

Degree	Type	Year
Computer Engineering	OB	3
Computer Engineering	OT	4

Contact

Name: Antonio Manuel Lopez Peña

Email: antoniomanuel.lopez@uab.cat

Teachers

Pablo Torrellas Perez

Rodolfo Alberto Guichon Aguilar

Teaching groups languages

You can view this information at the [end](#) of this document.

Prerequisites

This subject has a very practical character, but is based on the theoretical foundations taught in other subjects of the Software Engineering branch. Therefore, if you want to take this subject successfully, it is necessary to have previously studied:

- (1) Software requirements,
- (2) Software design,
- (3) Management and Administration of Databases,
- (4) Test and Quality of Software,
- (5) Software Development Management (this can be done in parallel, it is not necessary to have previously completed it).

Objectives and Contextualisation

This subject wants to provide a useful experience to students regarding what they can find when developing software projects professionally. Therefore, from the point of view of the students, it is about developing a complete and relatively long software project. In addition, the development process will be as or more important than the final result (software product). Therefore, the software should be developed in the most professional way possible, in particular, applying the best practices and working as a team.

Thus, the objectives of the subject are:

1. Work in a relatively large team, composed of sub-teams with different responsibilities.
2. Apply the theoretical knowledge that constitutes the best practices of software development.
3. Develop a complete software, that is, from the blank paper to an application, with its internal and external documentation, which satisfies the requirements of a client.
4. The development of this software will be also useful to gain experience in current SW environments.

The developed application, and the way it has been developed, should serve as a especially valuable curricular experience.

Competences

Computer Engineering

- Acquire personal work habits.
- Acquire thinking habits.
- Capacity to design, develop, evaluate and ensure the accessibility, ergonomics, usability and security of computer systems, services and applications, as well as of the information that they manage.
- Have the capacity to identify, evaluate and manage potential risks.
- Have the capacity to solve integration problems in accordance with available strategies, standards and technologies.
- Have the capacity to solve problems with initiative, decision making, autonomy and creativity. Have the capacity to know how to communicate and transmit the knowledge and skills of the IT engineering profession.
- Work in teams.

Learning Outcomes

1. Accept and respect the role of the various team members, and its different levels of dependence.
2. Adapt to unforeseen situations.
3. Critically evaluate the work done.
4. Define and manage the documentation generated during the development of a software application.
5. Demonstrate a high capacity for abstraction.
6. Design an architecture that can best solve a specific problem, taking into account the associated risks.
7. Design the architecture of a component based computer system.
8. Identify, manage and resolve conflicts.
9. Know how to communicate and transmit knowledge and skills in relation to the integration of software.
10. Make one's own decisions.
11. Manage time and resources available. Work in an organized manner .
12. Plan the integration of the different components developed in the coding process.
13. Select and use suitable CASE tools in each phase of software development.
14. With initiative and autonomy, resolve problems with the integration of software.
15. Work cooperatively.

Content

In this subject we do not seek to add more theoretical content to what is already developed in the other subjects of the Software Engineering branch. In other words, the theoretical knowledge of requirements capture, object-oriented design and coding, database management and administration, software testing and quality, and development management are already assumed. Thereofer, in this subject the focus in on putting all this into practice to develop a concrete application. Thus, as new content there are only two aspects:

1. Teamwork: roles and responsibilities of a software team, meetings, presentations, etc.
2. Agile methods of software development.

