Sri Lanka Institute of Information Technology

SOFTWARE ENGINEERING PROCESS AND QUALITY MANAGEMENT

Lecture 3 – Functional and Non-Functional Testing

Software Testing

Testing Software

Test Outputs

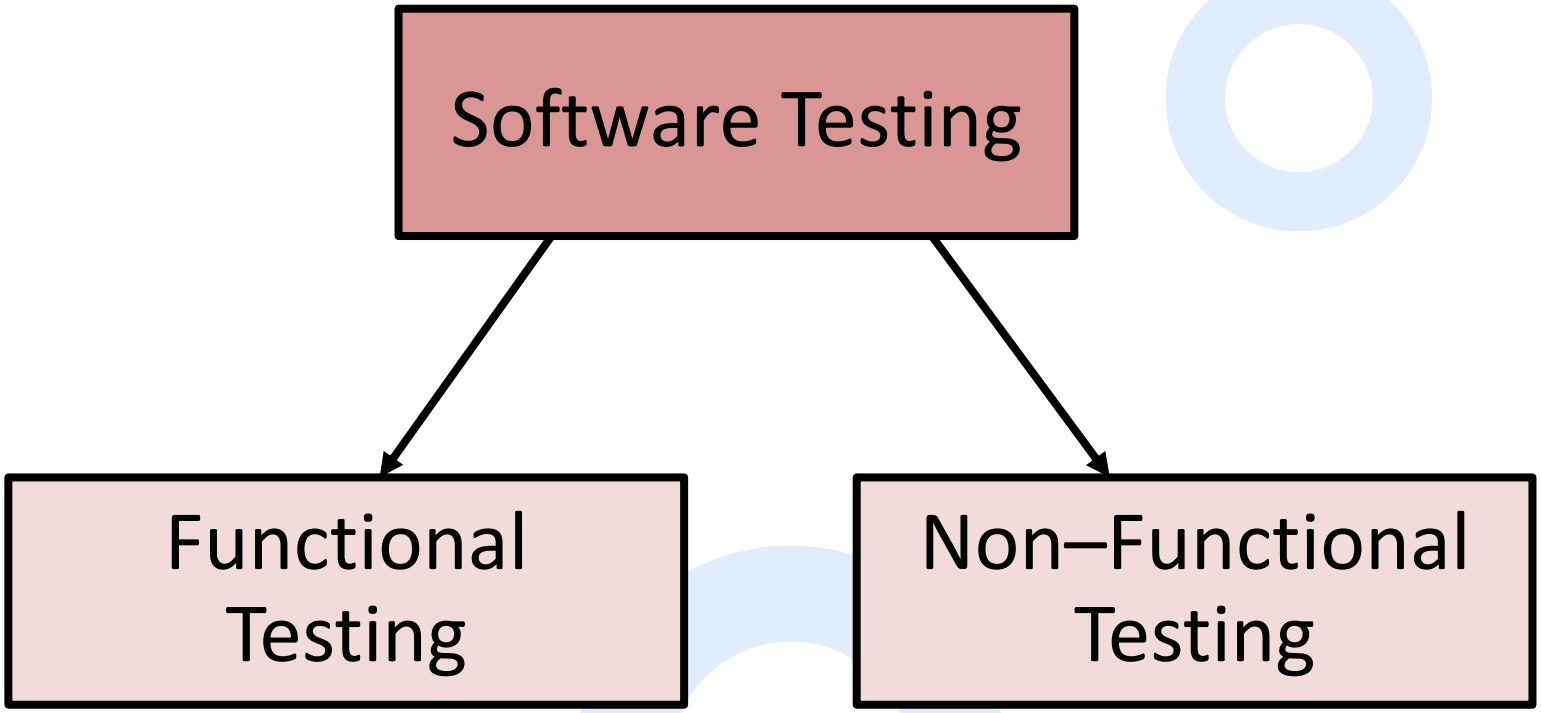
Acceptability Check

Test Inputs

Test Case Pass

Test Case Fail

Software Testing Types



Functional Testing

- A type of testing that verifies that each function of the software application operates in conformance with the requirement specification.
- Each functionality of the system is tested by providing appropriate input, verifying the output and comparing the actual results with the expected results.

