Nirma University
Institute of Technology
Computer Science & Engineering Department

B. Tech. CSE VII Sem **Academic Year: 2022-23**

Course Code & Name	:	2CSDE80 - Software Testing and Quality Assurance
Credit Details	:	Lectures-3, Tutorial-0, Praticals-2 Credits-4
Course Coordinator	:	Prof. Anuja Nair
Contact No. & Email	:	079-71652000 (9563) Email: anuja.nair@nirmauni.ac.in
Office	:	New building 5th floor, NF-3 Cabin
Course Coordinator Details	:	Visiting Hours: All days – 1:15 pm to 2:00 pm Odd Saturdays – 9:00 am to 4:00 pm
Other Course Faculty		Ms. Riya Kakkar (Research Scholar) [Lab]
LMS Link	:	https://lms.nirmauni.ac.in/course/view.php?id=4861

1. Introduction to Course

1.1 Importance of the course

The software testing industry is a fast-growing Industry, which constantly deals with the project requirements laid by the clients, its easy, prompt, and appropriate solutions along with proper testing mechanism for the quality software outcome. This requires maturity on the part of software companies, to see the probable solutions (with respect to required time, manpower, required tools etc.) of the queries and projects laid down by the clients. For a software company, the concept of reaching a maturity level requires the implementation of certain defined principles on the software testing and quality assurance, which is one of the important aspect in the software development life cycle (SDLC). In this course, we are aiming to cover all the important aspects of software testing and quality assurance, which includes, different types of testing techniques as per the requirement of the environment.

1.2 Objective of the Course

To learn and develop project documentations and soft skills for effective project presentation, develop practical skills related to software quality assurance and to apply software testing techniques for information systems development

1.3 Pre-requisite:

Student must have problem-solving skills and an aptitude for math. They should also possess good programming capability; an in-depth knowledge of programming languages, like C, C++, Java, etc; an attention to detail; and an ability to handle multiple tasks at once and an idea of how the manual testing takes place.

2. Course Outcomes (CO)*

After successful completion of this course, student will be able to:

- interpret different types of testing techniques in depth
- apply modern software testing strategies in relation to software development
- design project test plans, test cases, test data to conduct test operations
- develop practical skills related to software quality assurance

3. Syllabus

Course Content	Number of hours
Unit I Overview of Software Testing: Software Quality, Role of testing, testing approaches Unit Testing: Concept of Unit Testing, Defect Prevention, Mutation Testing, Debugging, Unit Testing in eXtreme Programming Control Flow Testing: Control Flow Graph, Paths in a Control Flow Graph, All-Path Coverage Criterion, Statement Coverage Criterion, Branch Coverage Criterion, Examples of Test Data Selection Data Flow Testing: Data Flow Anomaly, Data Flow Graph, Data Flow Testing Criteria, Feasible Paths and Test Selection Criteria, Comparison of Testing Techniques.	12
Unit II System Integration Testing: System Integration Techniques, Types of Interfaces and Interface errors, Software and Hardware Integration, Off the-shelf component testing, Built-in Testing System Test Categories: Basic Tests, Functionality Tests, Robustness Tests, Interoperability Tests, Performance Tests, Scalability Tests, Stress Tests, Load and Stability Tests, Reliability Tests, Regression Tests, Documentation Tests Functional Testing: Equivalence Class Partitioning Boundary Value Analysis, Decision Tables, Random Testing, Error guessing, Category Partition	08

Unit III System Test Design and Planning: Test Design Factors, Requirement Identification, Characteristics of Testable Requirements, Test Design Preparedness Metrics, Test Case Design Effectiveness System Test Planning, Automation: Structure of a System Test Plan, Beta Testing, System Test Automation	08
Unit IV System Test Execution: Metrics for tracking System Test, Beta Testing, System Test Report, Product Sustaining, Measuring Test Effectiveness Acceptance Testing: Types of Acceptance Testing, Selection of Acceptance Criteria, Acceptance Test Execution, Acceptance Testing in eXtreme Programming. Use cases for State-of-the-Art tools for carrying out Software Testing	07
Unit V Software Quality Assurance: Five views of Software Quality, McCall's Software Quality Factors, Quality Criteria, Relationship between Quality factors and Criteria, Components of SQA, Software Quality, Standards and their requirements, Software Quality Metrics, Software Reliability Models	10

3.1. Self-Study

The self-study components of the syllabus will be declared at the commencement of the semester. Around 10% of the questions will be asked from self-study content.

