An Overview of Non-Functional Testing

Manish
Assistant Professor
Department of Computer Applications,
Chandigarh Group of Colleges
Landran, Mohali (Punjab), India

Inderpreet Kaur
Assistant Professor
Department of Computer Applications,
Chandigarh Group of Colleges
Landran, Mohali (Punjab), India

Abstract
We generally focus on the basic testing techniques available. It is a very good practice to follow the very basic Unit Testing, Integration Testing and System Testing. Taking in consideration of this generation's need, an out-of-a-box testing technique needs to be established. A bug free product is a desire of every client and developer. But along with this there are many aspects that are to be taken under consideration. They include security, page load time, recovery of system, performance of software, response time etc. My paper intends to focus on techniques to improve these aspects of testing.
Keywords: Security Testing, Automation Testing, Performance Testing, Load Testing, Stress Testing, Volume Testing, Recovery Testing, Other types of testing

Introduction
There are many ways to test a product in market. They may include automation or manual tools. A user today deals with various websites, apps, software and products. The basic testing techniques for all of them are near about the same. All of them include testing from a small unit to a full system testing. There are some other untouched aspects of testing which equally need to be taken care of. They include security, page loading, performance, quantity of data it can handle, backup etc. these are some of the major concerns. For an instance if a user is using a particular website in order to pay his bills, the user will need a very secure gateway, through which only he/she can pass through, the page loading should not collapse time to time, appropriate messages should be displayed if any information is wrong, in case the page crashes appropriate recovery of the system should take place and so on. To make sure these run time bugs do not bother the user certain techniques are highlighted in this paper.

Security Testing
Security testing attempts to verify that protection mechanisms built into a system will protect it from improper penetration. During security testing, the tester plays the role(s) of the individual who desires to penetrate the system. Security testing involves designing test cases that try to penetrate into the system using all possible mechanisms.

Security Testing (Web applications)
In case of Web applications, one has to test with appropriate firewall set-up. For data security, one

has to take into consideration Data Transfer Check-sum, Encryption or use of digital certificates, MD5 hashing on all vulnerable data and database integrity.For user security, encrypted passwords, audit trial logs containing (who, where, why, when and what information), auto log out based on system specification (e.g. 5 minutes of inactivity), display of user information on the UI can be taken care by designing the code programmatic-ally.

Automation Testing

Automation testing involves automating a manual process already in place that uses a formalized testing process.Test automation in the use of software to control the execution of tests, the comparison of actual outcomes to predicted outcomes, the setting up of test preconditions, and other test control and test reporting functions.Testers are being asked to test more and more code in less and less time. Automation testing is one way to do this, as manual testing is time consuming.Test automation is expensive and it is an addition, not a replacement, to manual testing.

Performance Testing

Performance testing is designed to test run-time performance of software within the context of an integrated system.Performance testing occurs throughout all phases testing. Even at the unit level, the performance of an individual module is assessed as white-box tests are conducted.However the performance testing is complete when all system elements are fully integrated and the true performance of the system is ascertained as per the customer requirements.

Performance testing for Web Applications:

Performance testing must be an integral part of designing, building, and maintaining web applications.Automated testing tools play a critical role in measuring, predicting, and controlling application performance.

In the most basic terms, the final goal for any Web application set for high-volume use is for users to consistently have

i) Continuous availability
ii) Consistent response times even during peak usage times.

Performance testing has five manageable phases:

i) Architecture validation
ii) Performance bench marking
iii) Performance regression
iv) Performance tuning and acceptance
v) And the continuous performance monitoring necessary to control performance and manage growth.

Load testing

Load testing is used to test if the application works fine with the loads that result from large number of simultaneous users, transactions and to determine whether it can handle peak usage periods. It measures the system's ability to handle varied workload.It is a Process of creating demand on a system or device and measuring its response time page load time, throughput etc.In load testing,

the tester attempts to load an aspect of the system to the point of failure – the goal being to determine weak points in the system.The goal of load testing is to ensure that system meets the performance baseline defined during performance testing. Thus when the system has been subjected to the maximum load that it can withstand, the regression test is performed against it and response time is measured.It helps in finding the threshold workload at which the response time will start deteriorating.When the load size approaches a particular threshold, the workload will begin to have an impact on the response time.

Stress testing

Stress tests are designed to confront programs with abnormal situations. Stress testing executes a system in a manner that demand rescues in abnormal quantity, frequency or volume. Test cases may be tailored by keeping some of the following example in view:

I) Input data rates may be increased by an order of magnitude to determine how input functions will respond.
ii) Test cases that may cause excessive hunting.
iii) Test cases that may cause thrashing in a virtual operating system may be designed.
iv) Test cases that may create disk resident data.
v) Test cases that require maximum memory or other resources may be executed.

To achieve this, the software is subjected to heavy volumes of data and the behavior is observed.

Stress testing (Web application)

This refers to testing system functionality while the system is under unusually heavy or peak load; it is similar to the validation testing but is carried out in a "high-stress" environment.This requires some idea about expected load levels of the Web application. One of the criteria for web applications would be number of concurrent users using the application.

Volume Testing:

Testing is to ensure that the software

i) Can handle the volume of data as specified in the software requirement specification.
ii) Does not crash with heavy volumes of data, but gives an appropriate message and / or makes a clean exit.

