

Kadi Sarva Vishwavidyalaya, Gandhinagar

MCA Semester III

MCA-36 A : Software Testing & Quality Assurance

Rationale:

The objective of this course is to understand the basic view of software quality and quality factors, to understand the Software Quality Assurance (SQA) architecture and the details of its components and to understand of how the SQA components can be integrated into the project life cycle.

Prerequisite: Software Engineering Basics

Learning Outcome: Students will learn how to perform testing, how to develop test cases and how to assure Quality in the software.

Teaching and Evaluation Scheme: Students are evaluated on the basis of continuous evaluation system both in theory and practical classes based on various parameters like term work, class participation, practical and theory assignments, presentation, class test, Regular Attendance, etc.

Sub Total Credit	Teaching scheme		Examination scheme				
	(per week)		MID	CEC	External		Total Marks
	Th	Pr	Th	Th	Th.	Pr.	
4	3	2	25	25	50	50	150

Course Contents:

UNIT 1: Introduction to Software Testing

[20%]

Testing as an Engineering Activity, Role of Process in Software Quality, Testing as a Process, Basic Definitions, Software Testing Principles, The Tester's Role in a Software Development Organization, Origins of Defects, Defect Classes, The Defect Repository and Test Design, Defect Examples, Developer / Tester Support for Developing a Defect Repository.

UNIT 2: Test Case Design

[20%]

Introduction to Testing Design Strategies, The Smarter Tester, Test Case Design Strategies, Using Black Box Approach to Test Case Design Random Testing, Requirements based testing, positive and negative testing, Boundary Value Analysis, decision tables, Equivalence Class Partitioning state-based testing, cause- effect graphing, error guessing, compatibility testing, user documentation testing, domain 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 –Their Role in White-box Based Test Design, code complexity testing, Evaluating Test Adequacy Criteria.

UNIT 3: Levels of Testing

[20%]

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, types of system

testing, Acceptance testing, performance testing, Regression Testing, Alpha – Beta Tests – testing OO systems – usability and accessibility testing.a

UNIT 4: SQA components in the project life cycle [20%]

Software Quality and its factors, SQA Components, Integrating quality activities in the project life cycle, Reviews, Software testing – strategies and implementation, Assuring the quality of software maintenance components, Assuring the quality of external participants' contributions, CASE tools and their effect on software quality.

Procedures and work instructions, Staff training and certification, Corrective and preventive actions, Documentation control

UNIT 5: Management components of software quality, Organizing for quality assurance [20%]

Software quality metrics, Costs of software quality, Scope of quality management standards, ISO 9001, Certification, Capability Maturity Models – CMM and CMMI, Assessment methodology, Management and its role in software quality assurance, The SQA unit and other actors in the SQA system.

Text Books:

1. Srinivasan Desikan and Gopalaswamy Ramesh, “Software Testing – Principles and Practices”, Pearson education, 2006.
2. Aditya P.Mathur, “Foundations of Software Testing”, Pearson Education, 2008.
3. Daniel Galin, “Software Quality Assurance”, Pearson Publication, 2009

Reference Book:

- Ron Patton, “Software testing” , second edition, Pearson education, 2009.
- Boris Beizer, “Software Testing Techniques”, Second Edition,Dreamtech, 2003
- Elfriede Dustin, “Effective Software Testing”, First Edition, Pearson Education, 2003
- Alan C. Gillies, “Software Quality: Theory and Management”, International Thomson Computer Press, 1997.
- Mordechai Ben-Menachem “Software Quality: Producing Practical Consistent Software”, International Thompson Computer Press, 1997.

Practical List: Practical programs can be performed using online tools available.

1. Learn a test case template and prepare test cases for the given example site.
2. Learn a tool for Bug tracking.
3. Perform Boundary Value Analysis using PyTest
4. Write test cases related to Equivalence partitioning using given example site.
5. Write Decision table based test cases for the given example site.
6. Write state based test cases for the given example site.
7. Perform Compatibility testing and write test cases for the tested site.
8. Perform testing using JUnit.
9. Write test cases for the given Mobile application and Web based application
10. Perform Performance testing and write test cases.
11. Perform Code coverage testing using JCOCO and write test cases.
12. Perform Accessibility testing and write test cases.