Program Content

Semester	VI		
Course Code:	IT6206		
Course Name:	Software Quality Assurance		
Credit Value:	3 (2L+ 1P)		
Core/Optional	Core		
Hourly Breakdown	Theory	Practical	Independent Learning
	30 Hrs	30 Hrs	90 Hrs

Course Aim:
This course is aimed at delivering the fundamentals of software quality assurance and testing and their practical aspects. Students will learn how software testing and quality assurance tasks are performed in different stages of the software development life cycle. Further the students will learn about how continuous integration and continuous delivery mechanisms are being used in software testing.

Intended Learning Outcomes:
After following this course, students should be able to:

- explain fundamental concepts in software quality (e.g., internal / external quality, as well as quality in use)
- describe and explain definitions and activities related to software testing, such as faults, failures levels of testing and test automation
- explain the concept of continuous integration and relate them to software development processes
- write a successful systems test plan
- formulate a traceability matrix
- use selenium, TestNG and Java based tools to automate simple web application testing process
- integrate testing process with Continuous Integration (CI) and Continuous Delivery (CD)
- develop a suitable mobile testing strategy
- use Appium for testing native mobile applications
- use test report generation tools to create test reports

Course Content: (Main Topics, Sub topics)

Topic	Theory (Hrs)	Practical (Hrs.)
1. Introduction to Software Testing	2	-
2. Software Testing Process	2	-
3. Test Techniques and their Characteristics	2	-
4. Test Case Design	3	-
5. Levels of Testing	2	-
6. Software Testing Life cycle	2	-
7. Quality Control	2	-
8. Test Automation	2	-
9. XML Based Test Automation	3	4
10. Automated Testing Suites for Web Applications (e.g Selenium)	2	4
11. Javabased Test Automation (e.g TestNG)	3	8
12. Continuous Integration (CI) and Continuous Delivery (CD)	2	4
13. Mobile Test Automation (e.g Appium)	2	8
14. Test Reporting	1	2
Total	**30**	**30**

1 Introduction to Software Testing (2 hours)

1.1 Introduction to Software Quality Assurance and Testing [Ref 1: pg. 1 - 5]
1.2 Typical Objectives of Testing [Ref 1: pg. 3 - 4]
1.3 Testing and Debugging [Ref 1: pg. 4 - 5]
1.4 Errors, Defects, and Failures [Ref 1: pg. 7 - 9]
1.5 Defects, Root Causes and Effects [Ref 1: pg. 9 - 10]
1.6 Manual vs Automation Testing [Ref :Teacher's Note]
1.7 Seven Testing Principles [Ref 1: pg. 10 - 15]
1.8 Software Quality Assurance Models and Standards [Ref :Teacher's Note]

2 Software Testing Process(2 hours)

2.1 Testing Process [Ref 1: pg. 15 - 27]
2.2 Test Activities and Tasks [Ref 1: pg. 17 - 27]
2.3 Traceability between the Test Basis and Test Work Products[Ref 1: pg. 27]
2.4 Software Development and Software Testing [Ref 1: pg.36 - 46]
2.5 Software Testing Methodologies
    2.5.1 White Box Testing, Black Box Testing, Gray Box Testing [Ref 1: pg. 65, - 68]
    2.5.2 Functional Testing, Non-functional Testing, Change-related Testing [Ref 1: pg. 63 - 67]
    2.5.3 Test Types and Test Levels [Ref 1: pg. 47 - 68]
2.6 Static vs Dynamic Testing [Ref :Teacher's Note]

3 Test Techniques and Their Characteristics (2 hours)

3.1 Black-box Test Techniques [Ref 1: pg. 112 - 132]
    3.1.1 Equivalence Partitioning [Ref 1: pg. 113 - 115]
    3.1.2 Boundary Value Analysis [Ref 1: pg. 115 - 121]

3.1.3 Decision Table Testing [Ref 1: pg. 121 - 127]
3.1.4 State Transition Testing [Ref 1: pg. 127 - 130]
3.1.5 Use Case Testing [Ref 1: pg. 130 - 132]
3.2 White-box Test Techniques [Ref 1: pg. 132 - 140]
3.2.1 Statement Testing and Coverage [Ref 1: pg. 136 - 137]
3.2.2 Decision Testing and Coverage [Ref 1: pg. 138 - 139]
3.2.3 The Value of Statement and Decision Testing [Ref 1: pg. 139 - 140]
3.3 Experience-based Test Techniques [Ref 1: pg. 140 - 142]
3.3.1 Error Guessing [Ref 1: pg. 140 - 141]
3.3.2 Exploratory Testing [Ref 1: pg. 141 - 142]
3.3.3 Checklist-based Testing [Ref 1: pg. 142]