Functional Testing

- Unit Testing
- Smoke Testing
- Regression Testing
- Sanity Testing
- System Testing
- User Acceptance Testing

Unit Testing

- A process of testing the individual subprograms, subroutines, classes, or procedures in a program.
- Unit testing is a way of managing the combined elements of testing.
- Unit testing eases the task of debugging.

Unit Testing

Unit Specification

Unit Source Code

Unit Testing

Smoke Testing

- Testing the core functionality of a program.
- The term smoke test in technology is broadly used to test product features in a limited time.

Smoke Testing

- Smoke testing is a subset of all defined test cases that cover the main functionality of a system.
- Examples of capabilities tested during the smoke test;
  - Access to the application
  - Logging in with a set of users
  - Main modules of a particular application

Regression Testing

- Regression testing is testing an existing software application to ensure that a change or addition has not caused any errors with existing functionality.
- Regression testing re-runs the testing scenarios that were originally scripted.
- Regression testing typically requires an automated testing tool.

Selecting Regression Testing

- Requires knowledge about the system and how it is affected by the existing functionalities.
- Select tests based on the area of frequent defects.
- Tests include the areas which have undergone code changes several times.
- Tests are selected based on the criticality of the features.

Sanity Testing

- Performed after receiving a software build, with minor changes in code or functionality.
- Checks whether the planned functionality is working as expected.
- Sanity testing is typically non-scripted.
- Sanity testing is a subset of regression testing.

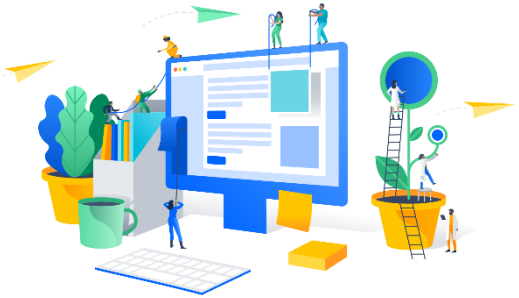
System Testing

- System testing compares the entire system or program to its original objectives.
- Attempting to demonstrate how the entire system fails to meet its objectives.
- Requires a set of measurable objectives for the product.

User Acceptance Testing

- Process of comparing the program to its initial requirements and the current needs of the end users.
- Usually is performed by the customer or end user.
- Developer will conduct user tests during the development cycle prior to delivering the finished product.

Example



TechSolutions

John Steve

Start Truck Cafe

Example

- Steve is the owner of Star Truck Café. He is a successful businessman with many workers working under him. He has faced several difficulties in calculating the salaries of his employees.
- John is working in a company named as TechSolutions. This company develops software products for business requirements.
- Steve wanted an automated system to calculate the salaries of his employees at Star Truck Café. Therefore, he contacted John and signed a contract with TechSolutions to get an automated salary management system for his café.

Example - Specification

- Employees of Star Truck Café required to work from 8.00a.m to 6.00p.m each day.
- Salary management system requires an attendance management system to record in time and out of time of the employees using a fingerprint machine. This sub-system should generate a report of the total hours worked by an employee within a month.
- Employees can work overtime as well. If an employee works after 6.00p.m then that employee will have an overtime payment added to the monthly salary.

Example – Specification (Cont.)

- Overtime payment calculation is another sub system in salary management system. This sub system should calculate the overtime hours worked by an employee using the attendance management system and calculate the overtime payment of an employee.
- Salary calculation unit is the other sub system. In this sub system monthly salaries of all employees will be calculated by considering the attendance and overtime payments of the employee.
- As the owner of Star Truck Café, Steve should be able to login to the system and see all sub systems. All other users can login to the system and see his or her attendance details and salary details.

Example – Development of the System

Andrew

Oliver

Overtime Calculation

Attendance Management

Start Truck Café
Salary Management System

User Management

Salary Calculation

Jessica

Lilia

Unit Testing

Unit Specification

Unit Source Code

Unit Testing

Attendance Management – Unit Testing

- Attendance Management System

  Record in an out time of an employee using the fingerprint machine and generate a report of total hours worked by an employee within a month.

