

## NIRMA UNIVERSITY

<b>Institute:</b>	<b>Institute of Technology, School of Technology</b>
<b>Name of Programme:</b>	<b>BTech CSE</b>
<b>Course Code:</b>	<b>3CS510IC24</b>
<b>Course Title:</b>	<b>Software Testing and Quality Assurance</b>
<b>Course Type:</b>	<b>Department Elective-III</b>
<b>Year of Introduction:</b>	<b>2025-26</b>

L	T	Practical Component				C
		LPW	PW	W	S	
3	0	2	-	-	-	4

### Course Learning Outcomes (CLO):

At the end of the course, the students will be able to –

1. identify different levels and types of testing techniques (BL3)
2. analyse quality assurance practices and principles throughout the software development lifecycle (BL4)
3. determine modern software testing strategies in relation to software development (BL5)
4. design project test plans, test cases, and test data to conduct test operations. (BL6)

Unit	Contents	Teaching Hours (Total 45)
Unit-I	<b>Introduction to Testing and Tools:</b> Overview of software testing, software quality, the role of testing, testing approaches, verification and validation, failure, error, fault and defect, white box and black box testing, test planning and design, monitoring and measuring test execution, test tools and automation, test team organization, and management	03
Unit-II	<b>Unit testing:</b> Concept of unit testing, defect prevention, mutation testing, debugging, control flow testing, control flow graph, paths in a control flow graph, all-path coverage criterion, statement coverage criterion, branch coverage criterion, examples of test data selection, data flow testing, data flow anomaly, data flow graph, data flow testing criteria, feasible paths and test selection criteria, comparison of testing techniques	12
Unit-III	<b>Integration testing:</b> Concept of integration testing, different types of interfaces and interface errors, granularity of system integration testing, system integration techniques, test plan for system integration, functional testing concepts, equivalence class partitioning, boundary value analysis, decision tables, random testing, error guessing	12
Unit-IV	<b>System Testing:</b> System test categories, basic tests, functionality tests, robustness tests, interoperability tests, performance tests, scalability tests, stress tests, load and stability tests, reliability tests, regression tests, documentation tests <b>System test design:</b> Test design factors, requirement identification, characteristics of testable requirements, test objective identification,	10

modeling a test design process, test design preparedness metrics, test case design effectiveness

System test planning and automation: Structure of a system test plan, test approach, test environment, test execution strategy, test effort estimation, scheduling and test milestones, system test automation

System test execution: Metrics for tracking system tests, defect causal analysis, types of acceptance testing

Unit-V **Software quality:** Five views of software quality, McCall's software quality factors, quality criteria, the relationship between quality factors and criteria, components of software quality assurance, software quality, standards, and their requirements, software quality metrics, software reliability models, capability maturity model 08

### Self-Study:

The self-study contents will be declared at the commencement of the semester. Around 10% of the questions will be asked from self-study content.

### Suggested Readings/ References:

1. William Perry, *Effective Methods for Software Testing*, Wiley
2. Sagar Naik, Piyu Tripathy, *Software Testing and Quality Assurance: Theory and Practice*, Wiley
3. Paul C. Jorgensen, *Software Testing - A Craftsman's Approach*, CRC Press.
4. Srinivasan Desikan and Gopalaswamy Ramesh, *Software Testing*, Pearson Education.
5. Louis Tamres, *Introducing to Software Testing*, Addison Wesley
6. Ron Patton, *SAMS Techmedia Indian Edition, Software Testing*, Pearson Education
7. Glenford J. Myers, *The Art of Software Testing*, John Wiley & Sons
8. Robert V. Binder, *Testing Object-Oriented Systems: Models Patterns and Tools*, Addison Wesley
9. Daniel Galin, *Software Quality Assurance*, Pearson Education.

### Suggested List of Experiments:

Sr. No.	Name of Experiments/Exercises	Hours
1	To create test cases based on given requirements for a sample application (e.g., e-commerce app, web page, or mobile app).	02
2	To execute manual test cases and log defects using a bug-tracking tool (e.g., JIRA).	04
3	To study and perform sample tests using the Test Link testing tool.	04
4	To study and perform sample tests using the JUnit testing tool.	04
5	To create automated test scripts using a tool/framework (e.g., Selenium WebDriver).	04
6	To perform manual mobile application testing, focusing on UI/UX, functionality, and compatibility and automated testing of a mobile app using Appium or a similar tool.	02
7	To conduct load testing on a web application using JMeter or a similar tool.	02
8	To perform testing SOAP Web Services for functionality and fault checking.	02
9	To implement a simple continuous testing pipeline using a tool like Jenkins.	02
10	To identify and document vulnerabilities in a web application and simulate a basic penetration testing scenario to identify security flaws.	04