4 Test Case Design (3 hours)

4.1 Static Methods [Ref 1: pg. 75 - 99]
4.2 Benefits of Static Testing [Ref 1: pg. 77 - 78]
4.2.1 Informal Reviews [Ref 1: pg. 88 - 89]
4.2.2 Walkthroughs [Ref 1: pg. 88 - 89]
4.2.3 Technical Reviews [Ref 1: pg. 89 - 90]
4.2.4 Inspection [Ref 1: pg. 90 - 91]
4.3 Dynamic Methods
4.3.1 Black Box Techniques [Ref 1: pg. 112 - 132]
4.3.2 Experience-based Techniques [Ref 1: pg. 140 - 142]

5 Levels of Testing (2 hours)

5.1 Functional Testing
5.1.1 Component/Unit Testing [Ref 1: pg. 48 - 50]
5.1.2 Integration Testing [Ref 1: pg. 50 - 53]
5.1.3 System Testing [Ref 1: pg. 53 - 55]
5.1.4 User Acceptance Testing [Ref 1: pg. 55- 59]
5.2 Non-functional Testing
5.2.1 Interoperability Testing [Ref 3: pg. 208 - 209]
5.2.2 Performance Testing [Ref 3: pg. 209 - 210]
5.2.3 Scalability Testing [Ref 3: pg. 210 - 211]
5.2.4 Stress Testing [Ref 3: pg. 211- 213]
5.2.5 Security Testing [Ref 3: pg. 203 - 204]
5.2.6 Load and Stability Testing [Ref 3: pg. 213 - 214]
5.2.7 Reliability Testing [Ref 3: pg. 214]
5.2.8 Regression Testing [Ref 3: pg. 214 - 215]
5.2.9 Documentation Testing [Ref 3: pg. 215 - 216]

6 Software Testing Life Cycle (2 hours)

6.1 Requirements Analysis/Design[Ref 14, 16]
6.2 Entry Criteria and Exit Criteria [Ref 14]
6.3 Test Planning [Ref 2: pg. 65 - 66][Ref 14, 16]
6.4 Test Case Design and Development [Ref 2: pg. 66] [Ref 14, 16]
6.4 Test Environment Setup [Ref 14, 16]
6.5 Defect Life Cycle [Ref 14]
6.6 Defects Classification [Ref 14]
6.7 Test Execution [Ref 14, 16]
6.8 Test Closure [Ref 14, 16]

7 Quality Control (2 hours)

7.1 Test Monitoring and Control [Ref 2: pg. 110 - 111]
7.2 Metrics Used in Testing [Ref 2: pg. 156 - 169]
7.3 Purposes, Contents, and Audiences for Test Reports [Ref 2: pg. 112 - 114]

8 Test Automation (2 hours)

8.1 Introduction to Test Automation [Ref :Teacher's Note]
8.2 Choosing the Right Tool [Ref :Teacher's Note]
8.3 Effective Use of Tools [Ref :Teacher's Note]
8.4 Test Automation Environments [Ref :Teacher's Note]
8.5 Manual Testing vs Test Automation [Ref :Teacher's Note]
8.6 Benefits and Risks of Test Automation [Ref :Teacher's Note]
8.7 Test Tool Considerations [Ref :Teacher's Note]

9 XML Based Test Automation (e.g XPath) (3 hours)

9.1 Introduction to XML Path in Selenium [Ref 08]
9.2 Absolute XPath [Ref 08]
9.3 Relative XPath [Ref 08]
9.4 Finding Elements using Attributes with XPath [Ref 08]

**** Guided practical 1**

At the end of the section, students are guided to complete the following practical;

- *Visit any web page your choice and open the source view. Capture at least five of the following html elements using absolute and relative xpath definitions.*
- *An anchor tag with a hyperlink*
- *Second list item in an unordered/ ordered list*
- *A button using its caption text*
- *Second last item of an item list*
- *A label using its text*
- *Identify a link using link text*
- *Capture the third table row of a table*

10 Automated Testing Suites for Web Applications (e.g Selenium) (2 hours)

10.1 Introduction to Automated *Testing* Suite for Web Applications [Ref. 5, 15]

10.2 Architectures of Automated *Testing* Suite for Web Applications [Ref. 5, 15]

10.3 Recorders in Automated *Testing* Suite for Web Applications (e.g Selenium IDE, Katalon) [Ref. 5, 12]

10.4 Working with Selenium API [Ref 15]

**** Guided practical 2**

At the end of the section, students are guided to complete the following practical;

Use Katalon recorder/ Selenium IDE to record a test case. Ideally you can use an online calculator (Eg: https://www.calculator.net/). Complete the following steps to record the test case.

