SWE 4603

Software Testing and Quality Assurance

Lecture 7

Prepared By Maliha Noushin Raida, Lecturer, CSE
Islamic University of Technology

Lesson Outcome

- Non-functionality Testing
  - Performance Testing
  - Compatibility Testing
  - Compliance Testing
  - Recovery Testing
  - Security Testing
  - Usability Testing

Non-functional Testing

- **Non-Functional Testing** is defined as a type of Software testing to check non-functional aspects (performance, usability, reliability, etc) of a software application.
- An excellent example of non-functional test would be to check how many people can simultaneously login into a software.
- Non-functional testing is equally important as functional testing and affects client satisfaction.

The Major Non-Functional Testing techniques include:

- Performance testing
- Compatibility testing
- Compliance testing
- Recovery testing
- Security testing
- Usability testing

Performance Testing

Performance Testing

- is a type of software testing that ensures that the software applications will perform well under their expected workload.
- The goal of performance testing is not to find bugs but to **eliminate performance bottlenecks**.
- Performance testing measures the quality attributes of the system, such as scalability, reliability and resource usage.

Performance Testing

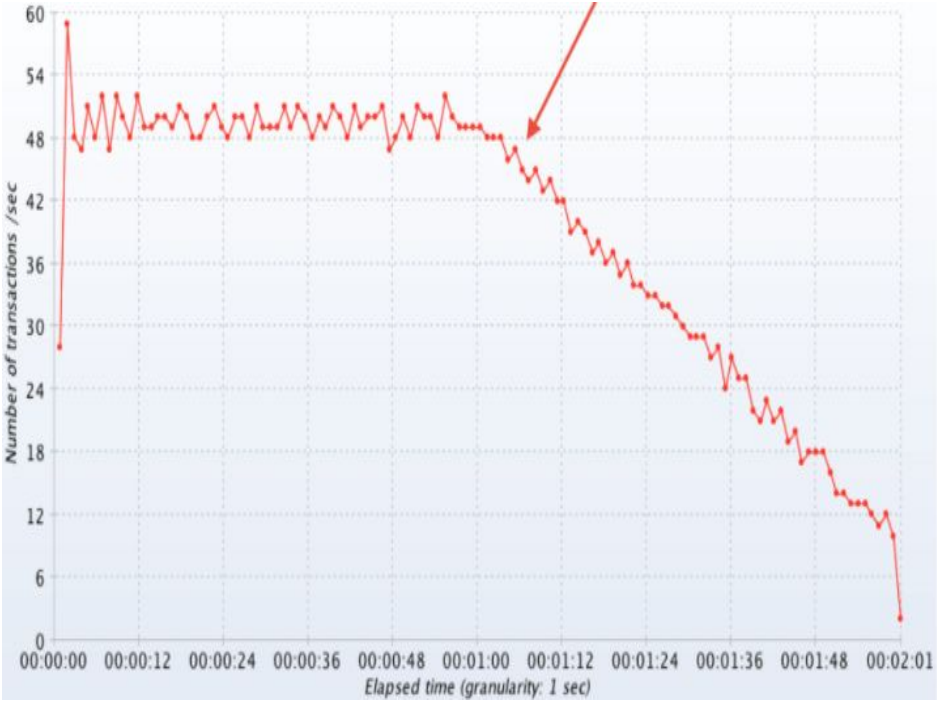
Some of the factors that governs Performance testing:

- Throughput
- Response Time
- Tuning
- Benchmarking

Performance Testing

Throughput:

