



Universitat de Lleida

# DEGREE CURRICULUM **QUALITY MANAGEMENT AND IMPROVEMENT**

Coordination: GARRIDO NAVARRO, JUAN ENRIQUE

Academic year 2017-18

## Subject's general information

<b>Subject name</b>	QUALITY MANAGEMENT AND IMPROVEMENT			
<b>Code</b>	102053			
<b>Semester</b>	2nd Q(SEMESTER) CONTINUED EVALUATION			
<b>Typology</b>	Degree	Course	Typology	Modality
	Bachelor's Degree in Computer Engineering	3	COMPULSORY	Attendance-based
<b>ECTS credits</b>	6			
<b>Groups</b>	1GG			
<b>Theoretical credits</b>	2			
<b>Practical credits</b>	4			
<b>Coordination</b>	GARRIDO NAVARRO, JUAN ENRIQUE			
<b>Department</b>	INFORMATICA I ENGINYERIA INDUSTRIAL			
<b>Teaching load distribution between lectures and independent student work</b>	6 ECTS x 25h = 150h 40% -> 60h classroom hours (4h per week) 60% -> 90h autonomous work			
<b>Important information on data processing</b>	Consult <a href="#">this link</a> for more information.			
<b>Language</b>	English			
<b>Office and hour of attention</b>	Office: 3.18 (EPS). Hour of attention: send me an e-mail.			

Teaching staff	E-mail addresses	Credits taught by teacher	Office and hour of attention
GARRIDO NAVARRO, JUAN ENRIQUE	juanenrique.garrido@diei.udl.cat	7,2	Office 3.18, appointment by email.

## 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 3rd course of the "Software Engineering" specialization:

- Quality Management and Improvement \*
- Process Model \*
- Requirements Engineering \*

Each subject focused on the corresponding aspects of project development. In the case of this subject, 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.

(\*) These three subjects must be enrolled simultaneously because they work together on a Joint Project. The only exception is if any of them is approved.

## 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 non 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

### *Key and basic aspects of software quality*

- Ch1. Introduction to Software Quality Assurance (SQA).
- Ch2. SQA in development.
- Ch3. Quality aspects in software design and programming.
- Ch4. Further aspects of software testing.
- Ch5. SQA post-development.

### *Other aspects of software quality. More advanced aspects of SQA*

- Ch6. Introduction to costs in SQA.
- Ch7. SQA from quality experts.
- Ch8. A good SQA engineer.

## Methodology

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

In group work sessions, we use project-based learning methodology. So, students must apply several techniques on 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.
- Some sessions will be devoted to Deliverable 1 and 2.

### **Activities**

- Activity A. Design Reviews and Program Inspections.
- Activity B. Software testing and software metrics (tools).

### **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 all 3 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). 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 Ch2		Study
2	Ch2 (continuation) and Activity A		Study and Activity A Development
3	Ch2 (continuation) and Activity A (continuation)		Study and Activity A Development
4	Ch2 (continuation), Activity A (submission) and Activity B		Study and Activity B Development
5	Ch3 and Activity B (continuation)		Study and Activity B Development
6	Ch4 and Activity B (submission)	Deliverable 1	Study and Deliverable 1 Development
7	Ch5	Deliverable 1	Study and Deliverable 1 Development
8		Deliverable 1	Study and Deliverable 1 Development
9	1st Midterm Exam		Study
10		Deliverable 1 (Submission and Presentation) and Deliverable 2	Study and Deliverable 2 Development
11	Ch5 (continuation)	Deliverable 2	Study and Deliverable 2 Development
12	Ch6	Deliverable 2	Study and Deliverable 2 Development
13	Ch7	Deliverable 2	Study and Deliverable 2 Development
14	Ch8	Deliverable 2	Study and Deliverable 2 Development
15		Deliverable 2	Study and Deliverable 2 Development
16-17	Written Exam	Deliverable 2 (Submission and Presentation)	Study
18		Project Presentation	Presentations Preparation
19	Remedial Exam		Study

## Evaluation

Acr.	Evaluation Activity	Rate	Minimum Qualification	Group Activity	Mandatory	Remedial Exam
ActA	Activity A	12,5%	4	NO	YES	YES
ActB	Activity B	12,5%	4	NO	YES	YES
D1	Deliverable 1	20%	NO	6 to 7	YES	NO
D2	Deliverable 2	20%	NO	6 to 7	YES	NO

FP	Final Presentation	10%	NO	6 to 7	YES	NO
1st	1st Midterm Exam	12,5%	4	NO	YES	YES
2nd	2nd Midterm Exam	12,5%	4	NO	YES	YES
Final Qualification = $0,125 \cdot \text{ActA} + 0,125 \cdot \text{ActB} + 0,2 \cdot \text{D1} + 0,2 \cdot \text{D2} + 0,1 \cdot \text{FP} + 0,125 \cdot \text{1st} + 0,125 \cdot \text{2nd}$						

## Bibliography

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

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

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

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

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

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