Topics/content for self-study are as listed below:

- Agile software testing methodology
- Software Quality Standards

Students are expected to study above mentioned topics on their own. These topics will not be taught in the classroom. Students should refer to books available in the library for the same.

3.2. References

1. Sagar Naik University of Waterloo, Piyu Tripathy, Software Testing and Quality Assurance: Theory and Practice, Wiley.
2. William Perry, Effective Methods for Software Testing, Wiley.
3. Paul C. Jorgensen, Software Testing - A Craftsman's Approach, CRC Press.
4. Srinivasan Desikan and Gopalaswamy Ramesh, Software Testing, Pearson Education.
5. Louis Tamres, Introducing to Software Testing, Addison Wesley Publications.
6. Ron Patton, SAMS Techmedia Indian Edition,Software Testing, Pearson Education.
7. Glenford J. Myers,The Art of Software Testing, John Wiley & Sons.
8. Robert V. Binder, Testing Object-Oriented Systems: Models Patterns and Tools, Addison Wesley.
9. Daniel Galin, Software Quality Assurance, Pearson Education.

4. Laboratory details

Each experiment will be of 10 marks. Evaluation for 100 marks will be done throughout the semester as part of the Continuous Evaluation scheme. The assessment of Laboratory work is as under:

Total Marks	Continuous Evaluation			Semester End Evaluation	
100 marks	No. of Practicals	Max. Marks	Weightage	Max Marks	Weightage
	10	100	75%	25	25%

Laboratory experiments/ exercises should be completed as per the given schedule. It is expected that a student does the same with full understanding of the concept, procedure and application involved.

Laboratory work will be based on above syllabus with the following 10 experiments to be performed.

Sr. No.	**Title**	**Hours**	**Mapped CO's**
1	To analyse given webpages from user interface, functionality, and security perspective, and perform manual testing	02	CO2, 3
2	To perform comparative analysis of software testing tools	02	CO2, 3
3	To study and perform sample tests using Test Link Testing tool	04	CO2, 3
4	To study and perform sample tests using J-Unit Testing tool	04	CO2, 3
5	To study and perform sample tests using Selenium Testing tool	04	CO2, 3
6	To perform automated functional GUI testing using any tool	04	CO2, 3
7	To demonstrate performance-based testing experiment	02	CO2, 3
8	To implement load testing method	02	CO2, 3
9	To demonstrate regression-based testing experiment	04	CO2, 3
10	To demonstrate testing of a mobile application	02	CO2, 3

The faculty will evaluate the student based on the proper planning and appropriate content and diagram representations in document related to the practical (6 marks), uniqueness of content (2 marks), proper formatting of document and timely submission as per given deadline (2 marks).

5. Tutorials/ Term assignments/ Innovative assignments/ Term paper (as applicable) details

SE Term Paper/Research Paper - Students will have to submit a term/research paper in a group of 2 students from within the same division of the class and topic of term paper should be some advanced concept related to software testing and quality assurance. The term paper will be evaluated out of 30 marks.

6. Assessment Policy

6.1 Component wise Continuous Evaluation (CE), Laboratory and Project Work (LPW) & Semester End Examination (SEE) weightage

	COMPONENTS WEIGHTAGE								
Course Details				**Continuous Evaluation (CE)**			**LPW**		
Subject Code	**Subject Name**	**Course Coordinator**	Quiz-1	Quiz-2	Sessional	Assignment (Term Paper)	No. of Practicals	Lab Work	Term End Exam
			WT	WT	WT	WT		WT	WT
2CSDE80	Software Testing and Quality Assurance	Prof. Anuja Nair	0.15	0.15	0.40	0.3	10	0.75	0.25
				0.15+0.15+0.40+0.3 =1.0 → WT: 0.4				**0.75+0.25= 1.0 → WT: 0.2**	

Course Assessment Scheme

Assessment scheme		**CE**			**LPW**		**SEE**
Component weightage		0.4			0.2		0.4
	Quiz-1 15 marks	Quiz-2 15 marks	Sessional 40 marks	Term Paper 30 marks	Continuous Evaluation 75 marks	Viva Voce 25 marks	100 marks

Final Course Evaluation: **0.4 CE + 0.2 LPW + 0.4 SEE = 1**

It is mandatory to clear each component with minimum 40%.

