

Degree	Type	Year
Computer Engineering	OB	3
Computer Engineering	OT	4

## Contact

Name: Xavier Otazu Porter

Email: xavier.otazu@uab.cat

## Teachers

Olivier Penacchio Vacio

Cristobal Pio Garcia

## Teaching groups languages

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

## Prerequisites

It is necessary to know how to program in Java.

It is not obligatory, but it is recommended to study in parallel the subject "Software Requirements" and "Software Design".

## Objectives and Contextualisation

In this subject we introduce the basic concepts to plan software tests and the techniques to implement them, as well as estimates of their quality.

## Competences

Computer Engineering

- Ability to develop, maintain and evaluate software services and systems that meet all user requirements and behave reliably and efficiently, are affordable to develop and maintain and meet quality standards, applying theories, principles, methods and practices of the Software engineering.
- Acquire personal work habits.

- 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.

## Learning Outcomes

1. Apply formal methods of software design review.
2. Apply formal techniques to the verification of compliance with the requirements of software applications.
3. Critically evaluate the work done.
4. Evaluate software systems ensuring that they meet all user requirements and behave reliably and efficiently, are affordable to develop and maintain and meet quality standards, applying the theories, principles, methods and practices of Software Engineering.
5. Plan and document the software design in practical cases.

## Content

The temporal order of the following contents throughout the course does not necessarily follow this order.

### Introduction to software testing

Objective of the test. Types of tests: white box, black box, validation, regression, loading, etc. Test levels: unity, integration, system, acceptance.

### Unit tests and integration tests

Objective and methodology. Why test for unity and integration? Unit Testing and Test Driven Development. "Mock" objects.

### Black box tests

Equivalence classes. Limit value analysis. Decision tables. Pairwise testing. Transition of states. Design of test cases from use cases. Example ATM.  
Test based on use cases.

### White box tests

Flow control (cyclomatic complexity). Define strings. Coverage test versus flow control test.

### Exploratory test

Simple and fast tests. Practical interest and effectiveness.

### Test automation

Creation of "automatic" tests. Web and recording tools (Selenium).

### System testing or End-to-End testing

Objective and methodology. Why do system tests? Behavior Driven Development (BDD). Review of Test Reports.

### Test Frameworks

Definition and examples. Implementation of a test framework and system tests using Cucumber (BDD).

### Test planning and documentation

Risk analysis. Mastertestplan. Examples of test plan and documentation.

### Quality control, Quality Assurance and Quality Engineering

Formal reviews, metrics, development standards.

Bug definition and bug reports. Bug Tracking Tools: Bugzilla and Jira. How to Integrate a Quality Engineer into a Development Team. Formal technical reviews.

Ecosystem of a Quality Engineer

Definition of the DevOps cycle. Continuous Integration, Continuous Testing, Continuous Delivery, Continuous Feedback. Best-known tools (Jenkins, GitHub / GitLab and change request reviews, Grafana).

## Activities and Methodology

Title	Hours	ECTS	Learning Outcomes
Type: Directed			
Theory sessions	22	0.88	2
Type: Supervised			
Laboratories	12	0.48	2, 1, 3, 4, 5
Seminars	12	0.48	2, 3, 4, 5
Type: Autonomous			
Evaluation tests study	60	2.4	2
Laboratories preparation	24	0.96	2, 4, 5
Seminars preparation	14	0.56	2, 4

This subject has a balance between the theoretical part and the practical part. It is not a totally ABP subject matter since the theoretical concepts are important, and the practical part will be carried out with tutorial sessions, delivery sessions and evaluation of the planetary problems and autonomous work on the part of the student.

**Theory classes:** These are master classes where the theoretical concepts of the subject are introduced. Although theory classes, a good part of the session (approximately half) will be used to solve in a group simple problems raised by the teacher.

**Seminars:** These are classes where practical cases will be considered that will require the application of the concepts explained in the theory classes. At the end of the class of seminars there will be a problem that the student will have to solve autonomously and that he will deliver voluntarily at the beginning of the next session of seminars.

**Laboratories:** These are sessions in which the teacher will conduct tutorials with the students in order to guide their work while developing the solution to practice a real problem of a certain complexity. Before these sessions, the student will have to perform a series of tasks autonomously that will be shown at the beginning of the laboratory session. Teachers will also be able to introduce and work on concepts associated with carrying out the practices, such as explaining computer tools, exercises with these tools, etc.

