



Universitat de Lleida

DEGREE CURRICULUM  
**REQUIREMENTS ENGINEERING**

Coordination: OLIVA SOLE, MARTA

Academic year 2020-21

## Subject's general information

<b>Subject name</b>	REQUIREMENTS ENGINEERING			
<b>Code</b>	102052			
<b>Semester</b>	2nd Q(SEMESTER) CONTINUED EVALUATION			
<b>Typology</b>	Degree	Course	Character	Modality
	Bachelor's Degree in Computer Engineering	3	COMPULSORY	Attendance-based
<b>Course number of credits (ECTS)</b>	6			
<b>Type of activity, credits, and groups</b>	<b>Activity type</b>	PRALAB	TEORIA	
	<b>Number of credits</b>	4	2	
	<b>Number of groups</b>	1	1	
<b>Coordination</b>	OLIVA SOLE, MARTA			
<b>Department</b>	COMPUTER SCIENCE AND INDUSTRIAL ENGINEERING			
<b>Teaching load distribution between lectures and independent student work</b>	ECTS 25x6 = 150 = 6h - 40% -> 60 classroom hours. - 60% -> 90 hours autonomous work.			
<b>Important information on data processing</b>	Consult <a href="#">this link</a> for more information.			
<b>Language</b>	Catalan and English. The materials and resources will be provided in English, but the lectures will be in Catalan.			

Teaching staff	E-mail addresses	Credits taught by teacher	Office and hour of attention
OLIVA SOLE, MARTA	marta.oliva@udl.cat	6	arrange by email

## Subject's extra information

Course taught during the 2nd semester of 3rd year of the degree. It is part of the specialization module in "Software Engineering", bearing in mind that to develop good software their requirements should be well identified, documented and validated.

A **Joint Project** is developed together with other subjects in the 3rd course of the "Software Engineering" specialization:

- Quality Management and Improvement
- Process Models
- Requirements Engineering

Each subject focuses on the corresponding aspects of project development. In the case of this subject, requirements elicitation and specification. 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.

The knowledge acquired in this course will be applicable in most careers, especially for those who are dedicated to develop applications.

## Learning objectives

- Understand the importance of discovering the requirements of a project.
- Determine the scope of a project.
- Practice different techniques for requirements elicitation (functional and non functional).
- Understand what involves an iterative process for requirements engineering.
- Describe scenarios.
- Specify requirements through user stories.
- Analyze the quality of each specific requirement.
- Assess the specification requirements.
- Analyze the risks.

## 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

- 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-IS2. Capacity to value the customer needs and specify the software requirements to satisfy these needs, solving conflicts by means of acceptable commitments taking into account the limitations of cost, time, existence of systems already developed and of the own organisations.
- GII-IS3. Capacity to give solution to problems of integration taking into account the strategies, standards and available technologies.

## Subject contents

- *Why developing software needs more time or more money than you initially expected?*
- *Why are there so many programs that, upon completion of their development, fail?*
- *Why users are usually not completely satisfied by software?*

The answer to these questions is related to how to include requirements engineering in software development. Not doing so usually leads to uncomplete or unclear requirements and, consequently, to software that does not satisfy users.

The contents of this course include various aspects to be treated in the requirements engineering phase, among other.

1. Introduction
2. The Requirement Process
3. Scoping the Business Problem
4. Investigating the work - Trawling techniques
5. Scenarios
6. Requirements and Iterative Development - User Stories
7. Quality Gateway
8. Non-Functional Requirements - Fit criterion
9. Requirements evaluation - Conflicting Requirements, Risk Analysis
10. Reusing Requirements

## Methodology

The classes of the subject include sessions in which the contents are debated (virtual sessions) and group work sessions in person (PraAula). In group work sessions, the project-based learning methodology is used. Thus, the students put into practice the techniques proposed in the subject within the framework of a project common to the three third-year subjects of the intensification of Software Engineering.

### Theory

- Requires prior review of teaching materials prepared or compiled by teachers.
- Discussion / Debate with the students of the contents corresponding to the session. The session is conducted in such a way that all the most relevant contents are reviewed and that they will end up being assumed during the development of the proposed project.

### Team work

- From the proposed project, the students progress in small weekly activities that end up forming each of the required deliverables.

### Joint Project

- Deliverable 1. First Iteration Requirements Modelling
- Deliverable 2. Second and third iteration Requirements Modelling
- Deliverable 3. Requirements Evaluation & Risk Analysis
- 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:** all 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 course). 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	Introduction	-	Presentations Preparation
2	The Requirement Process (students presentation)	Project Presentation	Study
3	Scoping the Business Problem Business Use Cases	Project launch (Scope, Stakeholders and Goals)	Study, Project Development
4	Investigating the work	Context diagram (Data Flow, Events)	Presentations Preparation
5	Trawling Techniques (students presentation)	Trawling Techniques Plan Requirements Trawling	Project Development
6	Scenarios	Requirements Trawling	Project Development
7	Requirements and Iterative Development Functional Requirements	Functional Requirements Modelling	Study, Project Development
8	User Stories	Functional Requirements Modelling	Project Development
9	1st Midterm Exam		Study
10	Non-Functional Requirements - Fit Criterion	Non-Functional Requirements Modelling	Study, Project Development
11	Quality Gateway	Non-Functional Requirements Modelling	Study, Project Development
12	Requirements Evaluation (Requirements Completeness)	Requirements Evaluation/Modelling (Requirements Completeness)	Project Development
13	Requirements Evaluation (Conflicting Requirements)	Requirements Evaluation/Modeling (Conflicting Requirements)	Study, Project Development
14	Requirements Evaluation (Risk Analysis)	Requirements Evaluation/Modeling (Risk Analysis)	Project Development
15	Reusing Requirements	Requirements Evaluation/Modelling (Risk Analysis)	Project Development
16-17	1st Midterm Exam	-	Study
18	-	Project Presentation	Presentations Preparation
19	Remedial Exam		Study

## Evaluation

Acr.	Evaluation Activity	Rate	Minimum Qualification	Group Activity	Mandatory	Remedial Exam
P1	1st Midterm Exam	20%	4	NO	YES	YES
P2	2nd Midterm Exam	20%	4	NO	YES	YES
D1	1st Project Deliverable	20%	NO	4 to 6	YES	NO
D2	2a Project Deliverable	15%	NO	4 to 6	YES	NO
D3	3a Project Deliverable	15%	NO	4 to 6	YES	NO
FP	Final Presentation	10%	NO	4 to 6	YES	NO
Final Qualification = $0,2 \cdot P1 + 0,2 \cdot P2 + 0,2 \cdot D1 + 0,15 \cdot D2 + 0,15 \cdot D3 + 0,1 \cdot FP$						

## Bibliography

- S. Robertson & J. Robertson.** *Mastering the Requirements Process: Getting Requirements Right* (3rd ed.). Addison-Wesley, 2012.
- D. Leffingwell.** *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*, Addison Wesley, 2011.
- M. Cohn.** *User Stories Applied: For Agile Software Development*, Addison Wesley, 2004.
- K. Pohl & C. Rupp.** *Requirements Engineering Fundamentals: A Study Guide for the Certified Professional for Requirements Engineering Exam - Foundation Level - IREB compliant*. Rocky Nook Computing, 2011.
- E. Hull, K. Jackson & J. Dick.** *Requirements Engineering* (3rd edition), Springer, 2011.
- K. Pohl.** *Requirements Engineering. Fundamentals, Principles, and Techniques*. Springer, 2010.
- K. Wiegers & J. Beatty.** *Software Requirements* (3rd edition). Microsoft Press, 2013.