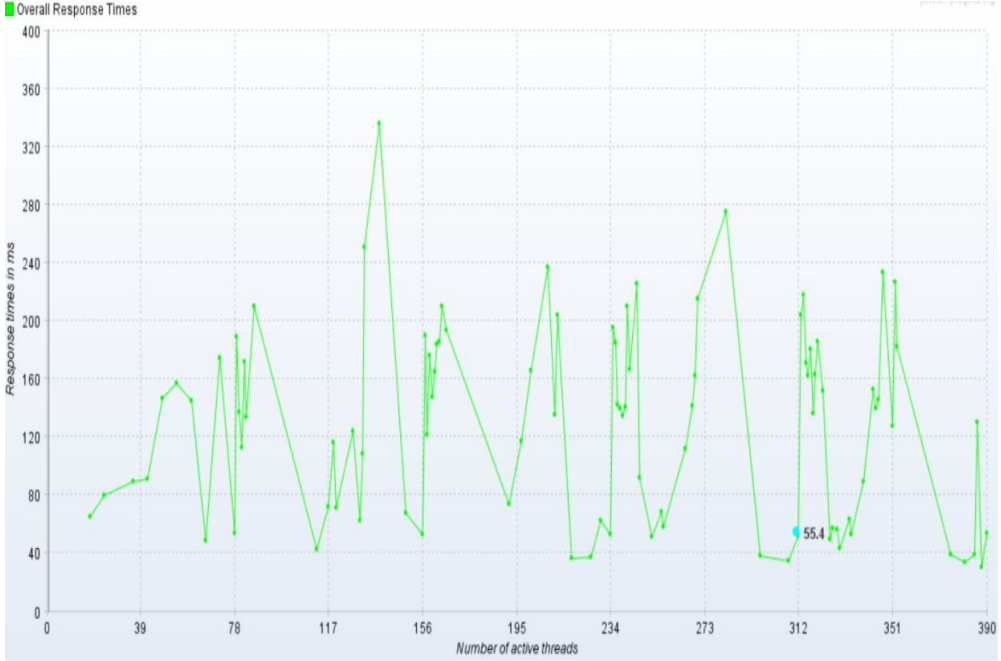
- Capability of a product to handle multiple transactions in a given period.
- Throughput represents the number of requests/business transactions processed by a product in a specified time duration [typically measured as transaction per second].
- It is one of the significant indicator that helps in evaluating the performance of application.



Performance Testing

Response Time:

- It is equally important to find out how much time each of the transactions took to complete.
- Response time is defined as the delay between the point of request and the first response from the product.
- The response time increases proportionally to the user load.



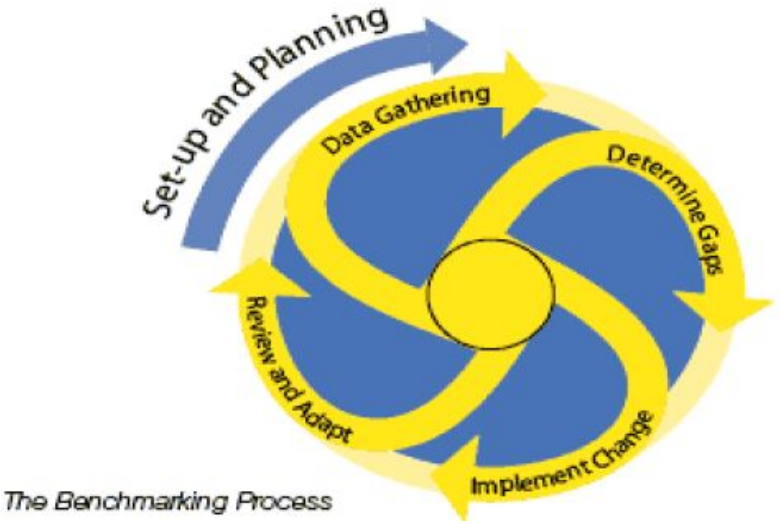
Performance Testing

Tuning:

- Tuning is the procedure by which product performance is enhanced by setting different values to the parameters of the product, operating system and other components.

Benchmarking:

- A very well-improved performance of a product makes no business sense if that performance does not match up to the competitive products.
- A careful analysis is needed to chalk out the list of transactions to be compared across products so that an **apple-apple comparison** becomes possible.



The Benchmarking Process

Performance Testing

Finally

- The testing to evaluate the response time (speed), throughput and utilization of system to execute its required functions in comparison with different versions of the same product or a different competitive product is called Performance Testing.
- Performance testing is done to derive benchmark numbers for the system.
- Tuning is performed until the system under test achieves the expected levels of performance.

Performance Testing

Why Performance testing???

- Identifies problems early on before they become costly to resolve.
- Reduces development cycles.
- Produces better quality, more scalable code.
- Prevents revenue and credibility loss due to poor Web site performance.
- Enables intelligent planning for future expansion.
- To ensure that the system meets performance expectations such as response time, throughput etc. under given levels of load.

Performance Testing

What should be tested???

High frequency transactions:

- The most frequently used transactions have the potential to impact the performance of all of the other transactions if they are not efficient.

Mission Critical transactions:

- The more important transactions that facilitate the core objectives of the system should be included, as failure under load of these transactions has, by definition, the greatest impact.

