Software Requirements

Code: 102763
ECTS Credits: 6

Degree	Type	Year	Semester
2502441 Computer Engineering	OB	3	1
2502441 Computer Engineering	OT	4	1

The proposed teaching and assessment methodology that appear in the guide may be subject to changes as a result of the restrictions to face-to-face class attendance imposed by the health authorities.

Contact

Name: Daniel Ponsa Mussarra

Email: Daniel.Ponsa@uab.cat

Use of Languages

Principal working language: catalan (cat)

Some groups entirely in English: No

Some groups entirely in Catalan: Yes

Some groups entirely in Spanish: No

Prerequisites

The subject has no prerequisites. However, students should be familiar with the most basic concepts of Object Oriented Analysis and Design, as well as UML. Students should review these topics, which are part of the subject in Software Engineering given in the second year.

Objectives and Contextualisation

The training objectives of the subject are:

- Study the main methodologies proposed to manage the requirements of a software project.
- Understand the problem of Requirements Engineering, and its relationship with other phases of Software Engineering.
- Know the participants involved in Requirement Engineering.
- Acquire practical experience in the application of techniques for collecting and managing requirements.

Competences

Computer Engineering
- Acquire personal work habits.
- Capacity to determine the requirements of information and communication systems in an organisation attending to security aspects and fulfilment of applicable standards and legislation.
- Communication.
- Have the capacity to appraise the customers needs and specify the software requirements to satisfy these needs, reconciling conflictive aims by means of a search for acceptable commitments within the derived limitations in terms of cost, time, the existence of systems that have already been developed and those of organisations themselves.
- Have the capacity to conceive, develop and maintain computer systems, services and applications employing the methods of software engineering as an instrument to ensure quality.
- Have the capacity to design appropriate solutions in one or more applicable domains employing software engineering methods that integrate ethical, social, legal and economic aspects.

- Have the capacity to identify and analyse problems and design, develop, implement, verify and document software solutions on the basis of suitable knowledge of current theories, models and techniques.
- Know and apply basic elements of economics, human resource management, project organisation and planning, as well as legislation, regulation and standardisation in the field of computer projects.

Learning Outcomes

1. Adapt software design to applicable security standards.
2. Analyse and evaluate requirements taking into account the existing limitations.
3. Analyse and evaluate requirements.
4. Communicate efficiently, orally or in writing, knowledge, results and skills, both in the professional environment and before non-expert audiences.
5. Design specifications that integrate the legal, ethical, social and economic restrictions of information flow.
6. Determine the requirements of computer applications in an organisations information systems.
7. Identify and model cases that employ UML.
8. Plan software development by taking into account the available resources.
9. Specify the clients needs in a software specification document.
10. Work independently.

Content

- Introduction to Requirements Engineering.
    - Motivation of Requirements Engineering.
    - Definitions
    - The 3 Dimensions of Requirements Engineering.
    - Knowledge and skills of a Requirements Engineer.
    - Context of a System.
    - Types of requirements.
- Requirements Elicitation
    - System context and elicitation.
    - Types of requirements and elicitation.
    - Identification and selection of sources.
    - Elicitation Techniques.
    - Elicitation support techniques.
- Requirements documentation.
    - Documentation techniques.
    - Standard document templates.
    - Uses of requirements specification documents.
    - Quality criteria.
- Documentation based on Natural Language.
    - Ambiguity
    - Syntactic patterns
    - Structure of natural language.
    - Requirement Attributes.
    - Acceptance criteria
- Model-based documentation.
    - Modeling languages.
    - Transformational effects of models.
    - Modeling of Objectives (Goals).
    - Scenario modeling.
    - Modeling of Requirements.
- Conflict management
    - Objectives and scenarios in the negotiation of requirements.
    - Conflict management process.
    - Principle-based Negotiation.

- Requirement validation.
  - Validation of objectives, scenarios and requirements.
  - Quality Aspects of requirements.
  - The 6 principles of validation.
  - Validation techniques.
- Requirements engineering management.
  - Context management.
  - Process management.
  - Artifact management.
- Requirements-based Testing.
  - Test cases.
  - Types of test.
  - Test levels
  - Definition of test cases based on requirements.

Methodology

The different activities that will be carried out in the subject are organized as follows:

Theory sessions: Students will have a series of videos on the virtual campus explaining the basic concepts of the subject, which they will have to view weekly following an established schedule. These videos may contain interactive activities that students must complete and pass, as well as associated supplementary questionnaires, to be answered via virtual campus, which will form part of the continuous assessment of students. A calendar of telematic theory sessions will also be established where doubts that may arise from the viewing of videos will be resolved and complementary activities will be carried out that will form part of the continuous assessment of the students.

