Universitat de Lleida

DEGREE CURRICULUM

QUALITY MANAGEMENT AND IMPROVEMENT

Coordination: GARRIDO NAVARRO, JUAN ENRIQUE

Academic year 2022-23

Subject's general information

Subject name	QUALITY MANAGEMENT AND IMPROVEMENT
Code	102053
Semester	2nd Q(SEMESTER) CONTINUED EVALUATION

Typology	Degree	Course	Character	Modality
	Bachelor's Degree in Computer Engineering	3	COMPULSORY	Attendance-based
	Bachelor's Degree in Computer Engineering	3	OPTIONAL	Attendance-based

Course number of credits (ECTS)	6

Type of activity, credits, and groups	Activity type	PRALAB	TEORIA
	Number of credits	3	3
	Number of groups	1	1

Coordination	GARRIDO NAVARRO, JUAN ENRIQUE
Department	COMPUTER SCIENCE AND INDUSTRIAL ENGINEERING
Teaching load distribution between lectures and independent student work	20% On-site 20% Online 60% Autonomous work
Important information on data processing	Consult this link for more information.
Language	English

QUALITY MANAGEMENT AND IMPROVEMENT 2022-23

Teaching staff	E-mail addresses	Credits taught by teacher	Office and hour of attention
GARRIDO NAVARRO, JUAN ENRIQUE	juanenrique.garrido@udl.cat	3	
SALES JOVÉ, SERGI	sergi.sales@udl.cat	3	

Subject's extra information

This course is part of the specialisation in Software Engineering of the Degree in Computer Engineering offered by the Polytechnic School at the University of Lleida.

A **Joint Project** is developed together with other subjects in the third course of the "Software Engineering" specialization:

- Quality Management and Improvement
- Process Model
- Requirements Engineering

Each subject is focused on its corresponding aspects of the project development. In the case of this subject, the focus is software quality management and improvement. The aim of this **Joint Project** is to bring the students with what would be a real-world project of Software Engineering and at the same time, we see that results of the aspects studied in the contexts of a course serve to meet the challenges presented in other courses.

To follow this course properly, some previous knowledge/skills on Software Engineering are recommended. A good level of English (reading, writing, oral) is also recommended.

Learning objectives

- To understand that software quality goes beyond the quality of the code.
- To design a reasonably professional and rigorous software quality plan.
- To implement key aspects of a software quality plan, i.e., conduct software quality assurance tasks, in an effective way.
- To be conversant with the field of software quality assurance (metrics, tools, costs, etc.).
- To keep strengthening important skills good software engineers should have (i.e. ability to make decisions, to meet deadline, to work in teams, to write technical documentation and make oral presentations).

Competences

Strategic Competences of the UdL

- CT2. Mastering a foreign language, especially English.
- CT3. Training Experience in the use of the new technologies and the information and communication technologies.

Cross-disciplinary competences

- EPS11. Capacity to understand the needs of the user expressed in a no technical language.

Specific competences (Software Engineering)

- GII-IS1. Capacity to develop, maintain and evaluate services and software systems that satisfy all the requirements of the user and behave in a reliable and efficient way, they can develop, keep and fulfil quality requirements, applying the theories, principles, methods and uses of the software engineering.
- GII-IS5. Capacity to identify, evaluate and manage the potential risks that can arise.

Subject contents

Block I

- Chapter 1. Introduction to Software Quality
- Chapter 2. Quality Management
- Chapter 3. Further Aspects

Block II

- Chapter 4. Testing and Metrics
- Chapter 5. Introduction to costs
- Chapter 6. Standards and Models

Methodology

Classes for the course include sessions in which the contents are introduced, through support of educational materials prepared or compiled by the lecturer, and group work sessions.

In group work sessions, we use the project-based learning methodology. So, students must apply techniques, concepts and tools learned in the subject under the common project to the three courses in the third year of Software Engineering module.

Theory

- Presentation of the contents of the course.
- Discussion with the students.

Activities

- Activity A. To investigate aspects related to quality.
- Activity B. To design and implement useful items in quality management.
- Activity C. To work and investigate aspects related to tests and metrics.

