# COURSE DATA

| Data Subject | |
|---|---|
| **Code** | 34662 |
| **Name** | Software Engineering II |
| **Cycle** | Grade |
| **ECTS Credits** | 6.0 |
| **Academic year** | 2020 - 2021 |

**Study (s)**

| Degree | Center | Acad. year | Period |
|---|---|---|---|
| 1400 - Degree in Computer Engineering | School of Engineering | 3 | First term |

**Subject-matter**

| Degree | Subject-matter | Character |
|---|---|---|
| 1400 - Degree in Computer Engineering | 7 - Software engineering and project management | Obligatory |

**Coordination**

| Name | Department |
|---|---|
| MARTINEZ DURA, JUAN JOSE | 240 - Computer Science |

# SUMMARY

The Software Engineering II course is part of the subject Software Engineering and Project Management. The overall objective of this course is to broaden the content expanded in the course Software Engineering I framing them within the methodology METRIC version 3 and entering the maintenance processes and reengineering of information systems, and describing a range of activities of organizational or support the development process and products that exist in the case of the organization should be applied to enrich or influence the execution of the activities of the main processes of software development methodology and if not there will have to make to complement and ensure the success of the project developed (configuration management software, management and quality assurance, safety management).

Overall, the objectives of the course are:

- To show students the overview of the process of developing information systems, explaining the characteristics of the processes of maintenance and reengineering of information systems.

- Present the activities of organizational or support related to software development process to be used to enrich or influence the execution of the activities of the main processes of the development methodology.

- To publicize the need to maintain the integrity of the products obtained during the development of systems, ensuring that no changes are made uncontrolled and all participants have the appropriate version of the products they handle.

- Provide students with a common frame of reference for the definition and implementation of specific plans for quality assurance applicable to specific projects.

- Raise awareness among students of the need to incorporate information systems security mechanisms that ensure proper development of any type of system along the processes that are carried out to obtain it.

From the educational point of view, the subject has a practical approach and is primarily focused on the development of practical skills for the engineer who must use their professional development as a project manager, or as part of the project team. To do this you must purchase a set of skills related to management of both material and human resources, and task decomposition.

## PREVIOUS KNOWLEDGE

### Relationship to other subjects of the same degree

There are no specified enrollment restrictions with other subjects of the curriculum.

### Other requirements

As the name suggests the course Software Engineering II has a very direct connection with what is taught in Software Engineering I, being necessary to have previously studied this subject. More specifically, the course is based on the following concepts will be assumed known:
Specification and modeling of software requirements
System analysis (modeling use cases, classes, etc.)
System design (modeling class diagram, interaction diagram, state diagram, etc).
Strategies and techniques for system t

## COMPETENCES (RD 1393/2007) // LEARNING OUTCOMES (RD 822/2021)

### 1400 - Degree in Computer Engineering

- G2 - Ability to lead project activities in the field of information technology, in accordance with both the knowledge and the specific skills acquired in the degree.
- G3 - Ability to design, develop, evaluate and ensure the accessibility, ergonomics, usability and security of computer systems, services and applications, and of the information that these manage.

- G5 - Ability to design, develop and maintain computer systems, services and applications using software engineering methods as an instrument for quality assurance, in accordance with both the knowledge and the specific skills acquired in the degree.
- G9 - Ability to solve problems with initiative, decision making, autonomy and creativity. Ability to communicate and transmit the knowledge, skills and abilities of a computer engineer.
- G10 - Knowledge to perform measurements, calculations, assessments, appraisals, surveys, studies, reports, scheduling and other similar work in the field of computer engineering, in accordance with both the knowledge and the specific skills acquired in the degree.
- G12 - Knowledge and application of the basic principles of economics and human resource management, project organisation and planning, and legislation, regulation and standardisation in the field of computer projects, in accordance with both the knowledge and the specific skills acquired in the degree.
- R1 - Ability to design, develop, select and evaluate computer applications and systems while ensuring their reliability, safety and quality, according to ethical principles and current legislation and regulations.
- R2 - Ability to plan, design, implement and manage computer projects, services and systems in all areas, leading their implementation and continuous improvement by assessing their economic and social impact.
- R3 - Ability to understand the importance of negotiation, effective work habits, leadership and communication skills in all software development environments.
- R16 - Knowledge and application of the principles, methodologies and life cycles of software engineering.
- R18 - Knowledge of the rules and regulations of computer science at the national, European and international levels.
- TI2 - Ability to select, design, implement, integrate, evaluate, build, manage, exploit and maintain hardware, software and network technologies, within adequate cost and quality thresholds.
- SI3 - Ability to actively participate in the specification, design, implementation and maintenance of information and communication systems.

## LEARNING OUTCOMES (RD 1393/2007) // NO CONTENT (RD 822/2021)

Learning outcomes of the "Software Engineering II" course are:

- Apply methods for developing, implementing and maintaining information systems.

