System Testing Supplemental

Prepared by: Zafar I Khan

- **SYSTEM TESTING** is a level of **software testing** where a complete and integrated **software** is tested. The purpose of this **test** is to evaluate the **system's** compliance with the specified requirements.

- System Testing involves testing the software code for following
  - Testing the fully integrated applications including external peripherals in order to check how components interact with one another and with the system as a whole. This is also called End to End testing scenario.
  - Verify thorough testing of every input in the application to check for desired outputs.
  - Testing of the user's experience with the application.

- Tasks
  - System Test Plan
    - Prepare
    - Review
    - Rework
    - Baseline
  - System Test Cases
    - Prepare
    - Review
    - Rework
    - Baseline
  - System Test
    - Perform
- **When is it performed?**
- System Testing is the third level of software testing performed after Integration Testing and before Acceptance Testing.
- **Who performs it?**
- Normally, independent Testers perform System Testing.

Different Types of System Testing

- There are more than 50 types of System Testing. For an exhaustive list of software testing types click here. Below we have listed types of system testing a large software development company would typically use
- **Usability Testing -** Usability Testing mainly focuses on the user's ease to use the application, flexibility in handling controls and ability of the system to meet its objectives
- **Load Testing -** Load Testing is necessary to know that a software solution will perform under real-life loads.
- **Regression Testing-** - Regression Testing involves testing done to make sure none of the changes made over the course of the development process have caused new bugs. It also makes sure no old bugs appear from the addition of new software modules over time.
- **Recovery Testing -** Recovery testing is done to demonstrate a software solution is reliable, trustworthy and can successfully recoup from possible crashes.
- **Migration Testing -** Migration testing is done to ensure that the software can be moved from older system infrastructures to current system infrastructures without any issues.
- **Functional Testing -** Also known as functional completeness testing, Functional Testing involves trying to think of any possible missing functions. Testers might make a list of additional functionalities that a product could have to improve it during functional testing.
- **Hardware/Software Testing -** IBM refers to Hardware/Software testing as "HW/SW Testing". This is when the tester focuses his/her attention on the interactions between the hardware and software during system testing.

What Types of System Testing Should Testers Use?

- There are over 50 different types of system testing. The specific types used by a tester depend on several variables. Those variables include:
- **Who the tester works for** - This is a major factor in determining the types of system testing a tester will use. Methods used by large companies are different than that used by medium and small companies.
- **Time available for testing** - Ultimately, all 50 testing types could be used. Time is often what limits us to using only the types that are most relevant for the software project.
- **Resources available to the tester** - Of course some testers will not have the necessary resources to conduct a testing type. For example, if you are a tester working for a large software development firm, you are likely to have expensive automated testing software not available to others.
- **Software Tester's Education** - There is a certain learning curve for each type of software testing available. To use some of the software involved, a tester has to learn how to use it.
- **Testing Budget** - Money becomes a factor not just for smaller companies and individual software developers but large companies as well.

SYSTEM TESTING vs. ACCEPTANCE TESTING

System Testing	*Acceptance Testing*
1. System testing is a type of **end to end** and black box testing type.	1. Acceptance testing is a functionality testing type.
2. It is performed to ensure that the system meets the specified requirements of the client.	2. It is performed to ensure the compliance of the product with business requirements.
3. It is performed after the completion of integration testing.	3. It is performed after the culmination of system testing.
4. System testing is performed by independent testers as well as developers.	4. Acceptance testing is performed by testers as well as customers.
5. It can be both functional & non-functional testing type.	5. This type of testing is mainly functional testing type.

SMOKE AND SANITY TESTING & ITS RELATION WITH SYSTEM TESTING…

- **What is Smoke Testing?**
- Smoke Testing is a kind of Software Testing performed after software build to ascertain that the critical functionalities of the program are working fine. It is executed "before" any detailed functional or regression tests are executed on the software build. The purpose is to reject a badly broken application so that the QA team does not waste time installing and testing the software application.
- In Smoke Testing, the test cases chose to cover the most important functionality or component of the system. The objective is not to perform exhaustive testing, but to verify that the critical functionalities of the system are working fine.
  For Example, a typical smoke test would be - Verify that the application launches successfully, Check that the GUI is responsive ... etc.
- **What is Sanity Testing?**
- Sanity testing is a kind of Software Testing performed after receiving a software build, with minor changes in code, or functionality, to ascertain that the bugs have been fixed and no further issues are introduced due to these changes. The goal is to determine that the proposed functionality works roughly as expected. If sanity test fails, the build is rejected to save the time and costs involved in a more rigorous testing.

Smoke Testing Vs Sanity Testing - Key Differences

Smoke Testing	Sanity Testing
Smoke Testing is performed to ascertain that the critical functionalities of the program is working fine	Sanity Testing is done to check the new functionality/bugs have been fixed
The objective of this testing is to verify the "stability" of the system in order to proceed with more rigorous testing	The objective of the testing is to verify the "rationality" of the system in order to proceed with more rigorous testing
This testing is performed by the developers or testers	Sanity testing is usually performed by testers
Smoke testing is usually documented or scripted	Sanity testing is usually not documented and is unscripted
Smoke testing is a subset of Acceptance testing	Sanity testing is a subset of Regression Testing
Smoke testing exercises the entire system from end to end	Sanity testing exercises only the particular component of the entire system
Smoke testing is like General Health Check Up	Sanity Testing is like specialized health check up

- Both sanity tests and smoke tests are ways to avoid wasting time and effort by quickly determining whether an application is too flawed to merit any rigorous testing.
- Sanity Testing is also called tester acceptance testing.
- Smoke testing performed on a particular build is also known as a build verification test.
- One of the best industry practice is to conduct a Daily build and smoke test in software projects.
- Both smoke and sanity tests can be executed manually or using an automation tool. When automated tools are used, the tests are often initiated by the same process that generates the build itself.
- As per the needs of testing, you may have to execute both Sanity and Smoke Tests in the software build. In such cases, you will first execute Smoke tests and then go ahead with Sanity Testing. In industry, test cases for Sanity Testing are commonly combined with that for smoke tests, to speed up test execution. Hence, it's a common that the terms are often confused and used interchangeably