

Performance Testing

- 1. Performance testing can be very expensive and time-consuming to perform.
- 2. It can be especially costly to establish the correct environment to test in.
- 3. In terms of hardware, operating system, middleware and concurrent use patterns or operational profiles.
- 4. As for all testing, failures must be reported, and retest and regression test must be performed after corrections.

Load Testing :

- 1. Load testing is a special subtype of performance testing concerned with the product's behavior under specified load conditions.
- 2. Two aspects namely:
 - 1. Multiuser with ordinary realistic numbers of users.
 - 2. Large, though still realistic number of concurrent users.
- 3. The loads that the product is expected to be able to handle and the applicable response times must be expressed in load requirements.
- 4. Load testing can be quite expensive.
- 5. Tools may be used to generate or simulate loads.

Stress Testing :

- 1. Stress is an expression of the product's capability for handling extreme situations.
- 2. We are dealing with risks concerning people, money, data and the environment.
- 3. The system shall not crash but issue an error message and stop execution after acknowledgment.
- 4. Stress-handling is particularly important for web applications, such as e-products and e-business in telecommunication, safety and security critical systems and real-time systems.

Portability Testing:

- 1. In portability testing we test the requirements or needs concerned with the product's ability to be transferred into its intended environment.
- 2. The environment may include the organization in which the product is used and the hardware, software and network environment.
- 3. Portability is primarily an issue for software products or software subsystems.

Ex: hardware subsystems.

Unit-5

What is Interoperability testing?????

- 1. Interoperability testing is a form of non functional testing. This form of testing is done to ensure end to end functionality between two interacting systems based on their specified standards and protocols

. In interoperability testing we test the requirements or needs concerned with the ability of our software system to interact with other specified systems

- 2. No software system stands alone, it will always have to interact with other systems in the intended deployment environment.
- 3. Such as h/w, s/w like operating systems, database systems, browsers and network facilities.
- It take place at the system integration testing level

Functional Security Testing:

- 1. Security testing can be split into functional security testing and technical security testing
- 2. In functional security testing we test the fulfillment of security attributes that can be explicitly expressed in requirements.

Ex: The system shall ensure that only users with extended valid privilege may see the full hospitalization history for a patient.

What is USABILITY TESTING???