Read Transactions:

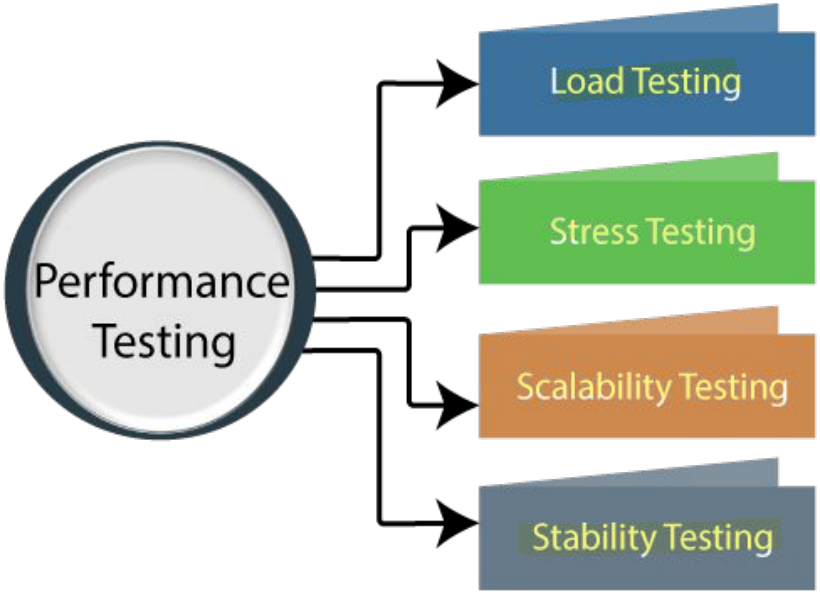
- At least one READ ONLY transaction should be included, so that performance of such transactions can be differentiated from other more complex transactions.

Update Transactions:

- At least one update transaction should be included so that performance of such transactions can be differentiated from other transactions.

Performance Testing

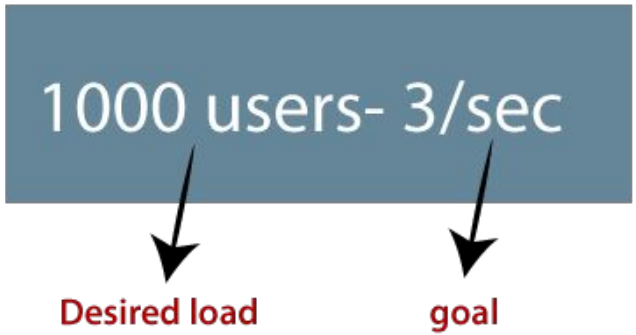
Types of Performance Testing:



Performance Testing

Load Testing

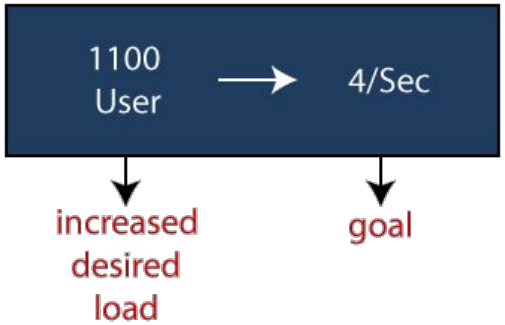
- Load testing is performance testing technique using which the response of the system is measured under various load conditions.
- The load testing is performed for **normal and peak load conditions**.
- It is used to check the performance of an application by applying some load which is either less than or equal to the **desired load**.



Performance Testing

Stress Testing:

- Stress testing a Non-Functional testing technique that is performed as part of performance testing.
- During stress testing, the system is monitored after subjecting the system to overload to ensure that the system can sustain the stress.
- That is with stress testing we check the behavior of an application by applying **load greater than the desired load**.
- The recovery of the system from such phase (after stress) is very critical as it is highly likely to happen in production environment.



Performance Testing

Reasons for conducting Stress Testing:

- It allows the test team to monitor system performance during failures/heavy load.
- To verify if the system has saved the data before crashing or NOT.
- To verify if the system prints meaning error messages while crashing or did it print some random exceptions.
- To verify if unexpected failures do not cause security issues.

