SOFTWARE QUALITY ASSURANCE

1. General Information

Course name: Software quality assurance

Course code: INT1416_CLC

Number of credits: 3

2. Objectives

Knowledge:

The aim of this course is to provide learners with background of software quality assurance and applying reviewing and testing techniques during the software development process.

On completion of this course, learners will be able to follow software quality assurance activities for the software development process, from review requirements to review analysis, review design, review source code, and testing.

Skills:

On successful completion of this course a learner will:

- know how to create SQA plan, test plan, test cases.
- execute reviewing documents, testing softwares.
- make use of SQA tools.

Attitude:

Learners are required to attend the classes and complete assignments/projects.

3. Abstracts

The aim of this course is to provide learners with the background of software quality assurance, the process of software quality assurance and testing, the techniques of important activities in software quality assurance, including: reviewing, testing. Besides, the course provides the standards of software quality assurance.

On completion of this course, learners will be able to follow a software quality assurance and testing process for a software project as well as to execute reviewing, testing activity during development of a software in the right way based on the software quality assurance process, from requirement to analysis, design, implementation, and testing.

4. Teaching and learning methods

- Lectures: 36h
- Exercises: 08h
- Projects: 00h
- Lab: 00h
- Individual reading: 01h

5. Prerequisites

6. Learning outcomes

After studying this course, the studier could:

[LO1]: Understand concepts related to software quality assurance, software development process quality standards.

[LO2]: Understand the reviewing, testing technologies in each stage of the software development process.

[LO3]: Apply the learned knowledge to execute reviewing, testing during software development.

7. Assignment criteria /Learning outcomes matrix

Learning outcomes	Assignment criteria
[LO1]: Understand concepts related to software quality assurance, software development process quality standards.	Chapter 1 Chapter 2 Chapter 7
[LO2]: Understand the reviewing, testing technologies in each stage of the software development process.	Chapter 3 Chapter 4 Chapter 5
[LO3]: Apply the learned knowledge to execute reviewing, testing during software development.	Chapter 3 Chapter 4 Chapter 5 Chapter 6

8. Outlines

Chapter 1: Introduction of software quality assurance

1.1. What is software?
1.2. Classification of the causes of software errors
1.3. Software quality assurance – definition and objective
1.4. Software quality factors.
1.5. Factors affecting intensity of quality assurance
1.6. Summarization and Exercise

Chapter 2: Integrating quality activities in the project life cycle

2.1. Software development methodologies
2.2. Testing level
2.2.1. Introduction
2.2.2. Unit test
2.2.3. Integration test
2.2.4. System test
2.2.5. Acceptance test
2.3. A model of SQA defect removal effectiveness and cost.
2.4. Create SQA plan
2.5. Summarization and Exercise

Chapter 3: Reviews

3.1. Review objectives
3.2. Formal review
3.3. Peer reviews
3.4. Implementation of reviews activity in projects
3.4.1. Reviewing requirement specification
3.4.2. Reviewing design document
3.4.3. Reviewing development plan and SQA plan

3.4. Summarization and Exercise
Chapter 4: Black Box testing
4.1. Definition and objectives
4.2. Software testing process
4.3. Black Box testing techniques
4.3.1. Equivalence partitioning technique
4.3.2. Boundary value analysis technique
4.3.3. Decision table technique
4.3.4. State transition testing technique
4.3.5. Pairwise testing technique
4.4. Summarization and Exercise
Chapter 5: White Box testing
5.1. Objectives
5.2. White Box testing technique
5.2.1. Control flow testing technique
5.2.2. Data flow testing technique
5.3. Automation unit test
5.4. Summarization and Exercise
Chapter 6: SQA tools
6.1. SQA management tools
6.2. Unit testing tool
6.3. Automated functional testing tool
6.4. Performance testing tool
6.5. Security testing tool
6.6. AI tools for testing
6.7. Applying SQA tools in projects
6.8. Summarization and Exercise
Chapter 7: Standards in SQA management
7.1. The scope of quality management standards
7.2. SQA in ISO standard
7.3. SQA in CMM, CMMI standard
7.4. Guidelines on the testing of AI-based systems ISO/IEC TR 29119-11:2020
7.5. Summarization and Exercise

9. Required Textbooks

[1] Murali Chemuturi. *Mastering Software Quality Assurance: Best Practices, Tools and Techniques for Software Developers*. J. Ross Publication Inc., 2011.

