MCAE314: SOFTWARE QUALITY ASSURANCE

Hours/Week: 3 I.A. Marks: 30
Credits: 3 Exam. Marks: 70

Course Learning Objectives: Students will able to try,

1. To study fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods.
2. To discuss various software testing issues and solutions in software unit test; integration, regression, and system testing.
3. To learn how to planning a test project, design test cases and data, conduct testing operations, manage software problems and defects, generate a testing report.
4. To expose the advanced software testing topics, such as object-oriented software testing methods, and component-based software testing issues, challenges, and solutions.

Course Outcomes: After completing the course, the students will be able to,

CO1: Identify and apply various software metrics, which determines the quality level of software

CO2: Identify and evaluate the quality level of internal and external attributes of the software product

CO3: Compare and Pick out the right reliability model for evaluating the software

CO4: Evaluate the reliability of any given software product

CO5: Design new metrics and reliability models for evaluating the quality level of the software based on the requirement

CO6: applying software testing knowledge and methods to practice-oriented software testing projects.

CO7: understand software test automation problems and solutions.

UNIT-I 9 Hrs.

Software Quality: Basics, Popular Views, Quality Professional Views, Software Quality, Total Quality Management and Summary. Fundamentals of Measurement Theory: Definition, Operational Definition and Measurement, Level of Measurement, Some Basic Measures, Reliability and Validity, Measurement Errors, Criteria for Causality, Summary. Software Quality Metrics Overview: Product Quality Metrics, In-Process Quality Metrics, Metrics for Software Maintenance, Examples for Metrics Programs, Collecting Software Engineering Data.

UNIT-II 9 Hrs.

Applying The Seven Basic Quality Tools In Software Development: Ishikawa’s Seven Basic Tools, Checklist, Pareo Diagram, Histogram, Run Charts, Scatter Diagram, Control Chart, Cause And Effect Diagram. The Rayleigh Model: Reliability Models, The Rayleigh Model Basic Assumptions, Implementation, Reliability And Predictive Validity.

UNIT-III 9 Hrs.

Complexity Metrics and Models: Lines of Code, Halstead’s Software Science, Cyclomatic Complexity Syntactic Metrics, An Example of Module Design Metrics in Practice .Metric And Lessons Learned for Object Oriented Projects: Object Oriented Concepts and Constructs, Design And Complexity Metrics, Productivity Metrics, Quality And Quality Management Metrics, Lessons Learned For object oriented Projects. Availability Metrics: Definition and Measurement of System Availability, Reliability Availability and Defect Rate, Collecting Customer Outage Data for Quality Improvement, In-Process Metrics for Outage and Availability.

UNIT-IV 9 Hrs.

Conducting Software Project Assessment: Audit Ad Assessment, Software Process Maturity Assessment and Software Project Assessment, Software Process Assessment Software Process Improvement: Measuring Process Maturity, Measuring Process Capability, Staged Versus Continuous Debating Religion, Measuring the Value of Process Improvement, Measuring Process Compliance, Using Function Point Metrics to Measure Software Process Improvement: Software Process Improvement Sequences, Process Improvement Economies, Measuring Process Improvement at Activity Levels

REFERENCE BOOKS:

1. Stephen H Khan, Metrics and Models in Software Quality Engineering, Pearson 2nd edition 2013.
2. Norman E-Fentor and Share Lawrence Pflieger, Software Metrics, International Thomson, Computer Press 1997.
3. S.A. Kelkar, Software Quality and Testing Market,. PHI Learing, Pvt, Ltd 2012.