Course Code: IT-33
Course Name: Software Testing and Quality Assurance

Credit Scheme			Evaluation Scheme				
Lecture	**Practical**	**Credit**	**Internal**			External	Total
			Written	**Practical**	**Tutorial**		
3 Hrs./Week	-	3	10	10	5	50	75

Course Description:

Course Objectives:

1. To understand the principles of software development emphasizing processes and activities of quality assurance
2. To study fundamental concepts in software testing, including software testing objectives, process, strategies and methods.
3. To understand test design techniques based on functionality and structure of software
4. To understand test planning, monitoring and control process
5. To gain the techniques and skills on how to use software testing tools to support software testing activities

Course Outcomes:

Student will be able to

CO1: Understand the role of software quality assurance in contributing to the efficient
delivery of software solutions. (Understand)

CO2: Demonstrate specific software tests with well-defined objectives and targets. (Apply)

CO3: Apply the software testing techniques in commercial environments. (Apply)

CO4: Construct test strategies and plans for software testing. (Analyze)

CO5: Demonstrate the usage of software testing tools for test effectiveness, efficiency and
coverage (Apply)

Course Structure:

Unit No.	Topics Details	Weightage in %	No of Sessions
1	1. Software Quality Assurance Fundamentals 1.1. Definition of Quality, Quality Assurance, Quality Control, Difference between QA and QC, Software Quality Assurance Challenges 1.2. Software Quality Assurance, SQA Planning & Standards (ISO 9000, Six Sigma) 1.3. SQA Activities	20	6

	1.4. Building Blocks of SQA 1.5. Software Quality factors 1.6. Software Quality Metrics: Process Metrics & Product Metrics 1.7. Software Reliability & Reliability Measurement Factors: ROCOF, MTTF, MTTR, MTBF, POFOD, Availability Extra Reading: ISO/IEC 9126, ISO/IEC 25010:2011, Malcom Balridge		
2	2. Software Testing Fundamentals 2.1. Definition & Objectives of Testing 2.2. Role of testing and its effect on quality 2.3. Causes of software failure: Definition of -Error, Bug, Fault, Defect and Failure, 2.4. Economics of Testing 2.5. Seven Testing Principles 2.6. Software Testing Life cycle 2.7. Validation & Verification Concepts - V Model and W Model 2.8. Agile Testing- Test Driven Software Development 2.9. Levels of Testing- 2.9.1. Unit (Component) Testing 2.9.2. Integration Testing 2.9.3. System Testing 2.9.4. User Acceptance Testing (UAT) 2.10. Test Types 2.10.1. Functional testing (Black-box) 2.10.2. Non-functional testing (Testing of software product characteristics) 2.10.3. Structural testing (White-box) 2.10.4. Testing related to changes - Confirmation (Re-testing) and Regression Testing 2.11. Non-Functional Testing Types – 2.11.1. Performance (Load & Stress) 2.11.2. Usability 2.11.3. Maintainability 2.11.4. Portability 2.11.5. Security 2.11.6. Localization & Internationalization 2.12. Concept of Smoke testing and Sanity Testing	17	10
3	3. Static Testing 3.1. Static Techniques – Review 3.1.1. Review Process (Informal & Formal) 3.1.2. Desk Checking, 3.1.3. Technical or Peer Review 3.1.4. Walkthrough	8	3

	3.1.5. Inspection 3.2. Static Techniques – Static Analysis 3.2.1. Data flow analysis 3.2.2. Control flow analysis, 3.2.3. Static Analysis by Tools (Automated Static Analysis) **Case Study on Preparation of Inspection Checklist**		
4	4. Dynamic Testing 4.1. Test Design Techniques-Black Box Testing Techniques: 4.1.1. Equivalence Partitioning 4.1.2. Boundary Value Analysis 4.1.3. Decision Table Testing 4.1.4. State Transition Testing 4.2. Test Design Techniques -White Box Testing Techniques (coverage based and fault-based) 4.2.1. Statement coverage 4.2.2. Branch & Decision coverage 4.2.3. Path coverage 4.2.4. McCabe's Cyclomatic Complexity Metric (Computation of Cyclomatic Complexity to be covered) 4.2.5. Data Flow based Testing 4.2.6. Mutation Testing 4.3. Test Design Techniques -Experience based techniques: 4.3.1. Error Guessing 4.3.2. Exploratory Testing **Problems based on Black Box and White Box Testing Techniques to be covered**	15	7
5	5. Test Management 5.1. Test Organization- Roles & Skills of Tester, Test Lead, Test Manager 5.2. Test Planning- Test Plan as per IEEE 829 STANDARD TEST PLAN TEMPLATE 5.3. Test Process Monitoring & Control 5.3.1. Test Monitoring through -Test Log (IEEE 829: TEST LOG TEMPLATE to be discussed) and Defect Density 5.3.2. Reporting Test Status (IEEE 829: TEST SUMMARY REPORT TEMPLATE to be discussed) 5.3.3. Test Control 5.4. Requirement Traceability Matrix (Horizontal & Vertical), Test Scenario, Test Suite, Test Cases (both Positive & Negative Test Cases, as per IEEE 829:	25	10