Joint Project

- Deliverable 1. Definition of a Quality Plan (and oral presentation).
- Deliverable 2. Implementation of a Quality Plan (and oral presentation).
- Oral presentation of the project to a committee involving the professors of the three subjects. The presentation, common to these subjects, addresses the overall project development.

A note about Joint Project: Both activities are conducted within the context of the Software Engineering Joint Project developed in three courses (Requirements Engineering, Process Models and Quality Management and Improvement). Given that each course addresses important aspects of Software Engineering (e.g. requirements, agile methodologies and quality), the project aims to encourage students to deal with them in a real-life scenario, which is simulated in this project.

Development plan

Week	Course Activities	Joint Project Activities	Autonomous Student Work
1	Presentation, Ch1 and Activity A		Study and Activity A development
2	Ch1 (continuation) and Activity A (continuation)		Study and Activity A development
3	Ch2, Activity A (submission) and Activity B		Study and Activity B development
4	Ch2 (continuation) and Activity B (continuation)		Study and Activity B development
5	Ch3 and Activity B (submission)	Deliverable 1	Study and Deliverable 1 development
6	Ch3 (continuation)	Deliverable 1	Study and Deliverable 1 development
7	Ch3 (continuation)	Deliverable 1	Study and Deliverable 1 development
8	Written Exam		Study
9	Ch4	Deliverable 1 (submission and presentation)	Study
10	Ch4 (continuation), Activity C	Deliverable 2	Study, Deliverable 2 development and Activity C development
11	Ch5 and Activity C (continuation)	Deliverable 2	Study, Deliverable 2 development and Activity C development
12	Ch5 (continuation) and Activity C (submission)	Deliverable 2	Study, Deliverable 2 development
13	Ch5 (continuation)	Deliverable 2	Study and Deliverable 2 development
14	Ch6	Deliverable 2	Study and Deliverable 2 development
15	Ch6 (continuation)	Deliverable 2	Study and Deliverable 2 development
16-18	Written Exam	Deliverable 2 (submission and presentation)	Study
19	Tutorials		
20	Remedial Exam		Study

Evaluation

Acr	Evaluation Activity	Rate	Minimum Qualification	Components	Rate	Group Activity	Remedial Exam
Ind	Individual Activities	20%	NO	Activity A	30%	NO	NO
				Activity B	35%	NO	NO
				Activity C	35%	NO	NO
D1	Deliverable 1	20%	NO			YES	NO
D2	Deliverable 2	20%	NO			YES	NO
FP	Final Presentation	10%	NO			YES	NO
Theo	Theory	30%	5	1st Written Exam	50%	NO	YES
				2nd Written Exam	50%	NO	YES

Final Qualification = 0,2*Ind + 0,2*D1 + 0,2*D2 + 0,1*FP + 0,3*Theo

If the minimum grade is not reached in the Theory Activity, the maximum grade that will appear in the minutes will be 4, regardless of whether the application of the indicated percentages may lead to a higher result.

Bibliography

Mario Piattini, Félix García, Ignacio García. Calidad de Sistemas de Información. 3rd edition. Ra-Ma Editorial. 2015.

Ian Sommerville. Software Engineering. Ninth version. Addison-Wesley, 2011.

Ronald Kirk. Software Engineering Quality Practice. Auerback Publications, 2006.

Jeff Tian. Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement. Wiley - IEEE Computer Society Press, 2005.

Daniel Galin. Software Quality Assurance. From theory to implementation. Pearson Education Limited, 2004.

Daniel Galin. Software Quality: Concepts and Practice. From theory to implementation. Wiley-IEEE Computer Society, 2018

Stephen Kan. Metrics and Models in Software Quality Engineering. 2nd Edition. Addison-Wesley, 2002.

Gordon Schulmeyer. Handbook of Software Quality Assurance. 4th edition. Artech House, 2002.

Daniel Galin. Software Quality: Concepts and Practice. From theory to implementation. Wiley-IEEE Computer Society, 2018

Stephen Kan. Metrics and Models in Software Quality Engineering. 2nd Edition. Addison-Wesley, 2002.

Gordon Schulmeyer. Handbook of Software Quality Assurance. 4th edition. Artech House, 2002.