

<b>BLOCK NAME</b>	SOFTWARE ENGINEERING I
<b>BLOCK CODE</b>	CS-L4B1
<b>COURSE</b>	2
<b>LEVEL</b>	4
<b>CREDITS</b>	5
<b>CLASS HOURS</b>	50
<b>HOMEWORK</b>	75
<b>TOTAL HOURS</b>	125

**DESCRIPTION**

This block introduces the basis of classic software engineering. Students are separated into groups, and each group thinks of a project. Then each group takes requirements from another one and follows the usual software development life cycle to generate all corresponding documents (requirements, design, implementation, test, installation and maintenance). No source code writing is needed for the mandatory part. Optionally, students can write the corresponding source code, verify and then validate it with the customer group.

**PRE-REQUISITES**

Basic programming skills are needed.  
**CS-L1B1**

**OBJECTIVES**

The goal is for students to understand and be able to make a good use of the most common techniques that software engineering provides.

**SKILLS TO BE DEVELOPED**

- 1 - Software Engineering fundamentals.**
  - 1.1 - Understand the fundamentals of software engineering.
- 2 - Software Development Life Cycle (models and phases).**
  - 2.1 - Identify the correct life cycle model for a specific software.
- 3 - Project Management.**
  - 3.1 - Perform project management tasks using tools for the case.
- 4 - Requirements specification.**
  - 4.1 - Be able to gather requirement specifications from meetings with clients.
- 5 - Analysis/Design tools and strategies.**
  - 5.1 - Be able to design a program or system.
- 6 - Implementation.**
  - 6.1 - Be able to implement a program or system following a design.
- 7 - Testing, validation and verification.**
  - 7.1 - Be able to test, validate and verify a program or system.
- 8 - Maintenance.**
  - 8.1 - Be able to maintain a program or system.

**SYLLABUS**

- 1 - Software Engineering fundamentals.
- 2 - Software Development Life Cycle (models and phases).
- 3 - Project Management.
- 4 - Requirements specification.
- 5 - Analysis/Design tools and strategies.
- 6 - Implementation.
- 7 - Testing, validation and verification.
- 8 - Maintenance.

## METHODOLOGY

Resolution of practical activities supervised by the mentor. Compulsory attendance.

## DEDICATION AND EVALUATION

The student must pass the mandatory activities (challenges/projects) that are covered in the block.

Each challenge/project produces its own score and has been designed to cover certain block percentages.

Such score is 80% objective (the program that solves the challenge/project works without errors and producing the expected results) and 20% subjective (solution elegance, how clean the code is, documentation).

Block scores are finally calculated by prorating individual activities with respect to their block coverage percentages.