Oliver

Overtime Calculation – Unit Testing

- Overtime Calculation System

  Identify whether each employee has worked overtime and calculate the overtime payment that has to be added to each employee.

Andrew

User Management – Unit Testing

- User Management System

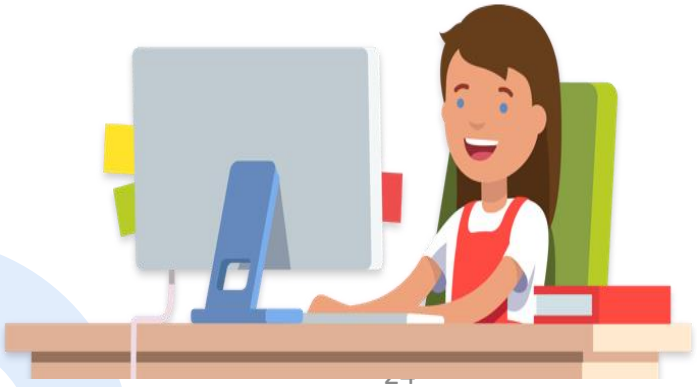
  When a user is logged in as the owner, that user can view all sub units of the system. When a user is logged in as a worker, that user can view his or her salary details and attendance details.

Jessica

Salary Calculation – Unit Testing

- Salary Calculation System

  Calculate the total monthly salaries of all employees by considering the attendance and overtime payments of each employee.



Lilia

Smoke Testing

Smoke Testing	
	Handling user management
	Generating monthly salary for each employee

System Testing

Start Truck Café
Salary Management System

Start Truck Café
Entire System Specification

Change Request

- Add the following feature in calculating overtime payment of an employee.
  - If an employee has worked after 9.00.p.m, double the overtime payment rate for hours after 9.00.p.m

Regression Testing

Regression Testing	
	Attendance management
	Overtime calculation
	User management
	Salary Calculation

28

Minor Change Request

- Change the overtime calculation as follows.
  - If an employee has worked after **10.00.p.m**, double the overtime payment rate for hours after **10.00.p.m**

29

Sanity Testing

Sanity Testing	
	Calculating overtime payment with double overtime value only when the overtime is after 10.00. p.m.

30

User Acceptance Testing

Start Truck Café
Salary Management System

31

Non-Functional Testing

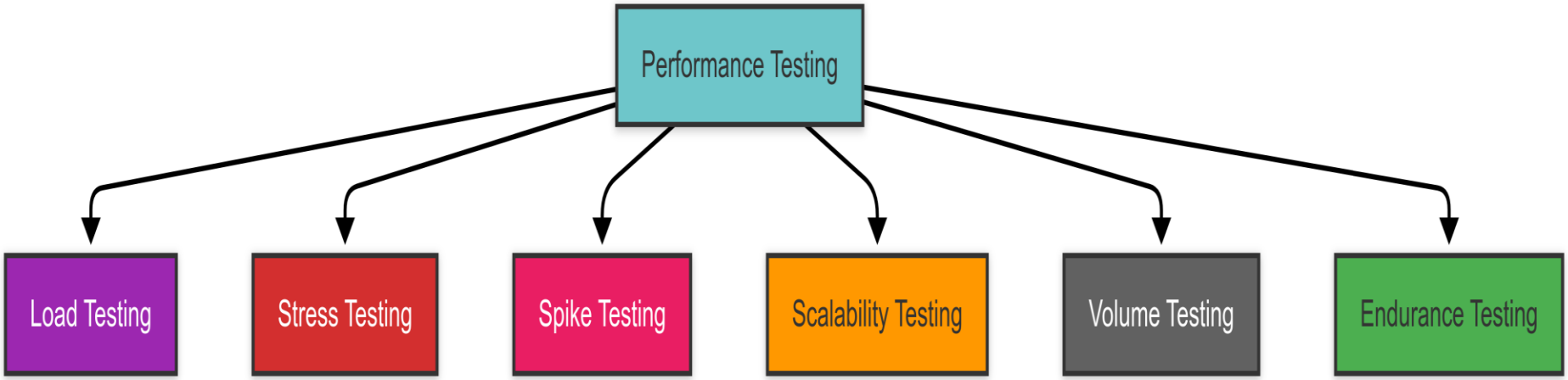
- A type of testing to check non-functional aspects of a software application.
- Examples for non- functional aspects of a software;
  - Performance
  - Usability
  - Reliability
- Explicitly designed to test the readiness of a system as per nonfunctional parameters which are never addressed by functional testing.