	TEST CASE SPECIFICATION TEMPLATE) 5.5. Configuration Management- Configuration Management support for Testing 5.6. Risk and Testing- Project Risk & Product Risk 5.7. Incident/ Defect Management 5.7.1. Defect Life Cycle 5.7.2. Defect/ Incident Report (IEEE 829: TEST INCIDENT REPORT TEMPLATE to be discussed) **Case Study on Test Plan for applications and Case study on Test Cases for different features within applications** **Extra Reading:** Version Control Tool: SVN, Defect Tracking Tool: Bugzilla, JIRA		
6	6. Tool Support for Testing 6.1. Types of Test tools –CAST (only type & their purpose should be covered) 6.2. Effective Use of Tools: Potential Benefits and Risks 6.3. Introduction of a tool into an organization 6.4. Testing tools 6.4.1. Selenium -WebDriver and Test NG 6.4.2. Appium 6.4.3. JMeter **Extra Reading**: Functional Test Automation Tools: Quick Test Professional (QTP), IBM Rational Robot, Non-functional Test Automation Tools: Load Runner, Test Management Tools: Test Director, Test Link, Bugzilla, Redmine, API Testing Tool: Postman, ETL Testing Tool, Big Data Testing Tool, AI based Testing Tool: Test Craft, UI Testing, Website Testing: TestRail	15	9
	Total:	**100**	**45**

List of Practicals (if any)

Practicals based on Selenium

1. Focusses on how to invoke the Firefox browser, maximizing the window, navigate commands, etc.

Scenario:

1.1. Open the Firefox browser.
1.2. Maximize the browser window.
1.3. Navigate to a particular URL (let say, SPPU website) “http://www.unipune.ac.in/”.

1.4. Write a method to print PASS if the title of the page matches with the page title else FAIL. (If you are familiar with TestNG or JUnit use assert statement like assert.assertequals(actual, expected) to give a verdict of the pass or fail status.

1.5. Navigate to another URL (let say, the Facebook page) “https://www.facebook.com”

1.6. Navigate back to the QA Tech Hub website.

1.7. Print the URL of the current page.

1.8. Navigate forward.

1.9. Reload the page.

1.10. Close the Browser.

2. Focusses on interacting with form elements like textbox, buttons, radio buttons and drop-down (Facebook Signup)

Scenario:

2.1. Open a browser.

2.2. Navigate to “http://www.fb.com”

2.3. Verify that the page is redirected to “http://www.facebook.com”, by getting the current URL. (use if-else condition to verify this condition or use Assert.assertequals() in case you are familiar with TestNG or JUnit)

2.4. Verify that there is a “Create an account” section on the page.

2.5. Fill in the text boxes: First Name, Surname, Mobile Number or email address, “Re-enter mobile number”, new password.

2.6. Update the date of birth in the drop-down.

2.7. Select gender.

2.8. Click on “Create an account”.

2.9. Verify that the account is created successfully.

3. Focusses on working with links and getAttribute() method.

Scenario:

3.1. Open a Browser (write the generic code such that by changing the parameter browser can be changed.)

3.2. Navigate to https://flipkart.com website.

3.3. Write a method to find the count (number of) links on the homepage of Flipkart.

3.4. Write another method to print link text and URLs of all the links on the page of Flipkart.

4. Focusses on Frame Handling, mouse over operations (Login to an application)

Scenario:

4.1. Open any browser of your choice, for example, Chrome Browser.

4.2. Navigate to Snapdeal site (http://www.snapdeal.com)

4.3. Move to Sign in Button and hold

4.4. Move to the Sign In button and click.

4.5. Enter valid Email Id and click continue.

4.6. Enter the valid password and click LOGIN.

4.7. Verify that the user is logged in successfully.

5. Focusses on writing dynamic xpath (Login to Gmail Account and sending a mail from Gmail)

Scenario:

- 5.1. Open any browser of your choice, say Mozilla Firefox
- 5.2. Navigate to https://www.gmail.com
- 5.3. Enter a valid Email Id or Phone Number
- 5.4. Click Next button
- 5.5. Enter Password and click "Sign in" button.
- 5.6. Verify that Gmail is logged in successfully.
- 5.7. Click compose button and verify that a new mail window is opened.
- 5.8. Enter an Email Id
- 5.9. Enter some subject, say "Test Mail"
- 5.10. Enter some text in body
- 5.11. Click send button.

Course References:

Recommended Books:

Text Books:

1. Foundations of Software Testing by Rex black, Erik Van Veenendaal, Dorothy Graham (2012)-Cengage Learning: London UK, 3rd Edition
2. Software Engineering by Sommerville-Pearson,8thEdition
3. Daniel Galin, "Software Quality Assurance: From Theory to Implementation", Pearson Addison-Wesley, 2012. 2.
4. Effective Methods for Software Testing by William Perry- Wiley Pub, 3rd Edition.

Reference Books:

1. Roger S. Pressman, "Software Engineering-A Practitioner's Approach", McGraw Hill pub.2010
2. Software Testing in Real World Edward Kit- Pearson Pub
3. Software Testing Techniques by Boris Beizer-DreamTech Pub,2nd Edition
4. Software Testing by Ron Patton, TechMedia Pub.
5. Introducing Software by Testing Louise Tamres
6. Fundamentals of Software Engineering –Rajib Mall, 3rd Edition
7. Allen Gilles "Software quality: Theory and management", International Thomson, Computer press 1997.
8. Software Testing Principles Techniques and Tools by Milind. G. Limaye- Tata McGraw Hill Pub.
9. Stephen H. Kan, "Metrics and models in software quality Engineering", Addison –Wesley 2003.

Web Reference:

1. www.istqb.org
2. https://www.seleniumhq.org/
3. https://www.softwaretestingmaterial.com/selenium-tutorial/
4. https://www.toolsqa.com/selenium-tutorial/
5. www.guru99.com/software-testing.html
6. www.guru99.com/selenium-tutorial.html
7. www.guru99.com/mobile-testing.html
8. https://www.softwaretestinghelp.com/appium-tutorial-for-beginners/
9. www.professionalqa.com
10. www.resources.sei.cmu.edu/library
11. www.iist.org