- *Open the online calculator site*
- *Click on the appropriate calculator link to open the calculator (Eg: Percentage calculator)*
- *Insert value and the percentage you want to calculate. (IF you choose another calculator, input whatever the relevant values*
- *Click on the button to calculate*
- *Assert the result/value*
- *Stop recording*

Finally,student should have a test case recorded using Katalon recorder/Selenium IDE

11 Javabased Test Automation (e.g TestNG) (3 hours)

11.1 Introduction to Test Automation Frameworks for the Java Programming Language (e.g TestNG) [Ref 6, 17]

11.2 Classes Handling in Test Automation(e.g TestNG) [Ref 6, 17]

11.3 Use of XML in Test Automation(e.g TestNG) [Ref 6, 17]

11.4 Use of Annotations in Test Automation(e.g TestNG) [Ref 6, 17]

**** Guided practical 3**

At the end of the section, students are guided to complete the following practical;

- *Use Eclipse IDE to complete the following practical.*
- *Configure eclipse with TestNG library*
- *Create a new TestNG class*
- *Export previously recorded (Guided Practical 2) test case to TestNG web driver format*
- *Copy the code into TestNG class*
- *Resolve the dependencies (You need to add browser driver (Eg: chrome) to the project build path*
- *At the end you should be able to execute the recorded test case from the TestNG class you created*

12 Continuous Integration (CI) and Continuous Delivery (CD) (2 hours)

12.2 Introduction to CI and CD tools(e.g Jenkins) [Ref. 19]

12.3 Setting up Jenkins [Ref. 20]

12.3 Jenkins Pipeline Process [Ref. 21]

12.4 Introduction to Version Control Systems (e.g Git) [Ref. 7]

13 Mobile Test Automation (e.g Appium) (2 hours)

- 13.1 Introduction to Mobile Applications and Architectures [Ref. 4: pg]
- 13.2 Types of Mobile Applications
  - 13.2.1 Native Applications [Ref. 4: pg 9 - 11]
  - 13.2.2 Browser-based Applications [Ref. 4: pg 11 - 13]
  - 13.2.3 Hybrid Applications [Ref. 4: pg 13 -15]
- 13.3 Test Strategy for Mobile Applications [Ref. 18]
- 13.4 Mobile Application Testing Fundamentals [Ref. 18]
- 13.5 Mobile Testing Tools and Frameworks [Ref. 10, 11, 12, 13]
- 13.6 Appium Architecture [Ref. 4: pg 15 - 18]
- 13.6 Native Type Mobile Application Testing using Appium [Ref. 4: pg 45 - 88]

**** Guided practical 4**

At the end of the section, students are guided to complete the following practical;

Develop a simple Android demo application with login feature
Automate testing simple login interface on Android application.

14 Test Reporting (1 hour)

- 14.1 Test Reporting Process [Ref :Teacher's Note]
- 14.2 Test Reporting Frameworks [Ref :Teacher's Note]
- 14.3 Software Tools for Test Reporting [Ref: 9]

**** Guided practical 5**

At the end of the section, students are guided to complete the following practical;

Refer to the test cases used in guided practical 2, 3 and 4 to generate test reports. Use Cucumber reports engine or Maven surefire plugin to generate the test reports.

Mini Project: A mini project covering wider topics already covered will be given to apply the practical knowledge gained.

Teaching /Learning Methods:

You can access all learning materials and this syllabus in the VLE: http://vle.bit.lk/, if you are a registered student of the BIT degree program.

Assessment Strategy:

Continuous Assessments/Assignments:

In the course, case studies/Lab sheets will be introduced, and students have to participate in the learning activities.

Final Exam:

The final exam of the course will be held at the end of the semester. This course is evaluated using a two-hour question paper consisting of 4 Structured Questions.

References/ Reading Materials:

[1] Foundations of Software Testing ISTQB Certification,Rex Black, Erik van Veenendaal, Dorothy Graham, ISBN 978-1473764798

[2] Concise guide to Software Testing, Gerard O'Regan, Springer Nature Switzerland AG, ISBN 978-3-030-28494-7

[3] Software Testing and Quality Assurance Theory and Practice, KSHIRASAGAR NAIK and PRIYADARSHI TRIPATHY, John Wiley & Sons, Inc., ISBN 978-0-471-78911-6

[4] Mobile Test Automation with Appium - comprehensive guide to build mobile test automation solutions using Appium (2017), Nishant Verma, Packt Publishing Ltd., ISBN 978-1-78728-016-8

Supplementary Materials:

[5] https://www.selenium.dev/documentation/

[6] https://testng.org/doc/documentation-main.html

[7] https://git-scm.com/docs

[8] https://www.guru99.com/xpath-selenium.html

[9] https://testng.org/reportng/

[10] http://appium.io/docs/en/about-appium/intro/

[11] https://support.smartbear.com/testcomplete/docs/index.html

[12] https://docs.katalon.com/katalon-studio/docs/index.html

[13] https://www.ranorex.com/resources/

[14] https://www.tutorialspoint.com/stlc/index.htm

[15] https://artoftesting.com/selenium-tutorial

[16] https://artoftesting.com/difference-between-sdlc-and-stlc

[17] https://artoftesting.com/testng-tutorial

[18] https://developer.android.com/training/testing/fundamentals

[19] https://www.jenkins.io/doc/

[20] https://www.jenkins.io/doc/book/installing/

[21] https://www.jenkins.io/doc/book/pipeline/