6.2 Assessment Policy for Continuous Evaluation (CE)

Assessment of Continuous Evaluation comprises of three components.

1. Quiz-1 will be conducted as per academic calendar. It will be conducted online and will be of 15 marks each.

2. Quiz-2 will be conducted as per academic calendar. It will be conducted online and will be of 15 marks each.

3. Sessional exam will be conducted as per academic calendar. It will be conducted pen and paper based and will be of 40 marks each.

4. Special Assignment: The students will have to submit a term/research paper in a group of 2 students from within the same division of the class and topic of term/research paper should be some advanced concept related to software testing and quality assurance. The term/research paper will be evaluated out of 30 marks.

6.3 Assessment Policy for Laboratory and Project Work (LPW)

Assessment of Laboratory and Project Work comprises of two components.

1. Continuous assessment for laboratory experiments will be conducted. There will be 10 experiments, each carrying weightage of 10 marks. At the end of the course total marks obtained out of 100 will be converted according to weightage assigned. Assessment of experiment will be carried out based on parameters like completion of lab work file, understanding of the experiment performed, originality, involvement of the student, regularity, discipline etc. during the session.
2. A Viva voce examination for LPW component will be conducted as per academic calendar. It will carry a weightage of 25 marks.

6.4 Assessment Policy for Semester End Examination (SEE)

A written examination of 3-hour duration will be conducted for the course as per academic calendar. It will carry 100 marks and marks obtained out of 100 will be converted as per weightage assigned.

7. Lesson Plan

Sr. No.	Topics	Hours	CLOs	Applications
1	Brief discussion on CLOs, scope and detail about books BASICS: • Software Quality. Role of testing, Verification and validation • White-Box and Black-Box Testing, Test Planning and Design, Monitoring and Measuring Test Execution • Test Tools and Automation, Test Team Organization and Management	[04] 1 1 1 1	CO1, 2, 3	Different tools available for performing automatic and manual testing
2	UNIT TESTING: • Concept of Unit Testing, Static Unit Testing, • Defect Prevention, Mutation Testing, Debugging, • Unit Testing in eXtreme Programming • JUnit frame work for Unit Testing	[04] 1 1 2	CO1, 3	A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits, like: used in Extreme programming, Facilitates change etc.
3	CONTROL FLOW TESTING: • Control Flow Graph, Paths in a Control Flow Graph, • All-Path Coverage Criterion, • Statement Coverage Criterion, Branch Coverage Criterion, □ Examples of Test Data Selection	[04] 1 1 2	CO1, 3 □	□Used to draw CFG, calculation of Cyclomatic complexity
4	DATA FLOW TESTING: • Data Flow Anomaly, Data Flow Graph, • Data Flow Testing Criteria • Feasible Paths and Test Selection Criteria	[04] 1 1 2	CO1, 3	Data Flow Testing is useful in modeling test design in order to find data flow anomalies which may possibly occur during data transformation

	• Comparison of Testing Techniques			at the time of execution of code.
5	SYSTEM INTEGRATION TESTING: • Concept of Integration Testing, Types of Interfaces and Interface Errors • System Integration Techniques • Software and Hardware Integration • Off-the-Shelf Component Testing, Built-in Testing	[04] 1 1 1 1	CO1, 3	□ The purpose of system integration testing is to ensure all parts of these systems successfully coexist and exchange data where necessary.
6	SYSTEM TEST CATEGORIES: • Taxonomy of System Tests, Basic Tests, Functionality Tests • Robustness Tests • Interoperability Tests, Performance Tests, Scalability Tests, Stress Tests, Load and Stability Tests, • Reliability Tests, Regression Tests, Documentation Tests.	[04] 1 1 1 1	CO1, 3	□ Help the tester to select appropriate testing category depending upon application specific nature.
7	FUNCTIONAL TESTING: • Equivalence Class Partitioning, Boundary Value Analysis • Decision Tables, Random Testing, Error Guessing, Category Partition	[04] 2 2	CO1, 2, 3	The goal of this testing is to check whether the system is functionally perfect
8	SYSTEM TEST DESIGN: • Design Factors, Requirement Identification, • Characteristics of Testable Requirements, • Test Design Preparedness Metrics, Test Case Design Effectiveness	[03] 1 1 1	CO1, 3	□ Appropriate design factors selection

