.	It is a part of regression testing.
Usually it is done every time there is a new build release.	It is planned when there is no enough time to do in-depth testing.