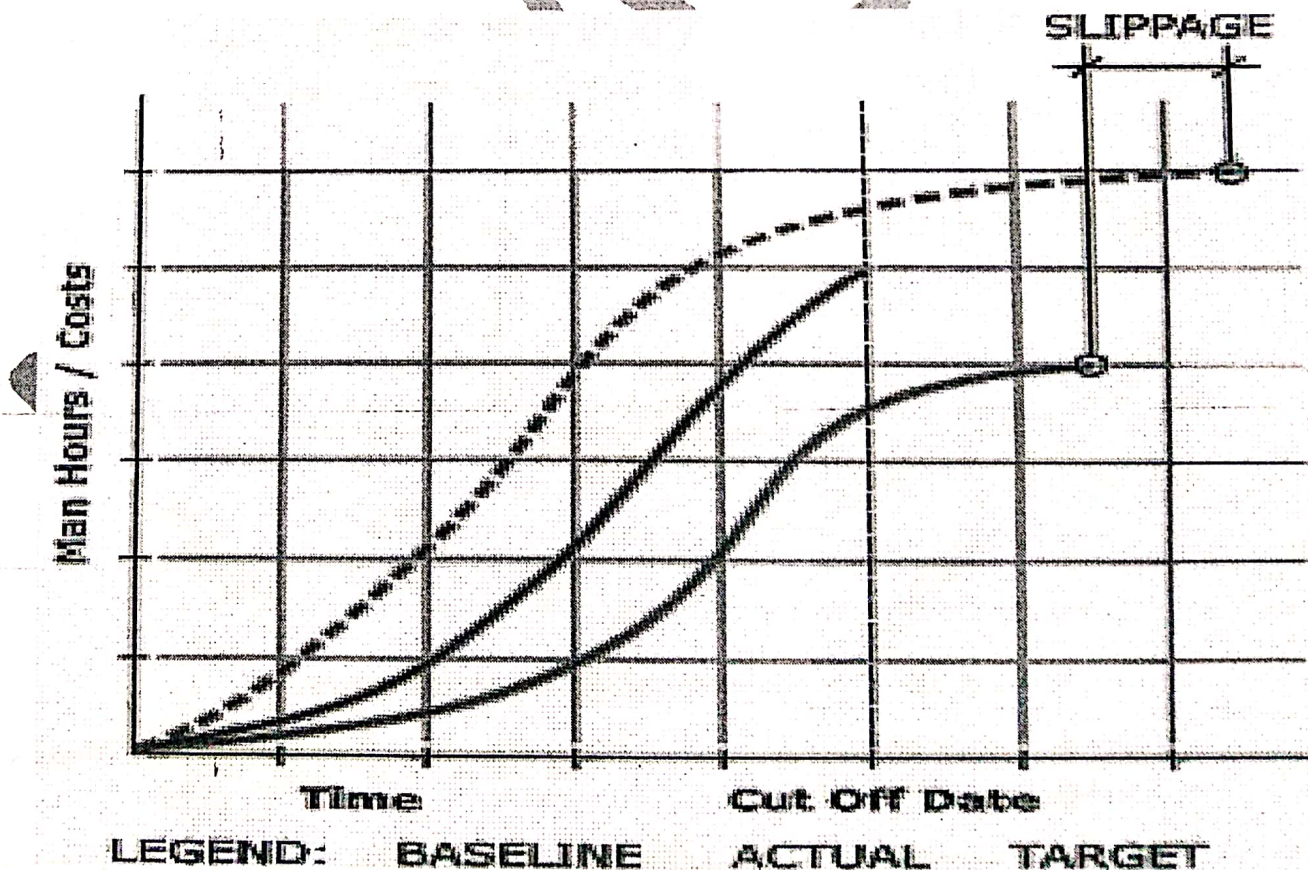
- 1. Usability is the suitability of the software for its users.
- In terms of effectiveness, efficiency and satisfaction with which specified users can achieve specified goals in particular environments.
- The effectiveness of a software product is its capability to enable users to achieve specified goals with accuracy and completeness.
- The satisfaction of a product is its capability to satisfy users.
- 5. User groups may also consist of those in charge of installing the product and those monitoring and maintaining it.
- 6. Groups getting information from the product.
- Ex: in the form of reports or letters.
- 7. For each of the user groups it is necessary to look at different characteristics.
- Ex: 1. Age (e.g. preschool, Children, teens, young, adults, mature adults and elderly).
- Education (e.g. no education yet, illiterate, basic education, middle education, workman, university education).
- Disabilities (e.g. blind, partially sighted, deaf, mobility – impaired).
- **Usability Subattributes**
- 1. The ISO 9126 standard classifies usability as a nonfunctional quality attribute.
- 2. How the functionality presents itself to the users.
- 3. Usability covers much more than just the look and feel of the product.

Reliability Testing:

- 1. Reliability evaluation or reliability estimation is an activity.
- 2. Where we analyze the fault finding curves.
- 3. we extrapolate from the curves and predict how many defects are left in the product.
- 4. To predict remaining defects is to use estimation models based on the structure of the program.

Ex: Knowledge of the size, the complexity and the data transactions

- **Why Use an S-curve?**
- S-curves are an important project management tool. They allow the progress of a project to be tracked visually over time, and form a historical record of what has happened to date. Analyses of S-curves allow project managers to quickly identify project growth, slippage, and potential problems that could adversely impact the project if no remedial action is taken.



- Robustness Testing:

- Robustness testing is any quality assurance methodology focused on testing the robustness of software. ... ANSI and IEEE have defined robustness as the degree to which a system or component can function correctly in the presence of invalid inputs or stressful environmental conditions

Robustness Testing with Fault Injection



IV&V Facility

- Robustness - the degree to which a system operates correctly in the presence of ***exceptional inputs*** or stressful environmental conditions
- The goal of robustness testing is to develop test cases and test environments where a system's robustness can be assessed.

2

- 1. Fault tolerance or robustness is the product's ability to maintain a specified level of performance in the presence of s/w defects.
- 2. Reactions to failures of a given severity.
- 3. Reactions to failures of a given severity.
- 4. Self-monitoring of the operations and self identification of defects.
- 5. Loss of specified operations in case of failure of specified severities in specified periods of time for specified parts of the system.
- 6. Loss of specified data in case of failure of specified severities in specified periods of time for specified parts of the system.

- Ex: Failures from external sources could, for example, be caused by lack of external storage capacity, external data storage not found, external services not available or lack of memory.



QUALITY COMPONENTS - Robustness

Jyrki Nummenmaa
University of Tampere, CS Department

- A program is robust, if it behaves reasonably (?) well even in unexpected circumstances - i.e. it tolerates unexpected difficulties.
- Dealing with errors? E.g. program input is often different from what is expected.
- The program may be reliable in a user's point of view even if it is not correct.
- A crucial property in some applications.