9	SYSTEM TEST PLANNING AND AUTOMATION: • Structure of a System Test Plan • System Test Automation • Test Selection guidelines for Automation	[03] 2 1	CO1, 2, 3	□ Selecting the right tool can be a tricky task and automation help in this area.
10	SYSTEM TEST EXECUTION: • Metrics for Tracking System Test • Beta Testing, System Test Report • Product Sustaining, Measuring Test Effectiveness.	[04] 2 1 1	CO1, 3	Finding the effectiveness of the system being developed
11	ACCEPTANCE TESTING: • Types of Acceptance Testing, Selection of Acceptance Criteria • Acceptance Test Execution, Acceptance Testing in eXtreme Programming. • Acceptance Test Plan, Acceptance Test Report	[03] 1 2	CO1, 3	Used in agile software □ development, particularly extreme programming, referring to the functional testing
12	SOFTWARE QUALITY: • Five Views of Software Quality, McCall's Quality Factors and Criteria, Quality Factors, Quality Criteria • Relationship between Quality Factors and Criteria, Quality Metrics, ISO 9126 Quality Characteristics, ISO 9000:2000 Software Quality Standard ISO 9000:2000 Fundamentals, ISO 9001:2000 Requirements	[04] 2 2	CO2	Cost Management, Industry data demonstrate that poor application □ structural quality in core business applications (such as enterprise resource planning (ERP), customer relationship management (CRM) or large transaction processing systems in financial services).
	Total	**45**		

8. Mapping of Session Learning Outcomes (SLO) with Course Outcomes (CO)

*already mentioned in the previous section

9. Teaching-learning methodology

1. Lectures: Point Presentations (PPTs) will be used to conduct the course. However, where required, Video Lectures, Simulations / Animations etc. will be used to enhance the teaching-learning process.
2. Laboratory: Explanation of Experiment to be performed along with co- relation with theory will be given. At the end of each session assessment will be carried out based on parameters like completion of lab work that includes observations, calculations, graphs and conclusions, individuality and involvement of the student, regularity, discipline etc. Students will be quizzed to check their understanding of the experiment/exercise conducted.

10. Active learning techniques

Active learning is a method of learning in which students are actively or experientially involved in the learning process. Following active learning techniques will be adopted for the course.

1. Muddiest topic:

   This technique is aimed at finding out the least understood point/topic in the session. This is then further explained to ensure that it is understood well.

2. The "One Minute Paper":

   The faculty will ask students to take out a blank sheet of paper, pose a question (either specific or open-ended), and give them one (or perhaps two - but not many) minute(s) to respond.

3. Wait Time:

   Rather than choosing the student who will answer the question presented, this variation has the faculty WAITING before calling on someone to answer it. The wait time will generally be short (15 seconds or so) - but it may seem interminable in the classroom. It is important to insist that no one raise his/her hand (or shout out the answer) before you give the OK, in order to discourage the typical scenario in which the five students in the front row all immediately volunteer to answer the question, and everyone else sighs in relief.

11. Course Material

Following course material is uploaded on the LMS link of course: https://lms.nirmauni.ac.in/course/view.php?id=4861

- Lecture Notes
- Books / Reference Books / NPTEL video lectures
- Tutorials, Lab Manuals
- Question bank
- Web-links, Blogs, Video Lectures, Journals
- Demonstration of various software testing tools
- Advanced topics

12. Course Learning Outcome Attainment

Following means will be used to assess attainment of course outcomes.

- Use of formal evaluation components of continuous evaluation, tutorials, laboratory work, semester end examination
- Informal feedback during course conduction

13. Academic Integrity Statement

Students are expected to carry out assigned work under Continuous Evaluation (CE) component and LPW component independently. Copying in any form is not acceptable and will invite strict disciplinary action. Evaluation of corresponding component will be affected proportionately in such cases. Turnitin software will be used to check plagiarism wherever applicable. Academic integrity is expected from students in all components of course assessment.