Universitat Autònoma de Barcelona

2020/2021

Software Integrated Lab

Code: 102788
ECTS Credits: 9

Degree	Type	Year	Semester
2502441 Computer Engineering	OB	3	2
2502441 Computer Engineering	OT	4	2

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: Antonio Manuel López Peña
Email: AntonioManuel.Lopez@uab.cat

Use of Languages

Principal working language: spanish (spa)
Some groups entirely in English: No
Some groups entirely in Catalan: No
Some groups entirely in Spanish: No

Teachers

Jose Luis Gomez Zurita

External teachers

Pablo Torrellas Pérez

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 ones 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.

Methodology

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 team work. 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 mention of 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 communications with students will be made through the mechanisms enabled in the virtual campus.

Activities

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

Assessment

The evaluation will be based on the following scores:

1) Development of the application (DA): it should be noted that not only the final result is valued, i.e. the application itself, but also the development process of the same is valued. Therefore, the continuous evaluation of the work done in the project meetings and, especially, in the state of development in pre-established milestones will be very important. In general, most of this score will be common to the whole team, but mechanisms will be established to ensure that all team members really deserve the score. This will be done from peer evaluations, which will also serve to complete the configuration of the DA total score. If a case is detected in which a student has not done his work or has a marginal contribution, the student involved may receive a lower score; or even more, since we are facing a continuous evaluation, the student can directly suspend the subject (secore of zero). In particular there are two partial tests of this type, DA1 and DA2.

2) Team presentations (PE): in addition to the material required to make the DA evaluation, there will be regular presentations of the work (weekly or biweekly) for all the students enrolled in the course. These presentations will be evaluated as such, and can be understood as a team test, therefore, the score will be for the whole team. Regarding the involvement of the members of the team, the same criterion applies as in the DA case. In particular there are two partial tests of this type, PE1 and PE2.

3) Individual exams (EI): these will be done to evaluate the theoretical-practical knowledge that each student has acquired. In particular, there are two partial tests of this type, EI1 and EI2.

To pass the subject, all these sections must be approved separately. That is, DA> = 5 and PE> = 5 and EI1> = 5 and EI2> = 5, where DA, PE, EI1 and EI2 are notes over 10.If this is the case, then the finalscore (NF) is calculated as:

[Opportunity 1] NF = 0.3 DA1 + 0.5 DA2 + 0.05 PE1 + 0.05 PE2 + 0.05 EI1 + 0.05 EI2 + Extra.

NOTE: If EI1 <5 or EI2 <5, the student may have a second chance to pass the subject through a new individual exam provided certain conditions are met. In particular, if DA = (5/4) * (0.3 DA1 + 0.5 DA2) and PE = 0.5 * (PE1 + PE2), to be able to do this test it is required that DA> = 5 and PE> = 5. Then, if ER is the score of these new exam, the final score of the subject will be calculated as:

[Opportunity 2, second-chance for Individual Exams] NF = min (6, 0.5 DA + 0.5 ER).

NOTE: If DA <5 or PE <5 the student will receive a final score (NF) between 0 and 4.5 (suspended) depending on the case. This student would not have "second chance". Note, that the maximum score that a student can receive in "second chance" is a 6.

It is understood that students "Not evaluable" 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 NF> 9.5 at the time of Opportunity 1. Now, let's say we can grant "N" MH. If there are more than N candidates, the teacher will examine the trajectory of the candidates to MH and select the N. Therefore, it is the teacher who evaluates the evidences 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 competences T01 and T02 are very important. In fact, the student who proposes an application that passes the whole process (that is, it becomes an application that will be carried outby a team), is selected as team leader and receives 1 "Extra" point (see NF in Opportunity 1). Speaking of transversal competences, note that the whole subject is inherently linked to T03. But, DA and PE quantify this particular competence. In addition, the team leader can distribute 1 "Extra" point (see NF in Opportunity 1) among team members with a more meaningful contribution seen from within the team.

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

Assessment Activities

Title	Weighting	Hours	ECTS	Learning Outcomes
Development of the application (DA1 and DA2)	80% (DA1: 30%, DA2: 50%)	0	0	2, 1, 3, 4, 5, 7, 6, 11, 8, 12, 10, 14, 9, 13, 15
Individual examinations (EI1 and EI2)	10% (EI1: 5%, EI2: 5%)	4	0.16	4, 7, 6, 12, 14, 9, 13
Team presentations (PE1 and PE2)	10% (PE1: 5%, PE2:	0	0	9

5%)

Bibliography

To the bibliography of the rest of the subjects of the Software Engineering branches, it must be added:

[1] Practical tips for Software-intensive Student Projects, D.C. Rajapakse.
[2] Agile Software Development: Principles, Patterns and Practices, R.C. Martin.
[3] Agile Software Development: The Cooperative Game, A. Cockburn.
[4] Crystal Clear: A Human-Powered Methodology for Small Teams, A. Cockburn.
[5] Agile Software Development with Scrum, K. Schwaber, M. Beedle.
[6] Agile Project Management with Scrum, K. Schwaber.