Problem sessions: Synchronous telematic sessions will be held where topics introduced in theory classes will be extended in a practical way. These sessions will therefore require that students have previously worked on the material associated with the corresponding theory classes. In these sessions problems will be solved and case studies will be discussed. With the proposed activities, autonomous and cooperative work, the capacity for analysis, synthesis and critical reasoning will be promoted, and the student will be trained in problem solving.

Practices: The practices will be carried out telematically in teams of 4 or 5 people. A series of activities will be scheduled to collect and document the requirements of a software system, with an established delivery schedule. The teams will carry out some of these activities in synchronous telematic sessions, in which their work will also be monitored and where they will receive feedback on the work done.

Transversal Competences.

The subject has two transversal competences assigned. Below are detailed, specifying for each one of them what activities will be worked on and how they will be evaluated.

T02.01 - Work independently.
The planning of the study, the assistance to tutorials, as well as the preparation of the entrusted activities, is something that the student must manage autonomously, and this good management affects the qualifications of their work. However, this competence will be directly evaluated in the following activities:

- Characterization of the context of a software application. In the practical work, based on a software problem described globally, the student must characterize on his/her own the different facets of the problem (thematic and technological aspects, use and development issues) looking for pertinent information sources. Based on the quality of this problem characterization the autonomous work performed is evaluated.
- Selection of development tools. Part of the practical work requires the generation of modeling diagrams. Instead of setting the tools that will be used, in the practical work it is requested to carry out a search of the available tools, making a justified choice of the tool that will be used in the practice carried out.

T04.01 - Communicate efficiently, orally or in writing, knowledge, results and skills, both in the professional environment and before non-expert audiences.

Communication skills are evaluated in the practice of the subject, based on

- Requirement elicitation interviews, which they carry out with a hypothetical user of the software in which their practice is focused.
- The documentation in natural language of the information gathered in the vision and the software specification document.
- The UML modeling of certain aspects of the system requested in the practical work to be done.

General considerations

The 'Campus virtual' platform will be used to disseminate information to students. The dates of continuous evaluation and delivery of works will be published through this medium, and may be subject to possible programming changes for reasons of adaptation to possible incidents. 'Campus virtual' will be used to inform about these possible changes, since this is the platform for the exchange of information between the teaching staff and the students.

Activities

Title	Hours	ECTS	Learning Outcomes
Type: Directed			
Evaluation tests	4	0.16	2, 3, 6, 5, 9, 7, 8
Exercise-based classes	12	0.48	2, 3, 6, 7, 10
Practical work sessions	12	0.48	1, 2, 3, 4, 6, 5, 9, 7, 10
Theoretical classes / lectures	22	0.88	2, 3, 6, 5, 7, 8
Type: Supervised			
Exercise resolution outside class	24	0.96	10
Preparing practical assignments	24	0.96	10
Type: Autonomous			
Study	50	2	10
Tutoring and consultations	2	0.08	10

Assessment

a) Processes and scheduled evaluation activities

The evaluation of the subject will be carried out continuously based on the learning evidences collected in the following processes:

- [E1]. Carrying out written tests (exams).
- [E2]. Resolution and delivery of questionnaires and exercises proposed in the sessions of theory and problems.
- [E3]. Completion of practical work, evaluated from different activities and deliveries.

The subject consists of the following assessment activities, each evaluated with a score between 0 and 10 (inclusive):

- [E1]-ExP1, partial exam 1, 30% on the final grade.
- [E1]-ExP2, partial exam 2, 20% on the final grade.
- [E2]-Prob, resolution of exercises proposed in the sessions of theory and problems, 10% of the final grade.
- [E3]-PracE, practical activities related to the eligibility of requirements, 15% on the final grade.
- [E3]-PracD, practical activities related to the documentation of requirements, 25% on the final grade.

In order to be able to pass the subject through continuous evaluation, a minimum grade of 5 must be drawn in the following 2 expressions.

- (0.6 * Grade [E1]-ExP1) + (0.4 * Grade [E1]-ExP2) + (0.1 * Grade [E2]-Prob)
- (0.3 * Grade [E1]-ExP1) + (0.2 * Grade [E1]-ExP2) + (0.1 * Grade [E2]-Prob) + (0.15 * Grade [E3]-PracE) + (0.25 * Grade [E3]-PracD)

Keep in mind that:

- If a value inferior of 5 is obtained in the first expression, this value will be assigned as the final grade of the subject.
- Exercises in [E2]-Prob must be delivered within an established period, and will be evaluated with a score between 0 and 10 (both inclusive). Exercises not delivered within their term will be evaluated with a score of 0, and can not be recovered.
- Activities [E3]-PracE and [E3]-PracD will be evaluated based on different sub-activities proposed, which will have a delivery deadline fixed.Each sub-activity will be evaluated with a score between 0 and 10 (both inclusive). Subactivities not made or delivered out of their term will be evaluated with a grade of 0, and can not be recovered.

