Course Title: ADVANCED SOFTWARE TESTING & QUALITY ASSURANCE

Course Level: PG
Course Code: IT610

Course Objectives:
The objective of this course is to expose the students to the Fundamentals and advance concepts in Software testing and quality management. This course discusses about Software testing approaches, Verification and validation Concepts, Guidelines for Building Tests and Test Plans: Designing and Creating Tests, Testing Measurements and Tools and Quality Assurance and Standards.

Prerequisites: Basic knowledge of numerical mathematics, Designing, Testing and Quality skills.

Course Contents/Syllabus:

<table>
<thead>
<tr>
<th>Module I Introduction</th>
<th>Weightage (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Testing Fundamentals - Software Testing Definition, Importance, objectives, why is it too hard? Errors, faults and failure. Testing process, STLC, QA and QC, Verification and Validation, Inspections and walkthroughs, Test Plan, test cases, drivers, stubs, Validation checks.</td>
<td>15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module II Black box, white box testing</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Black box testing - Definition, Equivalence Class, Boundary Value Analysis, White box testing – Definition, Difference between black box testing and white box testing, Path testing, Cyclomatic complexity, graph metrics, mutation testing, cause-effect graphing</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module III Levels of testing</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Levels of testing- Low level testing- Unit testing and Integration testing. High level testing- System testing, performance testing, stress testing, load testing, volume testing, smoke and sanity testing, Installation testing, usability testing, website testing, security testing, recovery testing, Domain testing, Static testing and dynamic testing,</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module IV Regression Testing and automated test data generation</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test cases– Designing, Execution. Reducing number of test cases- Prioritization guidelines, priority category, scheme, risk analysis, regression testing, test case prioritization, automated test data generation</td>
<td></td>
</tr>
</tbody>
</table>
Module V Quality assurance & TMM


The testing maturity model and the test process assessment: the need for a testing maturity model, approach to model development, process improvement model representation, The TMM Structure, relation of the TMM to other process improvement models.

Student Learning Outcomes:

1. Students will demonstrate the ability to manage a project including planning, scheduling and risk assessment/management
2. Student learn systematic approach to the development, operation, maintenance, and retirement of software
3. Student learn how to use available resources to develop software, reduce cost of software and how to maintain quality of software
4. Methods and tools of testing and maintenance of software
5. Have an ability understand and identify various software testing problems, and solve these problems by designing and selecting software test models, criteria, strategies, and methods.
6. Develop methods and procedures for software development that can scale up for large systems and that can be used to consistently produce high-quality software at low cost and with a small cycle time

Pedagogy for Course Delivery:

Subject will be taught on the basis of lectures, practical implementation of concepts learn in the classroom using various real life situations and discussing case study covering different module with special reference to enterprise applications

Lab/Practical’s details, if applicable:

List of Experiments:

1. Design test cases using Boundary value analysis by taking quadratic equation problem
2. Design test cases using Robust analysis by taking quadratic equation problem
3. Design test cases using Equivalence class partitioning by taking triangle problem
4. Design test cases using Decision table by taking triangle problem
5. Design independent paths by calculating cyclometric complexity using date problem
6. Design independent paths by taking DD path using date problem
7. Design the test cases for login page of AMIZONE
8. Design the test cases to enter integer in an array and to check the subscript of array entered
9. Overview of Testing process using Rational Robot
10. Write a script to record verification point using Rational Robot (For GUI testing of single click on window OS)
11. Write programs in „C? Language to demonstrate the working of the following a. constructs: i) do...while ii) while….do iii) if…else iv) switch v) for
12. Overview of Win Runner
13. A program written in C language for Matrix Multiplication fails. Introspect the causes for its failure and write down the possible reasons for its failure.
14. Take any system (e.g. ATM system) and study its system specifications and report the various bugs.
15. Write the test cases for any known application (e.g. Banking application)
16. Create a test plan document for any application (e.g. Library Management System)

**Open Ended Experiments**

1. Design test cases using Boundary value analysis by taking passing /promoting criteria of AUUP as per student handbook of AUUP
2. Write a script to record verification point using Rational Robot (For opening a windows application)
3. Write a script to record verification point using Rational Robot (For opening and working on Social Networking site)

**Assessment/ Examination Scheme:**

<table>
<thead>
<tr>
<th></th>
<th>Theory L/T (%)</th>
<th>Lab/Practical/Studio (%)</th>
<th>Total (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>60%</td>
<td>40%</td>
<td>100%</td>
</tr>
</tbody>
</table>

**Theory Assessment (L&T):**

<table>
<thead>
<tr>
<th>Components (Drop down)</th>
<th>Mid Term Exam</th>
<th>Home Assignment</th>
<th>Presentation/Viva</th>
<th>Attendance</th>
<th>End Term Examination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weightage (%)</td>
<td>10%</td>
<td>8%</td>
<td>7%</td>
<td>5%</td>
<td>70%</td>
</tr>
</tbody>
</table>

**Lab/ Practical/ Studio Assessment:**

<table>
<thead>
<tr>
<th>Components (Drop down)</th>
<th>Lab Performance</th>
<th>Lab File</th>
<th>Viva</th>
<th>Attendance</th>
<th>End Term Examination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weightage (%)</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>70</td>
</tr>
</tbody>
</table>

**Text Reading:**

- Software Testing, Srinivasan Desikan, Pearson Education
References:

- Foundations of software Testing, ISTQB Certification, Dorothy Graham
- Software Test Engineer's Handbook, Graham Bahms