Software Engineering – <http://www.cs.uta.fi/se>



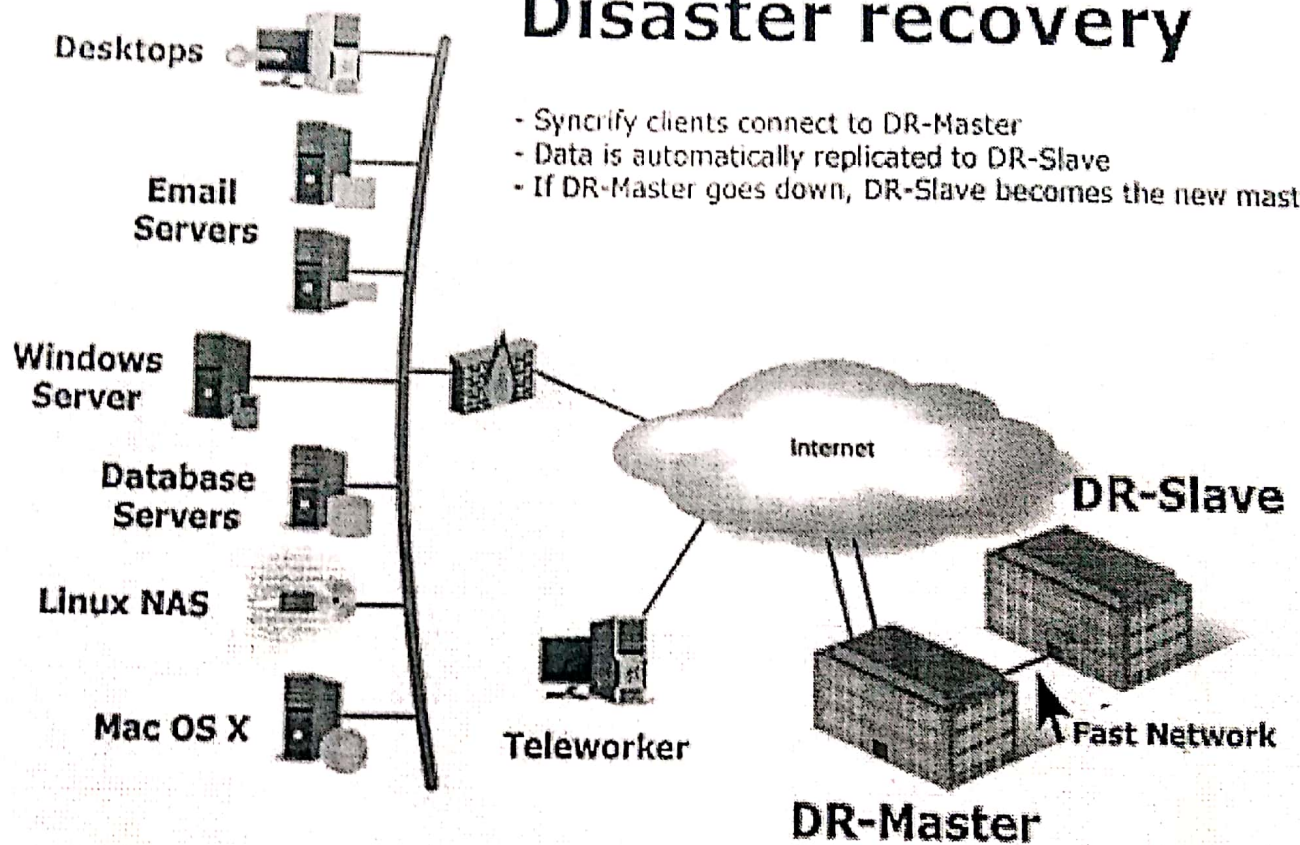
Recoverability Testing:

- In software testing, recovery testing is the activity of testing how well an application is able to recover from crashes, hardware failures and other similar problems. Recovery testing is the forced failure of the software in a variety of ways to verify that recovery is properly performed.

Recoverability Testing

- Crash recovery and transaction interruptions
- Validation of the effective application recovery situation post unexpected interruption/crash scenarios.
- Verification of how the application handles a transaction during a power failure (i.e. Battery dies or a sudden manual shutdown of the device)
- The validation of the process where the connection is suspended, the system needs to re-establish for recovering the data directly affected by the suspended connection.

Disaster recovery



-
- What is efficiency?
- Definition: **Efficiency testing** tests the amount of resources required by a program to perform a specific function. In software companies, this term is used to show the effort put in to develop the application and to quantify its user-satisfaction.

