Faculty of Science and Technology
R.T.M Nagpur University, Nagpur
Syllabus for B.Tech. Sixth Semester CT

Elective II: Software Testing and Quality Assurance(Theory)

Total Credits: 03	**Subject Code: BTCT603T-2**
Teaching Scheme : **Lectures: 03 Hours/Week** **Tutorials: 00 Hours/Week** **Practical: 00 Hours/Week**	**Examination Scheme :** **Duration of University Exam : 03 Hrs.** **College Assessment : 30 Marks** **University Assessment:70 Marks**

Course Objectives:

1. To learn the criteria for test cases.
2. To learn the design of test cases.
3. To understand test management and test automation techniques.
4. To apply test metrics and measurements.

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

1.Design test cases suitable for a software development for different domains.
2. Identify suitable tests to be carried out.
3. Prepare test planning based on the document, develop and validate a test plan
4. Document test plans and test cases designed.
5. Use automatic testing tools.

Unit I (08 Hrs)

Overview of software evolution, SDLC, Testing Process, Terminologies in Testing: Error, Fault, Failure, Verification, Validation, Difference between Verification and Validation, Test Cases, Testing Suite, Test Oracles, Impracticality of Testing All data; Impracticality of testing AllPaths

Introduction: Purpose – Productivity and Quality in Software – Testing Vs Debugging – Model for Testing – Bugs – Types of Bugs – Testing and Design Style.

Evolution of Quality Control, concept change, TQM Modern concept, Quality concept in design, Review of design.

Unit II (07 Hrs)

Test case Design Strategies – Using Black Box Approach to Test Case Design – Boundary Value Analysis – Equivalence Class Partitioning – State based testing – Cause-effect graphing -Compatibility testing – user documentation testing – domain testing – Random Testing -Requirements based testing

Using White Box Approach to Test design – Test Adequacy Criteria

– static testing vs. structural testing – code functional testing – Coverage and Control Flow Graphs

– Covering Code Logic – Paths – code complexity testing – Additional White box testing approaches-Evaluating Test Adequacy Criteria.

Unit III (07 Hrs)

The need for Levels of Testing – Unit Test – Unit Test Planning – Designing the Unit Tests – The Test Harness – Running the Unit tests and Recording results – Integration tests – Designing Integration Tests – Integration Test Planning – Scenario testing – Defect bash elimination System Testing – Acceptance testing – Performance testing – Regression Testing – Internationalization testing – Ad-hoc testing – Alpha, Beta Tests – Testing OO systems – Usability and Accessibility testing – Configuration testing -Compatibility testing – Testing the documentation -Website testing.

Unit IV (07 Hrs)

Software Testing Activities: Levels of Testing, Debugging, Testing techniques and theirApplicability, Exploratory TestingAutomated Test Data Generation:Test Data, Approaches to test data generation, test data generation using genetic algorithm, Test Data Generation Tools, Software Testing Tools, and Software test Plan.

Software test automation – skills needed for automation – scope of automation – design and architecture for automation – requirements for a test tool – challenges in automation – Test metrics and measurements – project, progress and productivity metrics.

Unit V (07 Hrs)

Object oriented Testing: Definition, Issues, Class Testing, Object Oriented Integration and System Testing. Testing Web Applications: What is Web testing?, User interface Testing, Usability Testing, Security Testing, Performance Testing, Database testing, Post Deployment Testing.

Linguistic –Metrics – Structural Metric – Path Products and Path Expressions. Syntax Testing – Formats – Test Cases .

Logic Based Testing – Decision Tables – Transition Testing – States, State Graph, State Testing.

Text Books:

1. Srinivasan Desikan and Gopalaswamy Ramesh, Software Testing – Principles and Practices, Pearson Education, 2006.
2. Ron Patton, Software Testing, Second Edition, Sams Publishing, Pearson Education, 2007.
3. Ilene Burnstein, Practical Software Testing, Springer International Edition, 2003.

Reference Books:

1. Edward Kit, Software Testing in the Real World – Improving the Process, Pearson Education, 1995.
2. Boris Beizer, Software Testing Techniques – 2nd Edition, Van Nostrand Reinhold, New York, 1990.
3. Aditya P. Mathur, Foundations of Software Testing _ Fundamental Algorithms and Techniques, Dorling Kindersley (India) Pvt. Ltd., Pearson Education, 2008.