Performance Testing

Scalability Testing:

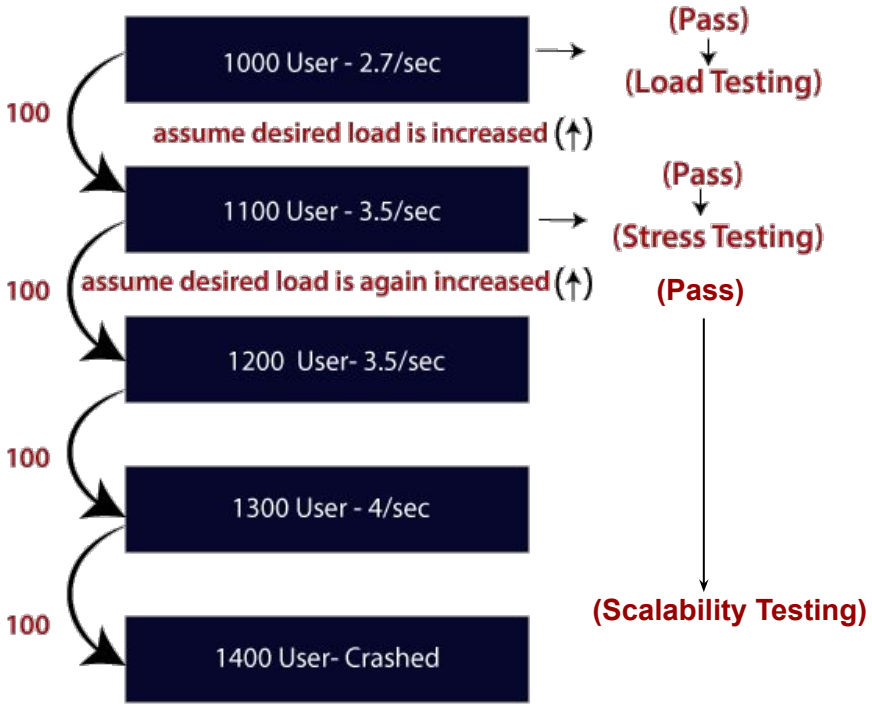
- Checking the performance of an application by increasing or decreasing the load in particular scales (no of a user) is known as scalability testing.
- Scalability testing is divided into two parts which are as follows:
  - **Upward scalability testing:** It is testing where we increase the number of users on a particular scale until we get a crash point. We will use upward scalability testing to find the maximum capacity of an application.
  - **Downward scalability testing:** The downward scalability testing is used when the load testing is not passed, then start decreasing the no. of users in a particular interval until the goal is achieved. So that it is easy to identify the bottleneck (bug).

Performance Testing

Stability Testing:

- Checking the performance of an application by applying the load for a particular duration of time is known as Stability Testing.
- It measures efficiency and ability of a software application to continuously function over a long period of time.
- The possible error can be faced,
  - the system slows down
  - the system encounters functionality problems
  - the system shows weird behavior
  - the system crashes altogether

Performance Testing



Problems in Performance Testing

Response Time

- how quickly the server respond to the client's request.
- If the user's request does not complete in the given response time, the user may be lost his/her interest in the particular software or application.

Scalability

- occur when the application cannot take the **n-numbers** of users and expected user requests at the same time.

Bottleneck

- It occurs when the application is limited by a single component and creates a bad impact on the system performance.
- most common bottlenecks:
  - Memory utilization
  - Disk usage
  - CPU utilization
  - Operating System limitations
  - Network utilization