- Apply techniques for evaluation and quality assurance software

- Define validation testing and verification requirements

- Implement procedures for configuration management

- Understand the improvements provided business management and project-based production compared to a continuous type methodology

- Know, differentiate and drafting the various documents that are often managed for the management and life of a project

- Analyze the basic concepts of project management

- Develop basic skills (techniques and tools) in planning and implementing projects

## DESCRIPTION OF CONTENTS

### 1. Software methodologies: METRICA version 3 vs SCRUM

1.1. Introduction to METRICA Version 3
1.2. Participants
1.3. Main processes
1.4. Interfaces Software Development
1.5. Systems Development Phase
1.6. Introduction to SCRUM methodology

### 2. Agile Methodologies: User Stories

2.1. Introduction to SCRUM
2.2. User stories
2.2.1. Description of a user story: Examples
2.2.2. Estimation and assessment of a story
2.2.3. Prioritisation of stories
2.2.4. Validation and quality criteria
2.2.5. Technical stories
2.2.6. Exercises

### 3. Testing

3.1. Introduction
3.2. Testing Techniques
3.3. Software Testing Strategies
3.4. Software Debugging
3.5. Unit Testing with JUnit
3.6. Exercises and activities

## 4. Software configuration management

4.1. introduction
4.2. Activities of SW configuration management
4.3. Techniques and tools for version control
4.4. SUBVERSION
4.5. Using SUBVERSION
4.6. Exercises and activities

## 5. Measurement

5.1. Measurement software
5.2. Product Measures
5.3. Process measures and resources
5.4. Methodologies and standards for measuring
5.5. Empirical studies
5.6. Metrics during software development
5.7. Exercises and activities

## 6. Maintenance

6.1. Software Maintenance
6.2. Software Maintenance Problems
6.3. Types of maintenance
6.4. Standards for maintenance
6.5. The maintenance process
6.6. Maintenance solutions

## 7. Software Quality

7.1. Introduction
7.2. Product Quality
7.3. Process quality
7.3.1. Quality Assurance
7.3.2. The CMMI
7.3.3. SPICE Model: ISO / IEC 15504
7.3.4. ISO 9000
7.3.5. Other models
7.4. Exercises and activities

## WORKLOAD

| ACTIVITY | Hours | % To be attended |
|---|---|---|
| Theory classes | 30,00 | 100 |
| Laboratory practices | 20,00 | 100 |
| Classroom practices | 10,00 | 100 |
| Development of group work | 20,00 | 0 |
| Development of individual work | 10,00 | 0 |
| Study and independent work | 15,00 | 0 |
| Readings supplementary material | 5,00 | 0 |
| Preparation of evaluation activities | 15,00 | 0 |
| Preparing lectures | 10,00 | 0 |
| Preparation of practical classes and problem | 15,00 | 0 |
| **TOTAL** | **150,00** | |

## TEACHING METHODOLOGY

The development of the course is structured around the following areas: theoretical lessons, problem sessions and practical exercises, laboratory sessions and activities and practical work to be performed by the student individually and in small groups in order to translate the knowledge acquired during the lessons of theory and problems.

•Theoretical lessons. In the lectures will develop the fundamental contents of the subject providing a global and inclusive vision, analyzing in detail the key issues and more complex. This media is used (such as presentations, transparency, chalkboard, etc.) at all times encouraging student participation.

•Practical exercises and problems. The lectures will be complemented by practical activities and problem solving in the classroom in order to verify and reinforce concepts presented earlier. He planned several types of activities and problems:

- Individual Activities: What will be done individually by students in order to assess individual understanding of matter. Encourage the involvement of students who will be responsible for his explanation and correction on the board and will be responsible to meet the concerns with the other partners.

- Activities in small groups: The problems will be solved by groups of students together to encourage teamwork and collective analysis of different approaches. These activities will be corrected by the teacher.

•Laboratory sessions. Consist in resolving problems related to the theoretical by software so that the student knows a practical way of implementing CASE tools development methodologies and software maintenance. You are laboratory sessions will be organized around working groups of a maximum of two people.

•Individual student. Consist of performing work, issues and problems posed by the teacher to outside the classroom and class preparation in advance, reading recommended texts and exam preparation (study). This will be done individually and try to promote self-employment. Some of these activities shall be voluntary and additional training of the subject.

•Work in small groups. At the beginning of the course is set up small groups of 3 or 4 students who will remain fixed throughout the course and will have to be led and coordinated by a student. These groups will have to perform work or solve problems outside the classroom, to supplement individual work, building the capacity of integration into working groups. The nature of this work will be diverse, but include the submission of a report of work performed, the exposure of such work and class participation as reviewers of the papers presented by other groups.

•Tutoring The students have a schedule of tutorials aimed at solving the problems, doubts, work orientation, etc. The schedule of these tutorials will be indicated at the beginning of the academic year. They will also have the opportunity to clarify some questions via email or discussion forums by using the tool "Virtual Classroom", which gives the University of Valencia.

