**ANEXO I**

Escuela Técnica Superior de Ingeniería

GENERAL SPECIFICATIONS

# COURSE 24/25

## Subject Data

| **Name:** |
|---|
| Ingeniería de Requisitos |
| **English name:** |
| Requirements Engineering |

| **Code:** | **Type:** |
|---|---|
| 606010218 | Compulsory |

| **Hours:** | | | |
|---|---|---|---|
| | **Total** | **In class** | **Out class** |
| **Time distribution** | 150 | 60 | 90 |

| **ECTS:** | | | | |
|---|---|---|---|---|
| **Standard group** | **Small groups** | | | |
| | **Classroom** | **Lab** | **Practices** | **Computer classroom** |
| | 40 | 20 | | |

| **Departments:** | **Knowledge areas:** |
|---|---|
| Tecnologías de la Información | Lenguajes y Sistemas Informáticos |
| | |
| **Year:** | **Semester** |
| 3 | 2 |

**ANEXO I**

## TEACHING STAFF

| Name: | E-mail: | Telephone |
|---|---|---|
| Victoria Pachón Álvarez | vpachon@uhu.es | +34959217373 |
| | | |
| | | |

**Others Data (Tutoring, schedule... )**

See: www.uhu.es/etsi

**ANEXO I**

# SPECIFIC INFORMATION OF THE COURSE

## 1. Contents description:

### 1.1 In English:

The objectives of this course are:

– To offer an overview of the importance of Requirements Engineering and of the software development process.

– To learn the procedures, techniques, products, and tools for Requirement Engineering to demonstrate the procedures

### 1.2 In Spanish:

- Fundamentos de la Ingeniería de Requisitos.
- El Proceso de la Ingeniería de Requisitos. Procesos, métodos y herramientas.
- Clasificación de tipos de requisitos: funcionales, no funcionales, de información y atributos de calidad.
- Estudios de viabilidad
- Técnicas de identificación de requisitos
- Análisis y negociación de requisitos
- Validación y verificación de requisitos
- Gestión de requisitos.
- Técnicas de especificación de requisitos, documento de especificación de requisitos y control del cambio.
- Introducción a la calidad en la ingeniería de requisitos.

## 2. Background:

### 2.1 Situation within the Degree:

The contents developed within this subject are directly related to those developed in Principles and Fundamentals of Software Engineering (2nd Year) and it is an essential subject within the Software Engineering itinerary of the Degree in Computer Engineering.

### 2.2 Recommendations

## 3. Objectives (as result of teaching):

The main objective of this course is to present the fundamentals of Requirements Engineering, its processes, methods, and tools. In general, the objectives encompass the following concepts:

- Requirements Engineering Process.
- Processes, methods, and tools.
- Classification of Software Requirements
- Feasibility Studies, Requirements Identification Techniques, Requirements Analysis and Requirements Negotiation
- Introduction to Quality in Software Requirements Engineering

## 4. Skills to be acquired

### 4.1 Specific Skills:

CE1-IS: Ability to develop, maintain and evaluate services and software systems that meet all the 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.
CE2-IS: Ability to assess customer needs and specify the software requirements to satisfy these needs, reconciling conflicting objectives by seeking acceptable compromises within the limitations derived from cost, time, the existence of already developed systems and the organizations.

### 4.2 General, Basic or Transversal Skills:

CB3: Ability to collect and interpret relevant data (usually within their area of study) to make judgments that include a reflection on relevant issues of a social, scientific or ethical nature.
CG0: Capacity for analysis and synthesis: Find, analyze, criticize (critical reasoning), relate, structure and synthesize information from various sources, as well as integrate ideas and knowledge.
G01: Capacity for organization and planning as well as information management capacity.
G04: Ability to make decisions based on objective criteria (experimental, scientific or simulation data available) as well as the ability to argue and logically justify such decisions, knowing how to accept other points of view.
G08: Ability to adapt to technologies and future environments, updating professional skills.
CT2: Develop critical thinking in relation to the capacity for analysis and synthesis.
CT3: Develop an attitude of inquiry that allows the revision and permanent advancement of knowledge.
CT4: Ability to use Computer and Information Competences (CI2) in professional practice.