Speed

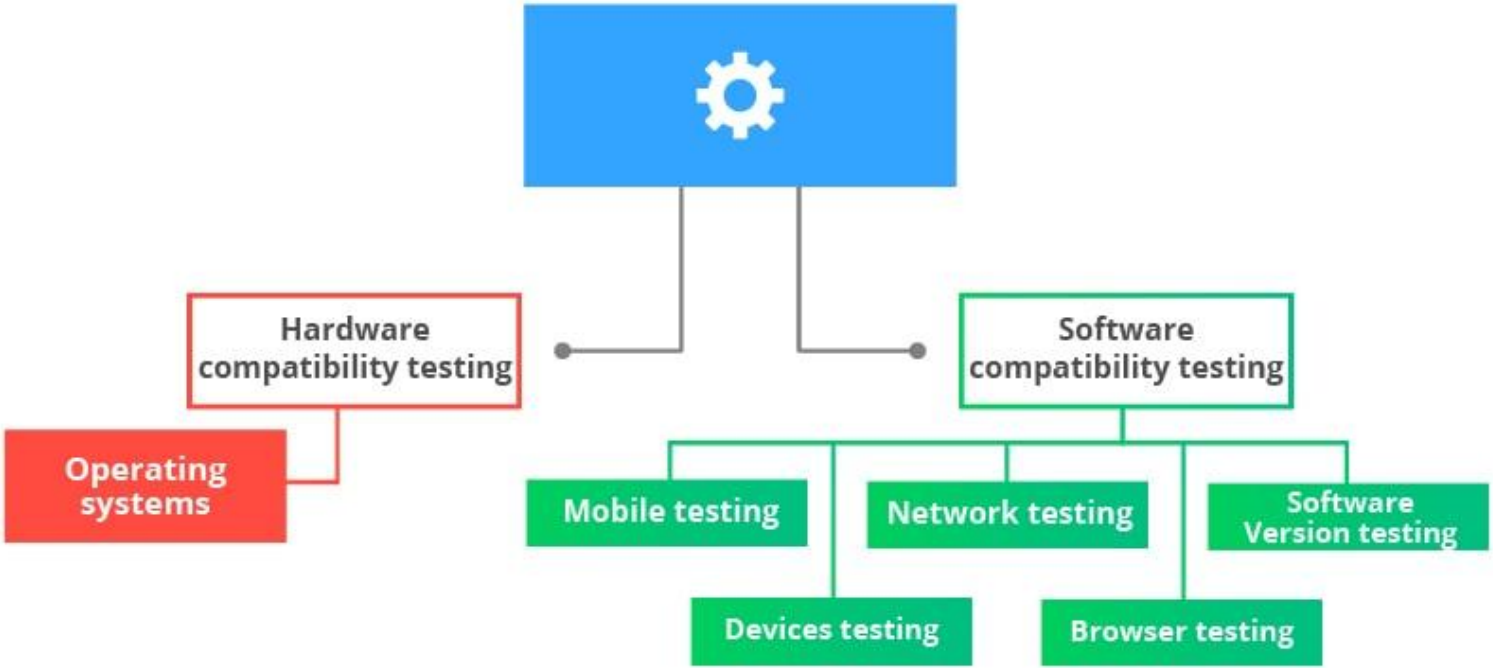
- When we perform performance testing on the application, the application should be faster in speed to get the user's interest and attention.

Compatibility Testing

Compatibility Testing

- Sometimes known as platform testing
- Compatibility Testing is a type of Non-functional testing
- Compatibility Testing is a type of Software testing to check whether your software is capable of running on different hardware, operating systems, applications, network environments or Mobile devices.

