

<b>ITE2004</b>	<b>Software Testing</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>J</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>4</b>	<b>4</b>
<b>Pre-requisite</b>	<b>ITE1005</b>	<b>Syllabus version</b>				
		1.0				
<b>Course Objectives:</b>						
<ul style="list-style-type: none"> <li>To familiarize the testing concepts and evolution</li> <li>To learn the testing strategies and their usage</li> <li>To understand the features and guidelines of testing</li> </ul>						
<b>Expected Course Outcome:</b>						
1) Demonstrate the knowledge of fundamentals of software testing.						
2) Test adequacy assessment using: control flow, data flow, and program mutations.						
3) Apply a wide variety of testing techniques in an effective and efficient manner.						
4) Communicate clearly and effectively use the technical language of the field correctly.						
5) Evaluate the limitations of a given testing process and provide a summary of those limitations.						
6) Check and confirm the Quality standards.						
7) Prepare software quality plan for a software project - focussing on change management, configuration management, defect elimination, validation and verification and measurement.						
8) Have basic understanding and knowledge of contemporary issues in software testing, such as component-based software testing problems.						
<b>Student Learning Outcomes (SLO):</b> 2, 7						
[2] Having a clear understanding of the subject related concepts and of contemporary issues						
[7] Having computational thinking						
<b>Module:1</b>	<b>Introduction</b>	<b>6 hours</b>				
Basic definitions-software testing principles- Role of tester- testing as a process- Overview of Testing maturity model- Defects -Hypothesis and tests						
<b>Module:2</b>	<b>Black box testing strategies</b>	<b>6 hours</b>				
Black-Box Testing Techniques- Random testing- Equivalent partitioning-Boundary Value Analysis (BVA)- Equivalence Class Testing - State Transition Testing - Cause-Effect Graphing Based Testing - Error Guessing -Black box TMM Maturity goals						
<b>Module:3</b>	<b>White box testing strategies</b>	<b>6 hours</b>				
White-Box Testing Techniques- Test adequacy Criteria – coverage and control flow graphs- Basis Path Testing - Loop Testing - Data Flow Testing - Mutation Testing Evaluating adequacy – white box and TMM levels						

<b>Module:4</b>	<b>Levels of testing- Phase-I</b>	<b>6 hours</b>	
Unit testing – Need- Functions- Plan –Design-Considerations – Test Harness, Integration testing- Goals-Strategies-Design- Plan-System testing			
<b>Module:5</b>	<b>Levels of testing- Phase– II</b>	<b>6 hours</b>	
Function test- Performance test-Stress test-Configuration test- Security test – Recovery test, Regression testing-Alpha - beta - Acceptance test- Special role of Use cases- levels of testing and TMM.			
<b>Module:6</b>	<b>Testing policies and organization</b>	<b>6 hours</b>	
Test planning- Components- Attachments-Locating test items- Test reports- Role of three critical groups-Building a test group- Structure- Technical training- Career paths- Certification- Integrating Testing Activities.			
<b>Module:7</b>	<b>Software quality</b>	<b>6 hours</b>	
Quality concepts- Cost estimation- Quality control- Role of operational Profiles and Usage models-statistical testing -Software Reliability –Measurements- applying reliability models- Confidence level-Usability Testing-Software quality control and critical views			
<b>Module:8</b>	<b>Contemporary issues:</b>	<b>3 hours</b>	
	<b>Total Lecture hours:</b>	<b>45 hours</b>	
<b>Text Book(s)</b>			
1.	Ilene Burnstein, Practical Software Testing, Springer Verlag International Edition, Springer (India) Pvt Ltd, 2012.		
<b>Reference Books</b>			
1.	Naresh Chauhan, Software Testing Principles and Practices, Oxford University Press, 2013.		
Recommended by Board of Studies		05-03-2016	
Approved by Academic Council		No. 40	Date 18-03-2016