To achieve this, the software is subjected to heavy volumes of data and the behavior is observed.Examples:

i) A compiler would be fed an absurdly large source program to compile.
ii) A linkage editor might be fed a program containing thousands of modules.
iii) An operating system's job queue would be filled to capacity.
iv) If a software is supposed to handle files spanning multiple volumes, enough data are created to cause the program to switch from one volume to another. As a whole the test cases shall try to test the extreme capabilities of the programs and attempt to break the program so as to establish a sturdy system.

Recovery Testing

In s/w testing, recovery testing in the activity of testing how well the software is able to recover from crashes, hardware failures and other similar problems.

- **Mean time to failure:** The average or mean time between initial operation and the first occurrence of a failure or malfunction. In other words, the expected value of system failure time.
- **Mean time between failures:** A statistical measure of reliability, this is calculated to indicate the anticipated average time between failures. The longer the better.

Recovery testing is a system test that forces the software to fail in a variety of ways and verifies that recovery is properly performed.If recovery is automatic (performed by the system itself), re-initialization, check pointing mechanisms, data recovery, and restart are each evaluated for correctness.If recovery requires human intervention, the time required to repair is evaluated to determine whether it is within acceptable limits.

Other testing types

Documentation Testing:

Documentation testing is concerned with the accuracy of the user documentation. This involves

i) Review of the user documentation for accuracy and clarity.

ii) Testing the examples illustrated in the user documentation by preparing test cases on the basis of these examples and testing the system.

Facility Testing:Facility Testing is the determination of whether each facility (or functionality) mentioned in SRS is actually implemented.The objective is to ensure that all the functional requirements as documented in the SRS are accomplished.

Install ability testing:Certain software systems will have complicated procedures for installing the system. For instance, the system generation (sysgen) process in IBM mainframes.The testing of these installation procedures is part of system testing.Proper packaging of application, configuration of various third party software and database parameters settings are some issues important for easy installation. It may not be practical to devise test cases for certain reliability factor.For e.g., if a system has a downtime objective of two hours or less per forty years of operation, then there is no known way of testing this reliability factor.

Procedure Testing:

If the software forms a part of a large and not completely automated system, the interfaces of the developed software with the other components in the larger system shall be tested.These may include procedures to be followed by:

I) The human operator

II) Database administrator

III) Terminal

These procedures are to be tested as part of system testing.

Reliability Testing

The various software-testing processes have the goal to test the software reliability. The "Reliability Testing" which is a part of system Testing encompasses the testing of any specific reliability factor that are stated explicitly in the SRS.However, if the reliability factor are stated as

say, mean-time-to-failure (MTTF) is 20 hours, it is possible to device test cases using mathematical models.

Serviceability Testing

Serviceability testing covers the serviceability or maintainability characteristics of the software the requirement stated in the SRS may include,

i) Service aids to be provided with the system, e.g., storage-dump programs, diagnostic programs.
ii) The mean time to debug an apparent problem.
iii) The maintenance procedures for the system.
iv) The quality of the internal-logic documentation.

Test cases are to be devised to ensure the coverage of the stated aspects.

Storage Testing

Storage testing is to ensure that the storage requirements are within the specified bounds.

For instance, the amounts of the primary and secondary storage the software requires and the size of temporary files that get created.

Link testing (for web based application)

This type of testing determines if the site's links to internal and external Web pages are working.

A Web site with many links to outside sites will need regularly schedule link testing, because web sites come and go and URLs change.Sites with many internal links (such as an enterprise-wide intranet, which may have thousands internal links) may also require frequent link testing.

HTML validation (for web based application)

The need for this type of testing will be determine by the intended audience, the type of browser(s) expected to be used, whether the site delivers pages based on browser type or targets a common denominator.There should be adherence to the HTML programming guidelines as defined in to qualify.

Validation or functional testing (for web application):

This is typically a core aspect of testing to determine if the Web site functions correctly as per the requirements specifications.Sites utilizing CGI-based dynamic page generation or database-driven page generation will often require more extensive validation testing than static-page Web sites.

Extensibility Promote-ability Testing

Software can be moved from one run-time environment to another without requiring modifications to the software, e.g., the application can move from the development environment to a separate test environment.

Summary

To conclude security testing involves designing test cases that try to penetrate into the system using all possible mechanisms.Automation testing involves automating a manual process already in place that uses a formalized testing process.Load testing measures the system ability to handle varied workloads. Stress Testing refers to testing system functionality while the system is under unusually heavy or peak load; it is similar to the validation testing but is carried out in a "high-stress" environment. Volume Testing is to ensure that the software can handle the volume of data as specified in the Software Requirement Specifications and does not crash with heavy volumes of data, but gives an appropriate message and/ or makes a clean exit. Recovery testing in the activity of testing how well the software is able to recover from crashes, hardware failures and other similar problems.

References

[1] http://istqbexamcertification.com/what-is-fundamental-test-process-in-software-testing/
[2] http://www.softwaretestinghelp.com/
[3] Introduction to SQA & Testing Vol. I by BTES
[4] Lessons Learned in Software Testing, by C. Kaner, J. Bach, and B. Pettichord
[5] Testing Computer Software, by C. Kaner, J. Falk, and H. Nguyen