Compatibility Testing



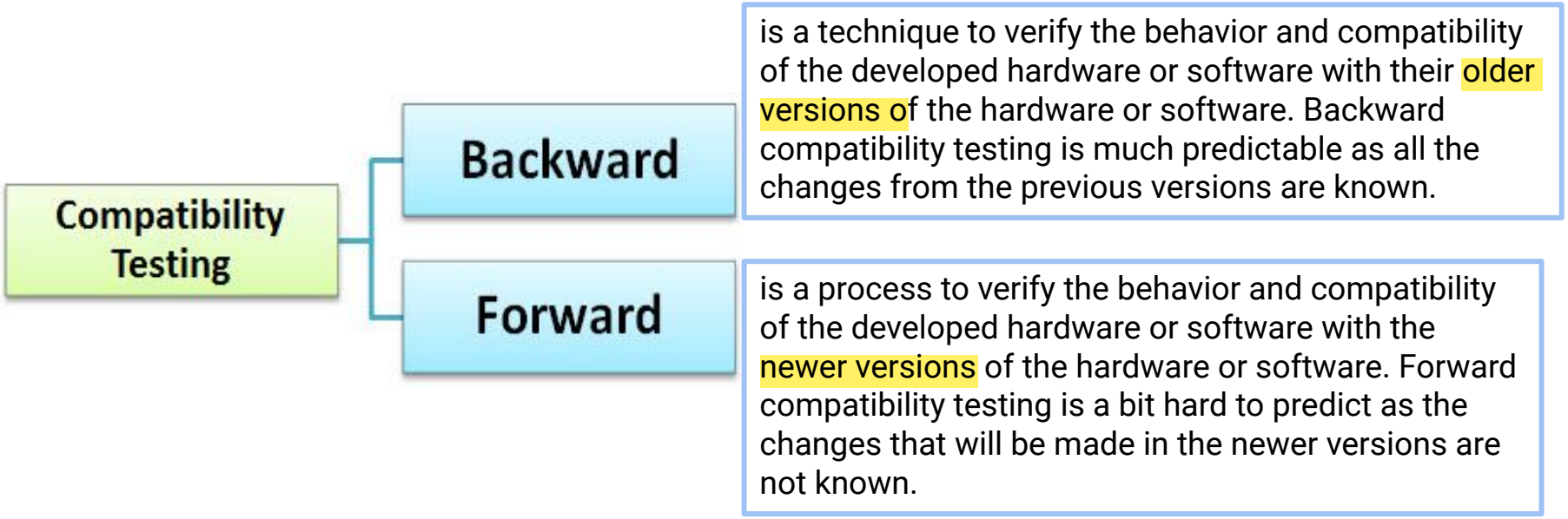
Compatibility Testing

Types of Compatibility Tests

- **Hardware:** It checks software to be compatible with different hardware configurations.
- **Operating Systems**: It checks your software to be compatible with different Operating Systems like Windows, Unix, Mac OS etc.
- **Network:** Evaluation of performance of a system in a network with varying parameters such as Bandwidth, Operating speed, Capacity. It also checks application in different networks with all parameters mentioned earlier.
- **Browser:** checks the compatibility of website with different browsers like Firefox, Google Chrome, Internet Explorer etc.
- **Devices:** checks compatibility of software with different devices like USB port Devices, Printers and Scanners, Other media devices and Bluetooth.
- **Mobile:** Checking software is compatible with mobile platforms like Android, iOS etc.
- **Versions of the software:** It is verifying your software application to be compatible with different versions of the software. For instance checking your Microsoft Word to be compatible with Windows 7, Windows 7 SP1, Windows 7 SP2, Windows 7 SP3.

Compatibility Testing

There are two types of version checking in Compatibility Testing :



Compatibility Testing

Benefits of Compatibility Testing:

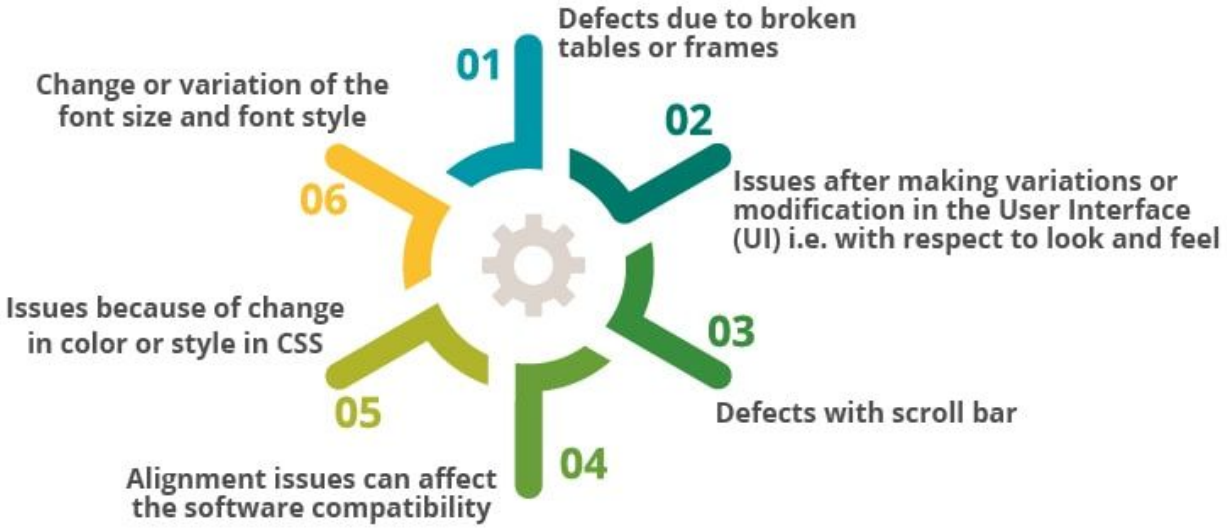
- Easy to validate application's stability, usability, and scalability across various platforms and deliver feedback.

- ensures to meet the fundamental expectation of users.
- A compatibility tests are performed on the app, it ensures to deliver best customer experience.