Activities and Methodology

Title	Hours	ECTS	Learning Outcomes
Type: Directed			
Presentations	16	0.64	3, 9
Project meetings	40	1.6	2, 1, 3, 4, 6, 8, 12, 14, 13, 15
Theory classes	15	0.6	1, 15
Type: Autonomous			
Study for individual exams	16	0.64	3
Tasks assigned to the project	134	5.36	2, 1, 3, 4, 5, 7, 6, 11, 8, 12, 10, 14, 13, 15

This course focuses on developing a software application in the most realistic way possible. Therefore, learning is done based on a practical case. In fact, there will be a case study for each practice team. A team will be formed by a group of between 6 and 9 students. On the one hand, this means that there will be a lot of autonomous work, but on the other hand, there will also be a lot of teamwork. Thus, the typology of the classes will support teamwork and software development based on an agile process. In particular, we will have:

Theory classes. They are intended to complement the aspects not covered by the other subjects of the specialty in Software Engineering, mainly teamwork.

Project meetings. Any team that develops a specific application will be divided into sub-teams with different responsibilities (eg, project managers, test managers, design and coding responsible, etc.). These sub-teams should meet and discuss the status of the project from different points of view.

Presentation classes. Each team has to make public presentations (for the rest of the teams) on the status of their project.

All digital communications with students will be made through the mechanisms enabled in the Virtual Campus and MS Teams.

On the use of artificial intelligence (AI) tools. The use of AI tools is permitted for completing project tasks. However, it is essential to keep the following in mind:

- (1) *Responsibility for Content:* All content generated with the help of AI will be considered as if it were entirely produced by the student(s). Therefore, each student must understand, be able to explain, and defend any part of the submitted work, whether it is design, code, documentation, or any other type of deliverable.
 - (2) *Defense and Justification:* If asked about any aspect of the project, students must be able to respond precisely and clearly. Answers such as "I don't know, the AI generated it" will not be accepted.
 - (3) *Active Learning:* The purpose of allowing the use of AI is to complement learning, not to replace it. Using these tools responsibly means reviewing, understanding, and adapting the results they provide.
- In summary, you may use AI, but you must understand and be able to explain any content it generates. The use of AI does not exempt you from responsibility for the submitted content.

Annotation: Within the schedule set by the centre or degree programme, 15 minutes of one class will be reserved for students to evaluate their lecturers and their courses or modules through questionnaires.

Assessment

Continuous Assessment Activities

Title	Weighting	Hours	ECTS	Learning Outcomes
Development of the application (DA1 and DA2)	60% (DA1: 10%, DA2: 50%)	0	0	2, 1, 3, 4, 5, 7, 6, 11, 8, 12, 10, 14, 9, 13, 15
Individual evaluations (I1 and I2)	25% (I1: 10% , I2: 15%)	2	0.08	4, 7, 6, 12, 14, 9, 13
Team presentation (TP)	15%	2	0.08	9

Individual assessments are not considered.

The evaluation is continuous and will be based on the following marks:

1) Development of the application: It should be noted that not only the final result is valued, i.e., the application itself, but also its development process is highly valued. Therefore, the continuous assessment of the work shown in the regular project meetings, especially, in the state of development on pre-established dates will be very important. In particular, there are two partial evaluations referring to the development of the application. The corresponding marks are DA1 and DA2.

2) Team presentations: At the end of the course, the developed application and the methodology followed to do so will be presented to the entire class and to the teaching staff through a "final presentation" that will be evaluated. Here we emphasize that the presentation is evaluated as such. The corresponding mark is TP.

3) Individual evaluations: Individual evaluations will be carried out to find out the degree of actual participation of each student in the application developed by their team. In particular, there are two partial evaluations of this type. The corresponding marks are I1 and I2.

To pass the subject, all these sections must be passed separately. In other words, $DA1 \geq 5$ and $DA2 \geq 5$, $TP \geq 5$, $I1 \geq 5$ and $I2 \geq 5$, where DA1, DA2, PE, I1, and I2 are grades out of 10. If this is true, then the final mark, FM, is calculated as:

$$FM = \min(10.0 ; 0.1 AD1 + 0.5 AD2 + 0.15 TP + 0.1 I1 + 0.15 I2 + Extra)$$

NOTE: In extreme cases in which a student systematically does not do his/her work or has a marginal contribution ($I1 < 5$ or $I2 < 5$), the student will have the subject fail. If $I1 < 5$ but $I2 \geq 5$, the student can have a second chance to pass the subject. In particular, the teaching staff will evaluate with the corresponding team if, although the student started badly ($I1 < 5$), the rest of the course he/she has improved significantly ($I2 \geq 5$). If $I1 \geq 5$ but $I2 < 5$, the student can have a second chance to pass the subject. We will proceed as in the previous case, to discern if, although the student has finished the course worse than he/she had started it ($I2 < 5 \leq I1$), there were circumstances that make it possible for the student to pass the subject deservedly.

NOTE: If $TP \leq 5$ and during the course, the team has made its regular presentations, then there will be a second chance to make the team presentation.

NOTE: If $DA1 < 5$ or $DA2 < 5$, no team member will pass the subject. If $DA1 < 5$ but $DA2 \geq 5$, the teaching staff will assess whether the team deserves a second chance. This second chance would consist of canceling the $DA1 \geq 5$ condition. If $DA2 < 5$, there is not second chance.

If, finally, a student fails to pass the subject, then his/her final mark (FM) will be $\max(0, \min(I1, I2, TP, DA1, DA2))$. It is understood that "Non-Evaluable" students are only those who have not undergone any evaluation activity.

The number of MH (with Honors) that can be awarded is proportional to the number of students enrolled. In order to obtain MH the student must meet $FM > 9.5$ at the time of Chance 1. Now, let's say we can grant "N" MH. If there are more than N candidates, the teaching staff will evaluate the trajectory of the candidates to MH and select the N. Therefore, it is the teacher who evaluates the evidence that he/she considers most appropriate (continued work, relevance within the team, NF, etc.) to make the final decision.

The developed applications are proposed by the students themselves and come out of a selection process based on presentations and voting. Therefore, during this first phase, the transversal competencies T01 and T02 are very important. In fact, the student who proposes an application that passes the whole process (i.e., it becomes an application that will be carried out by a team), is selected as team leader and receives 1 "Extra" point (see FM in Chance 1). Speaking of transversal competencies, note that the whole subject is inherently linked to T03. DA1, DA2, and PE quantify this particular competence. In addition, the team leader can distribute 1 "Extra" point (see FM in Chance 1) among team members with a more meaningful contribution seen from within the team.

It should also be noted that students repeating the course will not receive any special treatment, they must follow the subject as the rest of the students.

Without prejudice to other disciplinary measures deemed appropriate, and in accordance with current academic regulations, will be scored with a zero the irregularities committed by the student that may lead to a variation of the qualification of an evaluation activity. Therefore, plagiarizing, copying or allowing an assessment to be copied or any other evaluation activity, will involve suspending the activity with a zero so that it cannot be recovered in the same academic year. If this activity has a minimum associated score, then the subject will be suspended.

Bibliography

To the bibliography of the rest of the subjects of the Software Engineering branches, it must be added: "Crystal Clear: A Human-Powered Methodology for Small Teams", A. Cockburn.

Software

Each team will look for the SW that suits the best to develop their application.

Groups and Languages

Please note that this information is provisional until 30 November 2025. You can check it through this [link](#). To consult the language you will need to enter the CODE of the subject.

Name	Group	Language	Semester	Turn
(PAUL) Classroom practices	421	Catalan/Spanish	second semester	morning-mixed
(PAUL) Classroom practices	422	Catalan/Spanish	second semester	morning-mixed
(PLAB) Practical laboratories	421	Catalan/Spanish	second semester	morning-mixed
(PLAB) Practical laboratories	422	Catalan/Spanish	second semester	morning-mixed

(PLAB) Practical laboratories	423	Catalan/Spanish	second semester	morning-mixed
(PLAB) Practical laboratories	424	Catalan/Spanish	second semester	morning-mixed
(TE) Theory	420	Catalan/Spanish	second semester	morning-mixed