

# Software Testing: Smoke and Sanity

Varuna Gupta

Vivek Sen Saxena

Integrated Academy of Management and Technology,  
Ghaziabad

## Abstract

Software testing is one of the most critical activities which is carried out throughout the software development for getting the best quality of a software product. Many innovative methods have evolved for performing software testing, among them the important ones are, Smoke Testing and Sanity Testing which use minimum time to test the software and give effective results. Sanity testing which is based on regression testing is used to analyze any change in the source code and to make sure that the changes do not establish new bugs in the earlier validated codes. The paper proposes to bring out the basic difference between Smoke testing and Sanity testing and usefulness of these testing for the software companies.

**Keywords-** Software testing; Smoke testing, Sanity testing

## 1. Introduction

Software testing is the process of analyzing a software to find out the differences between existing and required conditions (that is, bugs) and to evaluate the features of the software <sup>[1,2]</sup> to show that the software is performing correctly as it is intended to function without any error. Software testing is an activity that should be done throughout the development process <sup>[3]</sup>.

In this paper researchers have compared these testing types on issues like approach, implementation time and effectiveness.

### 1.1 Testing

When a software program is developed to give the real representation of an algorithm, the developers are naturally concerned with the accuracy and performance of their

output. Therefore, Developers/Testers must ensure that their software systems achieve an appropriate level of quality. Software verification is the process of ensuring that a program meets its intended specification [Kaner et al., 1993] <sup>[4]</sup>.

One technique that can assist during the specification, design, and implementation of a software system is software verification through correctness proof. Software testing, or the process of assessing the functionality and correctness of a program through execution or analysis, is another alternative for verifying a software system. As noted by Bowen, Hinchley, and Geller, software testing can be appropriately used in conjunction with correctness proofs and other types of formal approaches to develop high quality software systems [Bowen and

Hinchley 1995<sup>[5]</sup>, Geller, 1978<sup>[6]</sup>. It is also possible to use software testing techniques in isolation from program correctness proofs or other formal methods.

## 1.2 STLC cycle

Software Testing Life Cycle (STLC) process is an integral part of the Software Development Life Cycle (SDLC). The overall aspect of STLC phase deals with testing and rectifying any error code generated within the program under various test conditions. Like SDLC, STLC has fixed phases which are mentioned in hierarchy in fig.1.

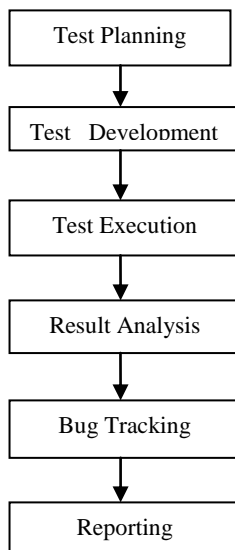


Fig. 1.1: STLC Phases

Following is the brief description of each phase of STLC <sup>[7]</sup>,

- i. **Test Planning** – Preparing the test strategy & planning
- ii. **Test Development** – Creating the testing environment
- iii. **Test Execution** – Writing the test cases/Creating & Executing the Test Script
- iv. **Result Analysis** – Analyse result and Bug report
- v. **Bug Tracking** -Analyze Bugs and application errors

- vi. **Reporting** – This is a post conditional process which involves collecting data from end users

## 1.3 Testing requirements

Software testing is performed to verify that the completed software package functions according to the expectations defined by the requirements/specifications. The overall objective of testing is not only to find every software bug that exists, but to uncover situations that could negatively impact the customer, usability and maintainability.

As per our study researchers are classifying smoke testing and sanity testing as per following fig 2'

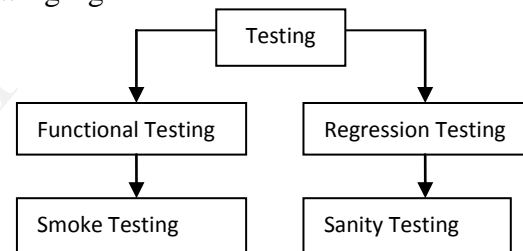


Fig. 2 Classification of Testing

## 2. SMOKE TESTING:

A smoke test is a series of test cases that are run prior to commencing with full-scale testing of an application. Smoke testing is non-exhaustive testing, that ascertains that the most crucial functions of a program work, but not bothers with finer details. <sup>[8]</sup>

In smoke testing, testing team checks only for the essential features of the product. They proceed further in the testing process if and only if the results are according to the basic requirements of the product.

If requirements do not match, then the testing team will not proceed further and return the product to development team for necessary changes. Testing will proceed further when smoke test suites succeed After

implementing smoke test tester can expect high quality of the software increase.

### 2.1 Implementing smoke Tests:

In smoke test implementation testers develop a set of test cases to check the major functionality of the product. These test cases run on new release provided by development team. The cases are higher level test cases that are used to check overall functionality of the product.

It will be more productive and efficient if the smoke test suite is automated or can be a combination of manual and automated testing.

Smoke tests are a list of about 20 to 30 test cases that address the major areas being tested.<sup>[8]</sup>

For example, if one is creating a software for a library (LMS), the smoke test suit include test cases to Add book to the system, remove old book from LMS, issues and return process, updating user records, penalty calculation, monthly report generation and new arrivals .

### 3. SANITY TESTING

K. Rungta[9] says “After receiving a software build, with minor changes in code, or functionality, Sanity testing is performed 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.”