- capable of timely identification of the defects
- Since defects are identified earlier before production, the testing practice gives enough time for the developer teams to fix them at the earliest.

Compatibility Testing

Common Defects encountered during Compatibility Testing:



Compliance Testing

Compliance Testing

- Compliance Testing is performed to maintain and validate the compliant state for the life of the software. Every industry has a regulatory and compliance board that protects the end users.
- Example: The software used in the pharmaceutical industry, the Food and Drug Administration (FDA) regulation comes into the picture.
- Checklists for Compliance Testing:
  - Professionals, who are knowledgeable and experienced, who understand the compliance must be retained.
  - Understanding the risks and impacts of being non-compliant
  - Document the processes and follow them
  - Perform an internal audit and follow with an action plan to fix the issues

Recovery Testing

Recovery Testing

- Recovery testing is a type of non-functional testing technique performed in order to determine how quickly the system can recover after it has gone through system crash or hardware failure.
- Recovery testing is the forced failure of the software to verify if the recovery is successful.

Recovery Testing

Recovery Plan – Steps

1. Determining the feasibility of the recovery process.
2. Verification of the backup facilities.
3. Ensuring proper steps are documented to verify the compatibility of backup facilities.
4. Providing Training within the team.
5. Demonstrating the ability of the organization to recover from all critical failures.
6. Maintaining and updating the recovery plan at regular intervals.

Security Testing

Security Testing

- Security testing is a testing technique to determine if an information system protects data and maintains functionality as intended.
- Process intended to reveal flaws in the security mechanisms of an information system
- Finding out the potential loopholes & weakness of the system
- To check whether there is an information leakage
- Passing Security Testing is not an indication that no flaws exist

Security Testing

Aims to verify 6 basic principles as listed below:

- Confidentiality – Is my secret safe?
- Integrity – Is my data being tampered?
- Authentication – Who am I? Something you know!! Something you have!
- Authorization – What can I do?
- Availability –Do the information ready for use when expected?
- Non-repudiation - prevent the later denial that an action happened, or a communication that took place

Security Testing

- Security Testing Techniques
  - Vulnerability Scanning
  - Security Scanning
  - Penetration Testing
  - Ethical Hacking
  - Risk Assessment
  - Security Auditing
  - Posture Assessment & Security Testing
  - Password cracking

Security Testing

Security Testing Techniques

- **Vulnerability Scanning** : It involves scanning of the application for all known **vulnerabilities**.
  A computer program designed to assess computers, computer systems, networks or applications for weaknesses. Generally done through various vulnerability scanning software. Ex : Nessus, Sara, and ISS.

- **Security Scanning** : Scanning and verification of the system and applications. Find out the weaknesses in the OS, applications and networks.

- **Penetration Testing**: Tester may try to enter into the application / system with the help of some other application or with the help of combinations of loopholes that the application has kept open unknowingly. It is the most effective way to practically find out potential loopholes in the application.

- **Ethical Hacking**: Ethical Hacking involves number of penetration tests over the wide network on the system under test. It is conducted by ethical hackers to find possible problems in the system.

Security Testing

Security Testing Techniques

- **Risk Assessment**: Is a method of analyzing and deciding the risk that depends upon the type of loss and the possibility / probability of loss occurrence. Risk assessment is carried out in the form of various interviews, discussions and analysis of the same.
- **Security Auditing**: Security Auditing involves hands on internal inspection of Operating Systems and Applications, often via line-by-line inspection of the code. A security audit is a systematic evaluation of the security of a company's information system.
- **Posture Assessment and Security Testing** : It combines Security Scanning, Ethical Hacking and Risk Assessments to show an overall Security Posture of the organization.
- **Password Cracking** : Password cracking programs can be used to identify weak passwords. Password cracking verifies that users are employing sufficiently strong passwords.

Usability Testing

Usability Testing

What is Usability?

