ELYSIUM ACADEMY
Milestone of Cognizance

SOFTWARE TESTING

ELYSIUM ACADEMY SPARK NOTES

VERSION 2.8

ELYSIUM ACADEMY
Milestone of Cognizance

01. Basics of Software Testing

- **Software Testing-** The process of evaluating a software application to ensure it meets the required quality standards
- **Objective-**

  o Verification: Ensuring the software meets its specified requirements.
  o Validation: Ensuring the software meets the user's needs and expectations.
- **Levels of Testing -**

  o Unit Testing: Testing individual components or modules.
  o Integration Testing: Testing the interaction between integrated components or systems.
  o System Testing: Testing the complete and integrated software system
  o Acceptance Testing: Testing the software against requirements.

02. Types of Software Testing

- **Manual Testing-**Test cases are executed manually without any automation tools.
- **Automated Testing -**Test cases are executed using automated tools.
- **Functional Testing -**

  o Unit Testing: Focuses on individual units of source code.
  o Integration Testing: Verifies the interaction between integrated components.
  o System Testing: Validates the complete and integrated software.
  o User Acceptance Testing (UAT): Ensures the software can handle required tasks in real-world scenarios, according to specifications.
- **Non-Functional Testing -**

  o Performance Testing: Determines the speed, responsiveness, and stability under a particular workload.
  o Load Testing: Tests the software's behavior under expected load conditions.

o Stress Testing: Evaluates the software's behavior under extreme conditions.
o Security Testing: Identifies vulnerabilities, threats, and risks in a software application.
o Usability Testing: Assesses how easy and user-friendly the software is.
o Compatibility Testing: Checks how the software performs across different devices, operating systems, and browsers.

- **Other Testing Types -**

o Regression Testing: Ensures that new code changes don't negatively impact existing functionalities.
o Smoke Testing: A quick test to check the basic functionality of the software.
o Sanity Testing: A subset of regression testing to verify specific functionality after changes.
o Exploratory Testing: Simultaneously learning, test design, and test execution.
o Ad-Hoc Testing: Unplanned testing with no specific approach

03. Testing Techniques

- **Black-Box Testing-**Testing without knowledge of the internal code structure

o Equivalence Partitioning: Dividing input data into equivalent partitions for testing.
o Boundary Value Analysis: Testing at the boundaries between partitions.
o Decision Table Testing: Using a table to represent combinations of inputs and outputs.

- **White-Box Testing-**Testing with knowledge of the internal code structure.

o Statement Coverage: Ensures every statement in the code is executed at least once.
o Branch Coverage: Ensures every branch (decision) in the code is executed.
o Path Coverage: Ensures every possible path in the code is executed.

- **Other Testing Types -** Combines both Black-Box and White-Box testing methodologies.

04. Test Automation Tools

- **Selenium-** A tool for automating web browsers.
- **JUnit-** Framework for unit testing in Java
- **TestNG-** Testing framework inspired by JUnit, used for integration and unit testing in Java.
- **PyTest-** Framework for testing Python applications.
- **Jest-** JavaScript testing framework maintained by Facebook, often used with React.
- **Cypress-** End-to-end testing framework for web applications
- **Appium-** Tool for automating mobile applications.
- **LoadRunner-** Performance testing tool for examining system behavior under load.
- **JMeter-** Open-source tool for performance and load testing.

05. Testing Life Cycle

- **Requirement Analysis-**Understanding what needs to be tested

  Test Planning-Defining the objectives and approach of testing

  o Test Plan: A document describing the scope, approach, resources, and schedule for testing activities.
- **Test Case Development-**Creating test cases and test scripts.

  o Test Case: A set of conditions under which a tester determines if a feature is working as expected.
- **Test Environment Setup-**Setting up the necessary hardware and software environment for testing.
- **Test Reporting-**Documenting the test results and any defects found.
- **Test Execution-**Running the tests based on the planned test cases.
- **Test Closure-**Finalizing and closing the testing process, often including post-test analysis.

06. Defect Life Cycle

- **New-** When a defect is found and reported.
- **Assigned-** The defect is assigned to a developer to fix.
- **Open-** The developer begins working on the defect.
- **Retest-** The tester retests the fixed defect.
- **Closed-** The defect is verified and closed if the fix works as expected.
- **Reopened-** If the defect still exists after retesting, it is reopened.

07. Test Documentation

- **Test Plan-**A document outlining the strategy, scope, and resources for testing.
- **Test Case-**Detailed documentation of testing actions.
- **Test Script-**Automated test steps.
- **Test Summary Report-**A report summarizing the testing activities and results.

08. Best Practices in Software Testing

- **Start Early-**Begin testing activities early in the software development lifecycle.
- **Automate When Possible-**Automate repetitive tasks to save time and reduce human error.
- **Prioritize Tests-**Focus on testing the most critical parts of the application first.
- **Regularly Review Test Cases-**Ensure test cases are up-to-date and relevant.
- **Perform Regression Testing-**Continuously test the application after changes to catch any issues.
- **Use Metrics-**Track testing metrics like defect density, test coverage, and test execution rate to monitor quality.

- **Collaborate-** Work closely with developers, business analysts, and other stakeholders to ensure a comprehensive testing process.

09. Test Metrics

Test Coverage-The percentage of the code or application tested.

Defect Density-The number of defects per size of the software module (e.g., per 1,000 lines of code).

Test Execution Rate-The number of test cases executed per unit of time.

- **Defect Removal Efficiency (DRE)-**The percentage of defects identified and removed in the testing phase.

10. Agile and Testing

- **Continuous Testing-**Continuous feedback loop to assess product quality at every stage of the software development lifecycle.
- **Scrum Testing-**Testing activities within a Scrum framework, often with a focus on collaboration and incremental delivery.
- **Test-Driven Development (TDD)-**Writing tests before writing the corresponding code.

This spark notes provides a comprehensive overview of software testing, covering the basics, types of testing, tools, techniques, life cycles, and best practices. It serves as a quick reference for testers and developers involved in the software testing process.

ELYSIUM ACADEMY
Milestone of Cognizance

Thank you

For Your Learning Today

elysiumacademy.org | info@elysiumacademy.org

Scan Here for More
Spark Notes