

MALLAREDDY UNIVERSITY**IV Year B. Tech – I Semester****L/T/P/C****3/0/0/3****(MR22-1CS0164) SOFTWARE QUALITY AND TESTING****Course Objectives:**

1. To understand the principles and importance of software quality assurance within the software development life cycle (SDLC).
2. To learn and apply industry-standard software quality models and distinguish SQA from QC and testing.
3. To develop effective test plans, scenarios, and cases using manual testing techniques and IEEE standards.
4. To gain hands-on experience in defect tracking, life cycle management, and end-to-end testing practices.
5. To implement automation testing using Selenium with Java, TestNG, Maven, and Git through practical projects.

Unit 1: Software Quality Assurance

Software Quality: Definition, Importance in SDLC, Cost of Quality (COQ): Prevention, Appraisal, and Failure Costs, Quality vs Reliability vs Security vs Maintainability.

Software Quality Models: McCall's Quality Factors, Boehm's Quality Model, CMMI, SQA Team Roles and Skills, SQA vs QC vs Testing

UNIT 2: Software Testing

Components of a Test Plan (IEEE 829 Standard), Creating a Test Plan, Test Scenarios and Test Cases, Positive and Negative Test Cases

UNIT 3: Testing Techniques

Boundary Value Analysis (BVA), Equivalence Partitioning (EP), Decision Table, State Transition, Use Case Testing, Detect life cycle and Bug Tracking, Writing test cases using BVA and EP, Logging defects in JIRA, End-to-end test design for sample feature (e.g., E-commerce Cart)

UNIT 4: Automation Testing

Introduction to Automation Testing, Automation Tools and Frameworks, Domain-Based Framework Recommendations

UNIT 5: Selenium with Java

Environment Setup, Selenium WebDriver + TestNG, Maven + Git Integration Framework Implementation Case study: Sample automation project (E-commerce app or demo login page)

Reference Books:

1. Software quality assurance: From theory to implementation (1st ed.) by Galin, D Pearson Education.
2. Software testing techniques Beizer, B. 2nd Edition , Van Nostrand Reinhold.
3. Selenium testing tools cookbook (2nd ed.), Gundecha, Packt Publishing

Course Outcomes

After completing this course, students will be able to:

CO1: Demonstrate key software quality concepts and cost components, and differentiate between quality, reliability, and maintainability.

CO2: Apply quality models like McCall's, Boehm's, and CMMI in evaluating software quality.

CO3: Develop structured test plans and test cases using techniques like BVA, EP, and state transition testing.

CO4: Use tools like JIRA for bug tracking and manage the defect life cycle effectively.

CO5: Develop and execute automation test frameworks using Selenium WebDriver and integrate with version control and build tools.