



Non-functional System Testing

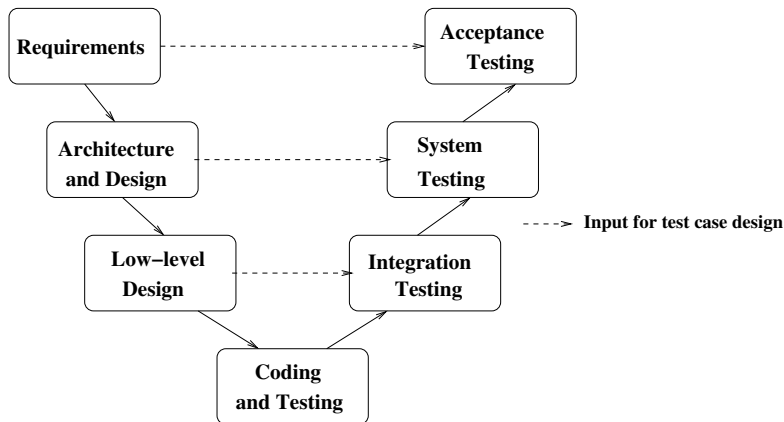
Meenakshi D'Souza

International Institute of Information Technology Bangalore.

Overview

- System testing: Taxonomy.
- Non-functional system level testing.
- Performance testing.

V-model for testing: Re-cap



System Testing: Two categories

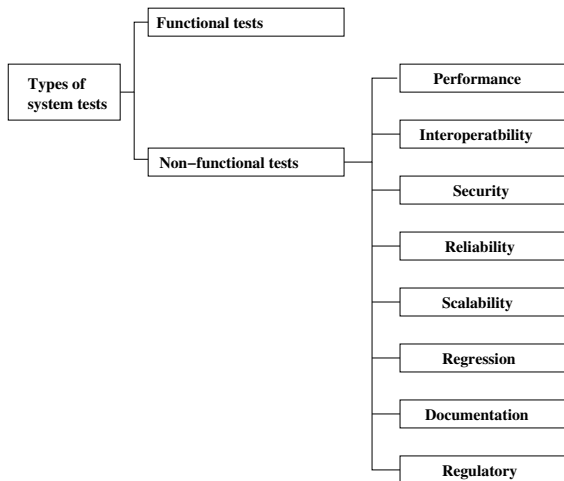
There are two categories of system testing that are relevant for this lecture:

- Functional testing.
- Non-functional testing.

Functional testing spans over the remaining phases too.

Non-functional testing begins at system testing phase only.

A taxonomy of system tests



Interoperability testing

- **Interoperability testing** determines whether the system can inter-operate with other third-party products.
- Could involve **compatibility testing** too.
- Compatibility testing involves testing for compatibility with different operating systems, different browsers, different database servers etc.
- Two kinds of compatibility: **forward** and **backward** compatibility.

Security testing

- **Security testing** determines if a system protects data and maintains security related functionality as intended.
- Aims at testing the following:
 - **Confidentiality**: Requirement that data and processes be protected from *unauthorized disclosure*.
 - **Integrity**: Requirement that data and processes be protected from *unauthorized modification*.
 - **Availability**: Requirement that data and processes be protected from *denial of service to authorized users*.
 - It also deals with **authorization** and **authentication** verification.

Security testing: Types/techniques

Types of security testing include the following:

- Verify that only authorized accesses to the system are permitted.
- Verify the correctness of both encryption and decryption algorithms for systems where data/messages are encoded.
- Verify that illegal reading of files, to which the perpetrator is not authorized is not allowed.
- Ensure that virus checkers prevent/curtail entry of viruses into the system.
- Identify *back doors* in the system left open by developers (e.g., buffer overflows). Test for access through back doors.
- Verify different authentication, client-server communication, wireless security protocols etc.

Reliability tests

- **Reliability tests** *measure* the ability of system to keep operating over specified periods (typically several months/years) of time.
- Includes both hardware and software reliability.
- In-depth mathematical analysis techniques are used, a separate area worth delving into.