In case of irregularities in the evaluation activities, the one detailed in section f) will be applied.

It is important to bear in mind that evaluation activities will not be carried out on a date or time different from the established one, except for justified causes, duly informed in advance to the teaching staff.

b) Programming of evaluation activities

The scheduling of the different assessment activities will be detailed in the Virtual Campus, in the Moodle classroom of the subject. The dates of the written tests (activities [E1]-ExP1 and [E1]-ExP2) will also be made public on the website of the School of Engineering, in the section of exams.

c) Recovery process

The only recoverable evaluation activities are the written tests [E1]-ExP1 and [E1]-ExP2.

The student has the option to improve the grades of these tests (one of them, or both) provided he/she has been evaluated in a set of activities that represent a minimum of two thirds of the total grade of the subject.

In order to compute the final grade of the subject, the mark obtained in the recovery exam will replace the one of the corresponding exam carried out in the continuous assessment.

In accordance with the coordination of the Degree and the direction of the School of Engineering the following activities can not be recovered:

- [E2]-Prob, 10% on the final grade.
- [E3]-PracE, 15% on the final grade.
- [E3]-PracD, 25% on the final grade.

d) Procedure for the review of qualifications

For assessment activities based on written tests ([E1]-ExP1 and [E1]-ExP2), , a procedure for booking a revision date ant time will be established in which the studentwill be able to review the activity with the teaching staff. In thiscontext, claims can be made about the activity grade, which will be evaluated by the teachers responsible for the subject. Likewise, it is possible to arrange with the teaching staff the review of the rest of the assessment activities up to two weeks before the recovery exams

e) Special qualifications

If the student has not performed any of the tests [E1]-ExP1 and [E1]-ExP2 the "Non-assessable" grade will be assigned. It must be remarked that according to current regulations "Non-assessable" qualifications also exhaust convocation.

In order to pass the course with honours, the final grade must be a 9.0 or higher. Because the number of students with this distinction cannot exceed 5% of the number of students enrolled in the course, this distinction will be awarded to whoever has the highest final grade. In case of a tie, partial-test results will be taken into consideration.

f) Irregularities by the student, copy and plagiarism.

Notwithstanding other disciplinary measures deemed appropriate, assessment activities will receive a zero whenever a student commits academic irregularities that may alter such assessment. Therefore, copying, plagiarizing, cheating, ... in any of the assessment activities will imply suspending it with a zero.

g) Evaluation of repeating students

From the second enrollment, repeater students may request to validate the evaluation of the activities [E3]-PracE and [E3]-PracD, taking the mark obtained the first time that the student enrolled in the subject. In order to be able to opt for this differentiated evaluation, repeater students must ask the faculty through an email.

Assessment Activities

Title	Weighting	Hours	ECTS	Learning Outcomes
[E1]-ExP1: Midterm Exam 1	30	0	0	1, 2, 3, 6, 5, 9, 7
[E1]-ExP2: Midterm Exam 2	20	0	0	1, 2, 3, 6, 5, 9
[E2]-Prob: Delivered exercises	10	0	0	2, 3, 6, 5, 9, 7
[E3]-PracD: Practical work:Documentation	25	0	0	1, 2, 3, 4, 6, 5, 9, 7, 8, 10
[E3]-PracE: Practical work:Elicitation	15	0	0	1, 2, 3, 4, 6, 5, 9, 7, 8, 10

Bibliography

- K. Pohl. Requirements Engineering. Fundamentals, Principles and Techniques. Springer, 2010
- K. Pohl, C. Rupp. Requirements Engineering Fundamentals. Rocky Nooc Inc, 2011
- R. Fisher, W. Ury, B. Patton. Getting to Yes:Negotiating Agreement Without Giving In, 2nd Edition. New York:Penguin Books. 1991
- D. Gray, S. Brown, J. Macanufo. Gamestorming: A Playbook for Innovators, Rulebreakers, and Changemakers. O'Reilly, 2010
- Thomas A. Pender. UML Weekend Crash Course. John Wiley & sons, 2002
- M. Delevic. Guide to the Logical Framework Approach, 2nd Edition. Republic of Serbia Government. European Integration Office, 2011
- Roger S. Pressman. Ingeniería del software. Un enfoque pràctico. 7ª Edición. McGraw Hill, 2010
- Roman Pichler. Strategize. Pichler consulting, 2016

- Jeff Patton with Peter Economy. User Story Mapping. O'Really, 2014
- Gojko Adzic and David Evans. Fifty Quick Ideas To Improve User Stories. Neuri Consulting LLP, 2014
- Roman Pichler. Agile Product Management with Scrum. Addison-Wesley, 2010