Test Efficiency (TE)

- ▶ The Test Efficiency metric as stated by Lokesh Gulechha. (2009) is performed to determine the overall efficiency of the software testing team in identifying the defective components of the software.
- ▶ The Test Efficiency metric is also used to indicate the number of defects which were missed out during testing phase and which migrated to the next testing phase.
- ▶ The Test Efficiency of the testing process is calculated by the following formula:

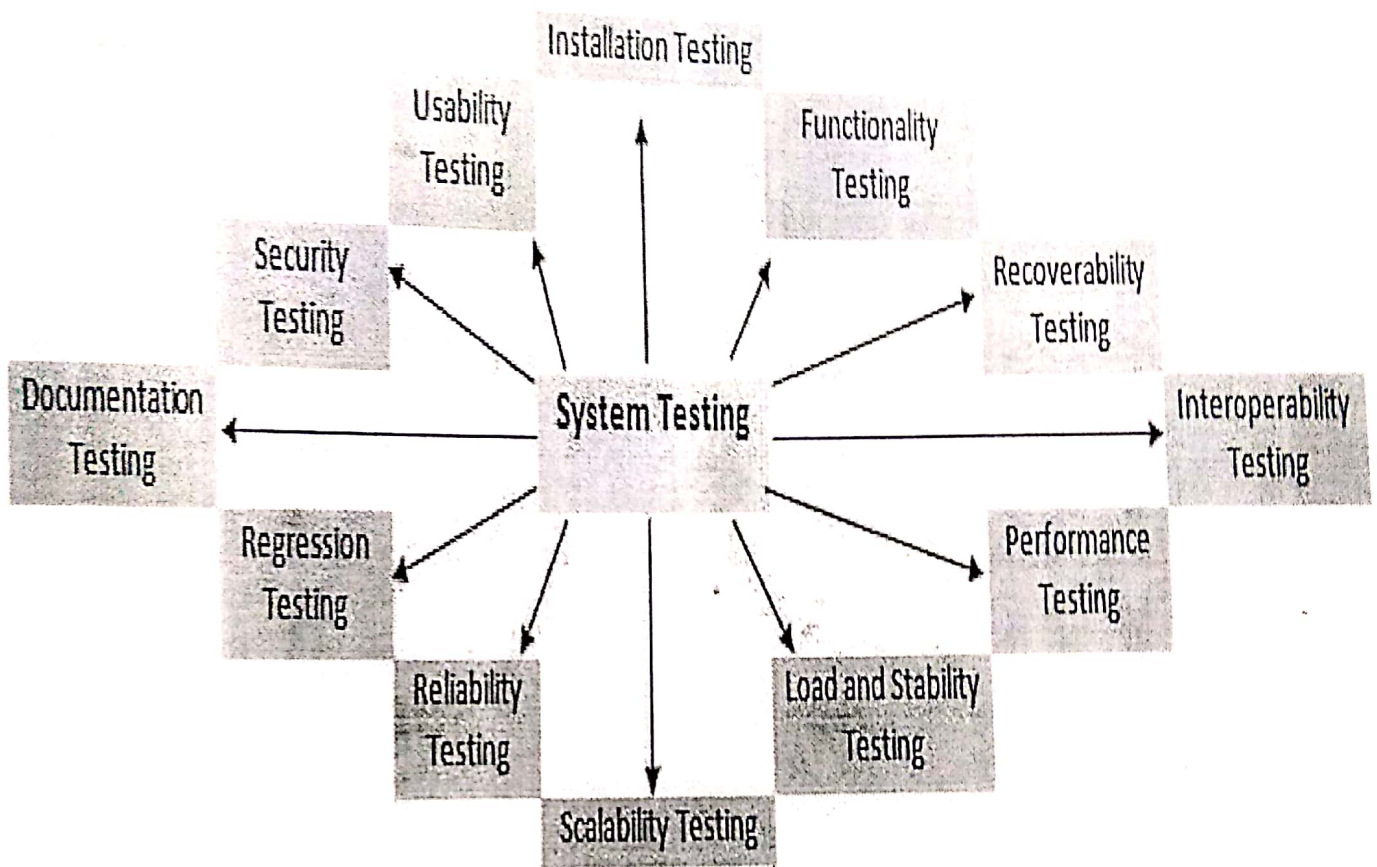
$$\text{Test Efficiency} = \text{DT} / (\text{DT} + \text{DU}) * 100\%$$

- ▶ Where,

DT = The overall number of valid defects identified during the testing process.

DU = The number of valid defects which have been identified by test user after the release of software application. In other words, post-testing defect

- ▶ The Test Efficiency metric should be gathered by the software tester because it gives a total percentage of the defective parts which were missed by the tester in the testing process for the software component and therefore this process will lead to better efficiency by the software tester when testing future software components.



System Testing - © www.SoftwareTestingHelp.com