## 5. Training Activities and Teaching Methods

### 5.1 Training Activities:

**ANEXO I**

- Lecture
- Problem Solving Sessions
- Practical sessions in specialized laboratories
- Evaluation activities and self-evaluation and other activities (Essay, debates, tasks delivery, conferences…)

5.2 Teaching Methods:

- Participatory magisterial class.
- Development of practices in specialized laboratories or computer classrooms in small groups.
- Problem solving and practical exercises.
- Evaluations and exams.

5.3 Development and Justification:

During each **theory session**, the instructor will present each topic through **interactive lectures**. Additionally, to enhance the learning process, these lectures will alternate with **problem-solving sessions**.

During the **practical sessions** (carried out in specialized laboratories), students will be given assignments to complete in class. These sessions will focus on conducting the **practical work** and addressing any questions. The assignments and materials are available on the course website; however, the use of additional books, resources, and sources of knowledge is recommended. The practicals will be assessed through **an individual practical test**.

Throughout the course, students will need to **work in groups (exceptionally it can be individuals)** on a project related to the subject matter and present it in class. Furthermore, ongoing assessment tests **(evaluation activities**) will be conducted.

The course features a dedicated website where students can find what they need to prepare for each class, along with the required documentation for each session. All available technological resources in the classroom, such as projectors and Wi-Fi, will be utilized. Students are welcome to bring their own materials to class, such as books, laptops, etc.

**6. Detailed Contents**

**ANEXO I**

**Chapter 1. Introduction to requirements engineering**

- Overview of Requirements Engineering
- Basic Concepts: Definition and Terminology
- Basic Process of Requirements Engineering
- Types of Requirements
- Requirements Engineering Process
- General Recommendations for Documenting Non-Functional Requirements
- Quality: Desirable Properties of Requirements
- Identifying Participants: Stakeholder Focus.
- Case Studies and Real-World Examples

**Chapter 2. Requirements elicitation**

- Introduction to Requirements Elicitation
- Understanding Elicitation
- Techniques to Study the Domain of the Problem
- Challenges in Elicitation
- Elicitation Problems
- Cultural Challenges in Requirements Elicitation
- Outcomes and Lessons Learned
- Elicitation Best Practices
- Preparation for Meetings
- Running Successful Meetings
- Documentation & Validation
- Tools for Elicitation
- Case Studies and Real-World Examples

**Chapter 3: Documenting Requirements Using Use Cases**

- Introduction
- Actors
- Use Case Diagrams.
- Actors
- Guide to writing use cases. Templates and best practices.
- Case Studies and Real-World Examples

**Chapter 4: Requirements Analysis. Class and Object Diagram**

- Understanding Requirements Analysis
- Tasks for Analyzing Requirements
- UML Overview
- Static Model of a Software System
- Class Diagram Details
- Object Diagrams
- Case Studies and Real-World Examples

**Chapter 5: Requirements Analysis and State Diagrams**

- Behavior Modeling
- Key Concepts in State Diagrams. Applications.
- Detailed Examination of State Diagram Components
- Constructing State Diagrams. Best practices.
- How to make a state diagram from a selection of requirements.
- Advanced Topics
- Case Studies and Real-World Examples

**Chapter 6: Requirements Analysis. Sequence and Activity Diagrams**

- Introduction to Activity Diagrams
- Constructing an Activity Diagram
- Advanced Activity Diagram Techniques
- Introduction to Sequence Diagrams
- Components of Sequence Diagrams

- Building a Sequence Diagram
- Advanced Concepts in Sequence Diagrams
- Case Studies and Real-World Examples

**Chapter 7: Requirements Management and Verification**

- Validation vs Verificacion
- Managing nonconformities
- Requirements Quality
- Requirements Traceability
- Characteristics of High-Quality Use Cases
- Techniques for Reviewing Requirements Documentation
- Managing Conflicts in Requirements. Standard Process for Conflict Negotiation in Requirements Management
- Understanding Requirement Changes. Strategies for Managing Requirement Changes
- Best Practices for Change Management
- The Role of Baselines in Change Managemen**t**