## EVALUATION

Knowledge assessment will be done in two ways:

**1) CONTINUOUS EVALUATION**

Recommended method for students. The following factors are evaluated to obtain the final mark:

- 55% theoretical knowledge and problems (TEO).
- 25% of the laboratory (LAB)
- 20% of additional work (ART)

To be able to average it is necessary to obtain a minimum grade of 4,5 in each one of the parts, being necessary that the final note is equal to or superior to 5 to pass.

**a) Theoretical knowledge and problems (TEO).**

The note of theoretical knowledge and problems are assessed according to the following factors:

- 80% OF INDIVIDUAL TESTS OBJECTIVES. During the course there will be different written tests on theoretical knowledge and problems. It will be necessary to get a grade of 4 or higher in each test so that you can compensate. In the final examination of the first call, those parts that have not been passed in the partial tests will have to be recovered.

- 20% OF PROBLEMS. We will evaluate the different problems that are proposed to the students, either to perform in class or at home. This activity is not recoverable.

**b) Laboratory (LAB).**

The laboratory grade will be obtained by averaging the grade obtained in the N practical sessions. In order to obtain the grade of the laboratory it will be necessary to have presented all the practices and to have attended a minimum 80% of the classes.

**c) Additional works (TRA).**

The note of additional works will be obtained by averaging the grades obtained in each of the works by the weight assigned to each one. It will be necessary to obtain a mark of 4 or more in each work so that this part can be compensated.

The note of each work will be obtained in function of the written memory, and optionally it will be possible to value the public exhibition of the work done.

**2) SINGLE EVALUATION SYSTEM AND SECOND CALL**

This method will apply to any student who, for a reason reasoned and admitted by the teacher, can not attend regularly to classes and in the second call.

The following factors are evaluated to obtain the final mark:

- 55% theoretical knowledge and problems (TEO).
- 25% of the laboratory (LAB)
- 20% of additional work (ART)

To be able to average it is necessary to obtain a minimum grade of 4,5 in each one of the parts, being necessary that the final mark is equal to or superior to 5 to pass.

**a) Theoretical knowledge and problems (TEO).**

The note of theoretical knowledge and problems are assessed by a single examination, not taking into account other factors such as attendance or problem exercises performed during the course.

**b) Laboratory (LAB).**

The laboratory grade will be obtained by averaging the grade obtained in the practical sessions, which must have been delivered, even if you have not attended the laboratory sessions.

**c) Additional works (TRA).**

The note of additional works will be obtained by averaging the grades obtained in each of the works by the weight assigned to each one. It will be necessary to obtain a mark of 4,5 or more in each work so that this part can be compensated.

The student must have submitted all the papers to pass, and only the part of the written report will be evaluated.

In any case, the evaluation of this subject will be done in compliance with the University Regulations in this regard, approved by the Governing Council on 30th May 2017 (ACGUV 108/2017)

## REFERENCES

### Basic

- Project Management Institute, "A Guide to the Project Management Body of Knowledge", 4th edition, Project Management Institute (2008), ISBN: 19-33890517
- Sanchez, S; Sicilia, M.A; Rodriguez, D. Ingeniería del Software un enfoque desde la guía SWEBOK. Gaceta grupo editorial; ISBN: 978-8492812400
- Domingo Ajenjo, A. Dirección y Gestión de Proyectos, un enfoque práctico. Editorial Rama, (2005). ISBN: 9701511301.
- Martín, G; Dawson, C. El proyecto fin de carrera en ingeniería informática. Editorial Prentice Hall; ISBN: 84-20535605.

### Additional

- Pereña, J. "Dirección y Gestión de Proyectos". Editorial Díaz de Santos (1991). ISBN: 8479782498
- Grashina M.N; Newell M.W, Preguntas y Respuestas Sobre La Gestión de Proyectos, Editorial Gestión 2000, (2005). ISBN: 9788480886864
- Gómez, J. F; Coronel, A.J; Martinez de Irujo, L; Lorente, A. "Gestión de proyectos". FC Editorial. Madrid, (2000). ISBN: 84-28317747.

## ADDENDUM COVID-19

**This addendum will only be activated if the health situation requires so and with the prior agreement of the Governing Council**

The teaching method for this subject will follow the teaching model approved by the Academic Committee of the GII / GIM degrees:
https://go.uv.es/catinfmult/ModeloDocenciaGIIGIM

In the event of a closure of the facilities due to the health situation, and if this affects all or part of the classes of the course, these will be replaced by voice and self-explanatory presentations of an asynchronous nature. An on-line consultation support will be offered at the established times, and a forum will be opened to solve theory questions and exercises.

In the event that the facilities are closed due to the health situation, and if this should affect any of the course's classroom tests, these will be replaced by tests of a similar nature but in virtual mode through the computer tools supported by the University of Valencia. The evaluation percentages will remain the same as those established in the guide