According to Exforsys Inc. “Sanity testing is a cursory test to confirm if particular software produces the desired results or not. This implies that the software has already passed other kinds of testing before they can

undergo sanity testing. It is important to note that sanity test is not as in-depth as other kinds of testing. Sanity testing can be categorized under regression testing because the procedures for both these tests are basically the same.<sup>[10]</sup> “

### 3.1 Implementation of Sanity Testing:

Sanity testing can be performed after the first release of the software. In sanity testing tester will check the functionality of the product, after the change in any feature of the product is to find out if it is working properly or not. Same tester can create number of test cases as per the changes done in the previous release.

### 3.2 Importance of sanity testing:

1. Sanity testing is required only for the necessary output.
2. Only the changed features of the product need to be tested.
3. In depth analysis of the software's functionality is not necessary.

### 4. Usefulness of the Smoke Testing and Sanity testing for Software Industry:

A common practice at Microsoft and some other shrink-wrap software companies is the "daily build and smoke test" process. Every file is compiled, linked, and combined into an executable program every day, and the program is then put through a "smoke test," a relatively simple check to see whether the product "smokes" when it runs.<sup>[11]</sup>

Following are the major objectives of the Smoke Testing and Sanity testing

1. It minimizes integration risk
2. It reduces the risk of low quality
3. It supports easier defect diagnosis

## 5. Comparison of Smoke and Sanity Testing

Following is the comparison of smoke testing and Sanity Testing. [14]

SNo	Key Words	Smoke	Sanity
1	Documentation	A smoke test is scripted- either using a written set of tests or an automated test	A sanity test is usually unscripted.
2	Approach <sup>[21]</sup>	smoke testing is a shallow and wide approach	Sanity testing is usually narrow and deep approach.
3	Prominence <sup>[22]</sup>	It is designed to touch every part of the application in a quick way.	It basically emphasis on the small sections of the application and check whether it is working properly after a minor change.
4	Level of Testing	It checks the crucial functions of a program is working according to the specification or not, but not bothering with finer details.	It is performed to ensure that the application is functioning according to specifications. This level of testing is a subset of regression testing.
5	Example	Smoke testing is normal health check up to a build of an application before taking it to testing in depth.	sanity testing is to verify whether requirements are met or not, Checking all features breadth-first.
6	Test Cases <sup>[23]</sup>	The test cases for smoke testing of the software can be either manual or automated.	However, a sanity test is generally without test scripts or test cases
7	Performed by <sup>[23]</sup>	This testing is performed by the developers or testers	Sanity testing is usually performed by testers

Figure 1.3 Difference between smoke and sanity testing

Smoke and Sanity testing are the most misunderstood topics in Software Testing. There is enormous amount of literature on the subject, but most of them are confusing. The key differences between Smoke and Sanity Testing can be learned with the help of following figure [15]

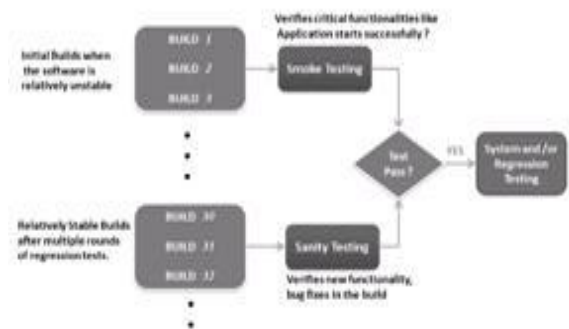


Figure 1.4 working of sanity and smoke testing

## 6. Conclusion

As per the needs of testing, tester or developer may have to execute both Sanity and Smoke Tests on the software. In such cases they 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 the terms are often confused and used interchangeably.

## Bibliography:

1. IEEE, "ANSI/IEEE Standard 1008-1987, IEEE Standard for Software Unit Testing," no., 1986
2. IEEE, "IEEE Standards Collection: Glossary of Software Engineering Terminology," IEEE Standard 610.12-1990, 1990.
3. Bertolino, "Chapter 5: Software Testing," in IEEE SWEBOK Trial Version 1.00, May 2001.
4. Cem Kaner, Jack Falk, and Hung Quoc Huyen. Testing Computer Software. International Thompson Computer Press, London, UK, 1993.
5. Jonathan P. Bowen and Michael G. Hinchley. Ten Commandments of formal methods. IEEE Computer, 28(4):56–63, April 1995.
6. Matthew Geller. Test data as an aid in proving program correctness. Communications of the ACM, 21(5):368–375, 1978.
7. "Software Testing Life Cycle (STLC) ", posted at <http://www.sdmc.ws/software-testing-life-cycle-stlc-complete-tutorial/>, August 6, 2012
8. "Smoke Testing-A Necessary Evil!" [http://www.mindlance.com/documents/test\\_management/smoke\\_testing\\_a\\_ne](http://www.mindlance.com/documents/test_management/smoke_testing_a_ne)

cessary\_evil!pdf Mindlance accessed on 18/Aug/2013.

9. K. Rungta, "Smoke and Sanity Testing - Introduction and Differences" posted at <http://www.guru99.com/smoke-sanity-testing.html> accessed on 12/08/2013.
10. Author: Exforsys Inc. "What is Sanity Testing" Published on: 23rd Jan 2012
11. S. McConnell, "Daily Build and Smoke Test", IEEE Software, Vol. 13, No. 4, July 1996 .
12. Hitesh Shah on December 4, 2011 at 2:30pm,  
<http://www.qualitytesting.info/forum/topics/smoke-vs-sanity>
13. Posted on Tuesday, August 24, 2004 11:53 AM Software Testing(<http://geekswithblogs.net/dthakur/archive/2004/08/24/10144.aspx>)
14. (<http://geekswithblogs.net/dthakur/archive/2004/08/24/10144.aspx>) accessed on 08-Aug-2013
15. "Smoke and Sanity Testing - Introduction and Differences", available at <http://www.guru99.com/smoke-sanity-testing.html> [accessed on 8-Aug-2013]