The transversal competence, T02.08 - Critically evaluate the work carried out, will be evaluated automatically in the practices, since this is a necessary capacity to be able to perform a practice autonomously.

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
Individual continuous assessments Theory 1	0.275	3	0.12	2
Individual continuous assessments Theory 1	0.275	3	0.12	2
Laboratories evaluation	0.45	0	0	2, 1, 3, 4, 5

This subject does not have a single assessment system.

The final grade for the subject is given by the formula:  $\text{Grade} = 0.55 \cdot \text{NTeo} + 0.45 \cdot \text{NLab}$ ,

NTeo is the Theory grade, and NLab is the laboratory grade. Each of the NTeo and NLab grades must be greater than 5, otherwise the final grade for the subject will be the minimum value of these two. The Grade value will only be calculated if NTeo and NLab are both greater than 5. If this is not the case, the Grade value will be the minimum value of NTeo and NLab.

The NTeo grade is

$$\text{NTeo} = 0.6 \cdot \text{NTeo1} + 0.4 \cdot \text{NTeo2}$$

where NTeo1 and NTeo2 are two assessment activities that will be carried out throughout the course during theory session times. The NTeo value will only be calculated if NTeo1 and NTeo2 are both greater than 5. If this is not the case, the NTeo value is considered to be the minimum value of NTeo1 and NTeo2. If NTeo is not greater than 5 after taking the two NTeo1 and NTeo2 tests, the student may re-evaluate NTeo1 and/or NTeo2 during the final assessment tests. In the final assessment tests, each of the NTeo1 and NTeo2 grades will be saturated at a value of 6.

A period will be given at the same time as each assessment is published to review the grade. The grade review periods will be scrupulously respected, that is, no grade will be reviewed outside of this grade review period.

The NLab grade will be a weighted average of the different NLab(i) laboratory sessions.

$$\text{NLab} = 0.75 \cdot \text{NEntr1} + 0.25 \cdot \text{NEntr2}$$

where NEntr(i) is the individual grade for each of the two practical assignments (predictably in laboratory sessions 3 and 6 of the course, respectively). The calculation of these grades is specified in the statement of each practice (which will be published in the Moodle classrooms)

There is no second call for the NLab grade or for any of the NLab(i) grades. The NLab value will only be calculated if each and every one of NLab(i) is equal to or greater than 0.1. Otherwise, NLab will be the minimum value among all NLab(i) values.

The order of assignment to the practical groups will be made according to the grade in the academic record.

NO grades from previous years will be validated.

A student will obtain a "Not Assessable" as the final grade for the subject only if they do not obtain any grade in any of the variables that appear in the previous mathematical formulas. If a student submits any work to be evaluated or carries out any of the evaluation activities that are contemplated in the previous formulas, they will no longer be able to obtain a "Not Assessable".

To obtain an Honors Degree, it is necessary, as a mandatory requirement, to obtain a final grade of Excellent.

The dates of continuous evaluation and submission of work will be published on the virtual campus (CV) or Moodle classrooms (MO) and may be subject to programming changes for reasons of adaptation to possible incidents. The CV/MO will always be informed of these changes since it is understood that the CV/MO is the usual mechanism for exchanging information between professors and students.

Without prejudice to other disciplinary measures deemed appropriate, and in accordance with current academic regulations, irregularities committed by a student that may lead to a variation in the grade will be graded with a zero (0). For example, plagiarizing, copying, allowing copying, ..., an assessment activity, will imply the suspension of this assessment activity with a zero (0). Assessment activities graded in this way and by this procedure will not be recoverable. If it is necessary to pass any of these assessment activities to pass the subject, this subject will be directly failed with a grade of zero (0), with no opportunity to recover it in the same course.

These irregularities include, among others:

- the total or partial copying of a practice, report, or any other assessment activity;
- allowing copying;
- presenting a group work not done in its entirety by the members of the group (applied to all members, not just those who have not worked);
- present as their own materials prepared by a third party, even if they are translations or adaptations, and in general works with non-original and exclusive elements of the student;
- have communication devices (such as mobile phones, smart watches, camera pens, etc.) accessible during individual theoretical-practical evaluation tests (exams);
- talk to classmates during individual theoretical-practical evaluation tests (exams);
- copy or attempt to copy other students during theoretical-practical evaluation tests (exams);
- use or attempt to use writings related to the subject during the theoretical-practical evaluation tests (exams), when these have not been explicitly permitted;
- use Artificial Intelligence tools (chatbots or LLMs like ChatGPT, Perplexity, Claude and similar) to carry out any task.