Scalability tests

- **Scalability testing** verifies that a system can scale up to its engineering limits.
- This involves testing the *limit* of the system, that is, the magnitude of demand that can be placed on the system while continuing to meet its latency, response time and throughput requirements.
- Major causes on limitations to scale up:
 - Data storage limitations
 - Network bandwidth limitations
 - Speed limit (CPU speed)
- Tests are performed by extrapolating basic data used for scalability testing.

Documentation testing

- **Documentation testing** is done to verify the technical accuracy and readability of various documentation including user manuals, tutorials, on-line help etc.
- Usually performed at three levels:
 - **Read test**: A documentation is reviewed for clarity, organization, flow and accuracy without executing the documented instructions on the system.
 - **Hands-on test**: On-line help is exercised and the error messages verified to evaluate their accuracy and usefulness.
 - **Functional test**: Instructions embodied in the documentation are followed to verify that the system works as it has been documented.

Documentation testing

Some of the tests recommended for documentation testing are:

- Read all the documentation to verify correct use of grammar and consistent use of technical terms.
- Use of graphics, images etc. as appropriate.
- Verify the glossary accompanying the documentation uses a standard, commonly accepted terminology and the glossary correctly defines the terms.
- Verify that there is a proper index for terms, verify the page numbers of the index.
- Verify that on-line and printed versions of the documentation are same.
- Verify the installation procedure by executing the steps in the actual system.
- Verify trouble shooting guide with actual error scenarios.

Regulatory testing

- Each country has regulatory bodies guiding the availability of a product in that country.
- CE (Conformite Europeene), CSA (Canadian Standards Association), FCC (Federal Communications Commission) etc. are some of them.
- In addition, safety critical systems have their own regulatory requirements, per, domain.
 - Aerospace: ARP standards, DO standards by RTCA etc.
 - Automotive: IEC 61508, MISRA guidelines etc.
- Exhaustive testing and documentation is done throughout the development to cater to these regulatory standards.

Performance testing

- **Performance testing** is done to determine the system parameters in terms of *responsiveness* and *stability* under various workloads.
- Performance testing measures the quality attributes of the system, such as scalability, reliability and resource usage.
- For e.g., an expected performance could be “A transaction in an on-line system requires a response of less than 1 second 90% of the time”.
- Tests measure the performance of the actual system compared to the expected quality parameters.

Performance testing techniques

- **Load testing:** It is the simplest form of testing conducted to understand the behaviour of the system under a specific load. Load testing will result in measuring important business critical transactions and load on the database, application server, etc., are monitored.
- **Stress testing:** It is performed to find the upper limit capacity of the system and also to determine how the system performs if the current load goes well above the expected maximum.

Performance testing techniques

- **Soak testing**: Soak Testing also known as **endurance testing**, is performed to determine the system parameters under continuous expected load. During soak tests the parameters such as memory utilization is monitored to detect memory leaks or other performance issues.
- **Spike testing**: Spike testing is performed by increasing the number of users suddenly by a very large amount and measuring the performance of the system. The main aim is to determine whether the system will be able to sustain the workload.

Goals and metrics in performance testing

For performance testing, one needs to be clear about the goals and specific data to be captured.

- Performance testing goals:
 - Demonstrate that the system meets performance criteria.
 - Compare two systems to find which performs better.
 - Identify and measure which parts of the system or workload cause the system to perform badly.
- Performance tests typically always include the following:
 - Number of concurrent users.
 - Throughput or transaction rate.
 - (Server) response time.
 - Any other specific measures.

Performance testing: Tools

The following are some of the tools that can create and run [performance test scripts](#).

- HP LoadRunner.
- NeoLoad: For web and mobile applications.
- Apache JMeter: Good for web applications, open source.
- Rational Performance Tester.
- Gatling: Good for web applications, open source.

Performance monitoring

- The system running the generated test scripts needs to be monitored to observe the intended performance parameters.
- All the decided parameters are monitored with help of special scripts or external devices hooked to the system in which the test scripts are running.

What we didn't look at?

We didn't cover the following topics:

- Usability testing
- Testing of GUI
- Security
- Reliability

There is fair amount of material available in the above areas, still active research is being done.