Usability Is a measure of how easy it is to use something:

- How easy the SW will be for a typical user to understand, learn, and operate
- e.g., “user-friendliness”

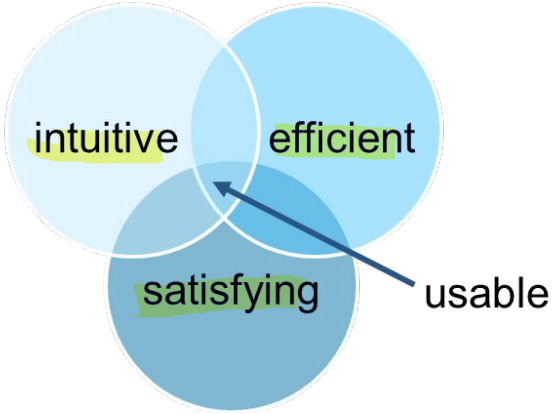
Usability is difficult to evaluate and measure

Usability Testing

- Usability testing, a non-functional testing technique that is a measure of how easily the system can be used by end users.

ISO 9241-11 (3.1):Definition of Useability

‘the extent to which a product can be used by specified users to achieve goals with effectiveness, efficiency and satisfaction in a specified context of use’



Usability Testing

- It is difficult to evaluate and measure but can be evaluated based on the below parameters:
  - Level of Skill required to learn/use the software. It should maintain the balance for both novice and expert user.
  - Time required to get used to in using the software.
  - The measure of increase in user productivity if any.
  - Assessment of a user's attitude towards using the software.

Usability Testing

Why Usability testing?

Uncover Problems
in the design

Discover Opportunities
to improve the design

Learn about Users
behavior and preferences

Usability Testing

Qualitative usability testing

- focuses on collecting insights, findings, and anecdotes about how people use the product or service. Qualitative usability testing is best for discovering problems in the user experience.
- This form of usability testing is more common than quantitative usability testing.

Quantitative usability testing

- focuses on collecting metrics that describe the user experience.
- Two of the metrics most commonly collected in quantitative usability testing are **task success** and **time on task**.
- Quantitative usability testing is best for collecting benchmarks.

Usability Testing

Usability Testing Process

Usability Testing

Usability Testing Process : Step 1- Plan & Prepare

- **Develop a test plan**:
  - For simple testing, prepare a list of questions
  - For more detailed testing, have a script prepared
- Test Plan is important because you can create a framework for your testing process
- It allows you to communicate your goals with the client & align expectations

Usability Testing

Usability Testing Process : Step 1- Plan & Prepare

Create a Task List:

- Create lists of tasks or questions that a typical user should be able to complete in an hour
- Tasks should not be too simple nor too difficult to accomplish e.g.,
    1. Find a concert show you want to see
    2. Purchase tickets online
    3. Find directions to the venue

Usability Testing

Usability Testing Process : Step 2 - Find Participants

- A challenging aspect in usability testing is finding suitable participants
- Important to gather on ongoing user base
- Test outside the team—testing with people who are not associated with your company or your Web site

Usability Testing

Usability Testing Process : Step 3 - Conduct Session

- Introduce yourself, explain the process to the user
- User will be asked to perform a set of predefined tasks (but do not tell them how many or how long each will take)
- Make the user feel comfortable
- Speak only to give a new task and take notes during the process
- Once the usability test session is over, prepare a short summary of the session and the results
- Outline specific problem areas and any unexpected results
- Include any personal observations

Usability Testing

Usability Testing Process : Step 3 - Conduct Session

- **Collect basic data**:
  - Could the user complete the task?
  - Did they need help?
  - Track how much time it took them
  - Note any stumbling blocks (problems/obstacles)
  - Overall observations, commentary
  - Debrief the user, allow user to speak their mind
  - Prepare a post-test survey

Post-Test Survey:

- Prepare a survey online or in paper form for the user to fill out after they have completed the testing process
- Questions should include what the user thought the Website was like: graphics, logic, content, navigation, and their overall satisfaction
- Gather data about overall effectiveness of the site in relation to the goals of each task

Usability Testing

Usability Testing Process : Step 4 - Analyze Results

- Compile and summarize data
- Transfer handwritten notes to computer
- Write your reports while they are fresh in your mind,
- Create a summary after testing is complete, into a table that shows the results of each test, include problem areas, comments and user feedback from the survey
- Identify difficulties and problem areas
- Identify why there was difficulty or the source of any problems (specific factors such as navigation, text, graphics, etc.)
- Identify any specific task-oriented issues

Usability Testing

Usability Testing Process : Step 4 - Analyze Results

- Compile and recommend
  - Gather all your compiled information and translate into recommendations
  - Concentrate on high-level functionality first
  - Then focus on recommendations for improved user experience (what works and what does not work well for users!)
  - Determine the implementation plan
- Write up a formal report