In this subject, the use of Artificial Intelligence (AI) technologies is not allowed in any of its phases. Any work that includes fragments generated with AI will be considered a lack of academic honesty and may lead to a partial or total penalty in the grade of the activity, or greater sanctions in serious cases.

A student who has committed irregularities in an assessment act will not be validated for any of the assessment activities carried out. In summary: copying, allowing copying or plagiarism (or the attempt to do so) in any of the assessment activities is equivalent to a FAIL, non-compensable and without validation of parts of the subject in subsequent years.

## Bibliography

- [1] A Software Testing Primer: an Introduction to Software Testing. Nick Jenkins.  
<http://www.nickjenkins.net> Lectura obligada!
- [1,2,3,5,6] Manage Software Testing. Peter Farrell-Vinay. Auerbach Publications, 2008.  
Esta bien la parte de documentación y planning, però la de tècniques de prova no tanto.
- [1,3,4] A practitioner's guide to software testing. Lee Copeland, 2004.  
Libro muy sencillo y breve (quizás demasiado), enfoque práctico.
- [4] How to break software: a practical guide to testing. James A. Whittaker. Addison Wesley, 2003.  
EL libro sobre prueba exploratoria.
- [5] Pragmatic Unit Testing in Java with Junit. Andy Hunt, Dave Thomas. The Pragmatic Programmer, 2003.

- [6] IEEE Standard for Software and System Test Documentation IEEE Std82-2008. IEEE Computer Society, 2008.
- [6] Systematic Software Testing by. RickD. Craig, Stefan P.Jaskiel. ArtechHouse, 2002.  
El mejor libro que he encontrado sobre la planificación de pruebas. Incluye una explicación del estándar IEEE de documentación.
- [8] Why the Vasa sank: 10 problems and some antidotes for software projects. Richard E. Fairley, Mary Jane Willshire. IEEE Software, març-abril 2003.  
Lectura interesante y divertida.
- [9] Video "Scenes of software inspections. Video dramatizations for the classroom." L.Deimel. Software Engineering Institute, CMU/SEI-91-EM-5-0, 1991.
- [10] Thinking i Java, 3rd edition. B. Eckel. Versió electrònica a <http://www.planetpdf.com/> . Versió impresa en castellà de Pearson Educación, 2002.  
Llibre de referència per programació en Java (pràctica).
- [11] The Way of the Web Tester: A Begginer's Guide to Automating Tests, J. Rasmusson, 2016
- [12] Software Testing: A Craftman's Approach, 4th edition. Paul C. Jorgensen, 2017
- [13] <https://testautomationu.applitools.com/junit5-tutorial/> Tutorial sobre JUnit.
- [14] <https://testautomationu.applitools.com/unit-testing/index.html> Introducció al Unit Testing.
- [15] <https://site.mockito.org/Paquet> per implementar mock object.
- [16] <https://testautomationu.applitools.com/codeless-test-automation-with-selenium-ide/> Tutorial sobre selenium
- [17] <https://testautomationu.applitools.com/cucumber-java-tutorial/> Tutorial sobre cucumber.

## Software

Eclipse IDE (<https://eclipseide.org/>)

IntelliJ (<https://www.jetbrains.com/idea/>)

JUnit (<https://junit.org/junit5/>)

Selenium (<https://www.selenium.dev/>)

Cucumber (<https://cucumber.io/>)

GitHub/GitLab (<https://github.com/>) (<https://about.gitlab.com/>)

Bugzilla (<https://www.bugzilla.org/>)

Jira (<https://www.atlassian.com/software/jira>)

## 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	first semester	morning-mixed
(PAUL) Classroom practices	422	Catalan	first semester	morning-mixed
(PAUL) Classroom practices	423	Catalan	first semester	morning-mixed

(PLAB) Practical laboratories	421	Catalan/Spanish	first semester	morning-mixed
(PLAB) Practical laboratories	422	Catalan/Spanish	first semester	afternoon
(PLAB) Practical laboratories	423	Catalan/Spanish	first semester	morning-mixed
(PLAB) Practical laboratories	424	Catalan/Spanish	first semester	morning-mixed
(PLAB) Practical laboratories	425	Catalan/Spanish	first semester	morning-mixed
(PLAB) Practical laboratories	426	Catalan/Spanish	first semester	afternoon
(TE) Theory	420	Catalan	first semester	morning-mixed