10. Suggested Textbooks

[2]. Neil Walkinshaw, *Software Quality Assurance Consistency in the Face of Complexity and Change,* Springer Nature, 2017.
[3] Renzo Cerquozzi, Wim Decoutere, Klaudia Dussa-Zieger, Jean-François Riverin, Arnika Hryszko, Martin Klonk, Michaël Pilaeten, Meile Posthuma, Stuart Reid, Eric Riou du Cosquer (chair), Adam Roman, Lucjan Stapp, Stephanie Ulrich (vice chair), Eshraka Zakaria, *Certified Tester Foundation Level (CTFL) v4.0*, International Software Testing Qualifications Board, 2023.
[4]. Paul Ammann and Jeff Offutt. *Introduction to Software Testing*. Cambridge University Press, 2016.
[5]. ISO/IEC TR 29119-11:2020(en)— Part 11: Guidelines on the testing of AI-based systems - https://www.iso.org/obp/ui/en/#iso:std:iso-iec:tr:29119:-11:ed-1:v1:en (Nov 03rd 2023)

11. Schedules

Main contents	Duration	Specific contents
Chapter 1: Introduction of software quality assurance	4h lecture	1.1.What is software? 1.2. Classification of the causes of software errors 1.3. Software quality assurance – definition and objective 1.4. Software quality factors. 1.5. Factors affecting intensity of quality assurance 1.6. Summarization and Exercise
Chapter 2: Integrating quality activities in the project life cycle	4h lecture	2.1. Software development methodologies 2.2. Testing level 2.2.1. Introduction 2.2.2. Unit test 2.2.3. Integration test 2.2.4. System test 2.2.5. Acceptance test 2.3. A model of SQA defect removal effectiveness and cost. 2.4. Create SQA plan 2.5. Summarization and Exercise
Chapter 3: Reviews	6h lecture 2h exercise	3.1. Review objectives 3.2. Formal review 3.3. Peer reviews 3.4. Implementation of reviews activity in projects 3.4.1. Reviewing requirement specification 3.4.2. Reviewing design document 3.4.3. Reviewing development plan and SQA plan 3.4. Summarization and Exercise
Chapter 4: Black Box testing	6h lecture 2h exercise	4.1. Definition and objectives 4.2. Software testing process 4.3. Black Box testing techniques 4.3.1. Equivalence partitioning technique 4.3.2. Boundary value analysis technique 4.3.3. Decision table technique 4.3.4. State transition testing technique 4.3.5. Pairwise testing technique 4.4. Summarization and Exercise
Chapter 5: White Box testing	6h lecture 2h exercise	5.1. Objectives 5.2. White Box testing technique 5.2.1. Control flow testing technique 5.2.2. Data flow testing technique 5.3. Automation unit test

		5.4. Summarization and Exercise
Chapter 6: SQA tools	4h lecture 2h exercise	6.1. SQA management tools 6.2. Unit testing tool 6.3. Automated functional testing tool 6.4. Performance testing tool 6.5. Security testing tool 6.6. AI tools for testing 6.7. Applying SQA tools in projects 6.8. Summarization and Exercise
Chapter 7: Standards in SQA management	4h lecture	7.1. The scope of quality management standards 7.2. SQA in ISO standard 7.3. SQA in CMM, CMMI standard 7.4. Guidelines on the testing of AI-based systems ISO/IEC TR 29119-11:2020 7.5. Summarization and Exercise
Summary	2h lecture	

12. Grading Policy

Attendance:	10%
Exercises:	20%
Mid-term projects/exams:	20%
Final examination (lab):	50%