Performance Testing

- Designed to test whether the program satisfies its performance objectives.
- Measure the performance of each component to identify which components cause the system to perform poorly.
- Performance testing involves quantitative tests done in a laboratory environment.
- Can compare the performance of two systems.

33

Types Performance Testing



Load Testing

- Test a system with constantly increasing load until “the time to load” reaches to its threshold value.
- Use to distinguish the performance between two different systems.
- Monitor the response time and staying power of an application, when it is performing under a heavy load.

35

Load Testing - Examples

- Testing a printer by sending a large job.
- Editing a very large document in a word processor.
- Continuously reading and writing data into hard disk.
- Running multiple applications simultaneously on the server.
- Testing of a mail server by accessing thousands of mailboxes.

Stress Testing

- Validate an application's behavior when it is pushed beyond normal or peak load conditions.
- Checks the stability of software when the hardware resources are insufficient.
- Determine failures of a system and identify how the system recovers from the failures. This quality is known as the recoverability.

37

Spike Testing

- Performed by suddenly increasing the number of users by a very large amount
- The main aim is to determine whether the system will be able to sustain the sudden heavy workload.

38

Endurance Testing

- Testing a system with an expected amount of load over a long period of time.
- Test cases are executed to check the behavior of a system.
- Consider factors such as memory leaks or system fails or random behavior.

39

Scalability Testing

- Testing of a software application to determine the capability of scaling up in terms of any non-functional requirement.
- Determines the peak of a system, when it has reached a level which prevents from more scaling.

40

Volume Testing

- Testing a software application with a large amount of data.
- Monitor the performance of the application under varying database volumes.

41

Uses of Performance Testing

- Improve user experience.
- Gather metrics useful for tuning the system.
- Identify bottlenecks such as database configuration.
- Determine if a new release is ready for production.
- Provide reporting to business stakeholders regarding performance against expectations.

42

Top Performance Testing Tools

- LoadRunner
- Apache JMeter
- NeoLoad
- Rational Performance Tester
- Loadster
- QEngine (ManageEngine)
- Testing Anywhere
- Loadstorm

43

Compatibility Testing

- Test an application to ensure that it is compatible across operating systems, hardware platforms, web browsers, etc.
- Validation for compatibility requirements that have been set at the planning stage of the software.
- Validates that the application runs properly in versions.

44

Types of Compatibility Testing

- **Backward compatibility Testing**

  Verify the behavior of the developed application with the older versions of the application.

- **Forward compatibility Testing**

  Verify the behavior of the developed application with the newer (upcoming) versions of the application.

45

Security Testing

- Security testing is the process of testing an application to check whether it is according to the specific security objectives.
- Security testing test cases can be derived by studying known security problems in similar systems.
- Web-based applications often need a higher level of security testing than most applications.

46

Usability Testing

- Test the user-friendliness of an application and identify usability defects.
- Testing is performed by using a small set of target end-users.
- Mainly focuses on the user's ease to use the application, flexibility in handling controls and the ability of the system to meet its objectives.

Localization Testing

- Test whether the software behaves according to the local culture or settings.
- Test whether the application has appropriate linguistic and cultural aspects for a particular locality.
- Localization testing of a globalized application is the process of identifying whether all components of the application are designed according to the local culture of target countries and regions.

Non-Functional Testing

Performance Testing

Localization Testing

Compatibility Testing

Security Testing

Usability Testing

49

Activity

- Create a mind map showcasing what you've learned about Software Testing in this lecture.
- Use any online tool of your choice (e.g., MindMeister, Lucidchart, Canva, Miro, XMind).
- Ensure it is detailed, creative, and informative, covering both Functional and Non-Functional Testing along with their subtypes.
- Once completed, upload your mind map to CourseWeb for submission.

50

THANK YOU!