## 7. Bibliography

### 7.1 Basic Bibliography:


- Cáceres García de Marina, P., Garrido Blázquez, M. Á., & Sierra Alonso, A. (2019). Especificando software mediante casos de uso y UML : ejercicios resueltos. Editorial Universitaria Ramón Areces. https://columbus.uhu.es/permalink/34CBUA_UHU/1jebu06/alma991008720980504993
- http://www.juntadeandalucia.es/servicios/madeja/contenido
- Klaus Pohl and Chris Rupp. Requierements Engineering Fundamentals. Segunda Edición.2015
- Roger Pressman. Ingeniería del Software. Séptima Edición. 2010
- Software Requirements. Third Edition. Karl Weigers and Joy Beatty. 2013
- B. Silver. BPMN Method & Style (2nd edition).
- M. Cohn. User Stories Applied for Agile Software Development. Addison-Wesley, 2004.
- C. Larman. UML y Patrones. Ed. Prentice-Hall, 1999

### 7.2 Additional Bibliography:


- Carlos Fontela. UML Modelado de Software para Profesionales. Primera Edición. 2012
- S. Lauesen. Software Requirements: Styles and Techniques. Addison-Wesley, 2002.
- Wohlin et al. Experimentation in Software Engineering: An Introduction. Kluwer Academic Publishers, 2000.
- C. Larman. UML y Patrones (2ª edición). Ed. Prentice-Hall, 2003.
- M. Fowler. UML Distilled (3rd edition). Ed. Addison-Wesley, 2004.
- Ivar Jacobson. Object-oriented software engineering: A use case driven approach. Addison-Wesley, 1992.
- Alistair Cockburn. Writing effective use cases. Addison-Wesley, 2001
- G. Kontoya e I. Sommerville. Requirements Engineering: Processes and Techniques. John Wiley & Sons, 1997.
- I. Sommerville y P. Sawyer. Requirements Engineering: A Good Practice Guide. John Wiley & Sons, 1997.C.


## . Systems and Assessment Criteria

### 8.1 System for Assessment:

- Examination of theory / problems
- Defense /Examen of practice
- Individual monitoring (partial tests and activities)
- Work Defense

**ANEXO I**

## 8.2 Assessment Criteria and Marks:

### 8.2.1 Examinations Convocatory I

The overall grade for the course will be calculated using the following formula:

Final grade = 0.4 Theory grade + 0.4 Practice grade + 0.1 Individual Monitoring + 0.1 Work Defense

The theory exam will be multiple-choice. During the practical defense, the student will solve problems related to the development of their work in the practical sessions, and the documentation developed during these sessions will be assessed.

The student must obtain more than 3 points in the grades corresponding to the theory and practical components. Regarding the theory and practical grades, for the purpose of recording grades from Convocatory I to Convocatory II, it will be considered passed when the grade is above 5 points out of 10. Passed components will be carried over from Convocatory I to Convocatory II if the student wishes to do so, utilizing the mechanisms provided for this purpose. If no notification is made, it will be assumed that the student wishes to carry them over. There is no minimum requirement for the grade of Individual Monitoring or for the Work Defense. Both can only be obtained during the dates indicated at the beginning of the course and are automatically carried over from Convocatory I to Convocatory II.

### 8.2.2 Examinations Convocatory II

The overall grade for the course will be calculated using the following formula:

Final grade = 0.4 Theory grade + 0.4 Practice grade + 0.1 Individual Monitoring + 0.1 Work Defense

The theory exam will be multiple choice. During the practices defense the student will solve problems related to the development of their work in the practice sessions and the documentation developed during them. Individual Monitoring and Work defense grades are passed from Convocatory I.

Student must obtain more than 3 points in the grade corresponding to the theory mark and to the one corresponding to the practice grade.

### 8.2.3 Examinations Convocatory III

Same as Single Final Evaluation

### 8.2.4 Extraordinary Convocatory

Same as Single Final Evaluation

### 8.3 Single Final Evaluation:

The default evaluation method will be continuous assessment. Students who want to benefit from the final single evaluation must communicate it in the first two weeks of the subject, or in the two weeks following enrolment if it has occurred after the beginning of the subject. To do this, a link will be enabled on the e-learning platform.

Final mark = 0.6 * Theory exam + 0.4 * Practical defense

In this case, both the theoretical exam and the resolution of the practical problem will be held on the day set by the University. The theory exam will consist of solving problems and theoretical-practical questions related to the theory syllabus. To carry out this exam, it will not be possible to use additional material to that indicated. The practice defense will consist of the preparation of a requirements specification and analysis documentation and the resolution of a problem related to the document developed on the day set by the University for the theoretical exam. To pass the subject, the student must obtain at least 5 points out of 10 in the theoretical exam and 5 points out of 10 in the practical defense.