UNIVERSIDAD DE CASTILLA - LA MANCHA
GUÍA DOCENTE

1. General information

Course: SOFTWARE ENGINEERING II	**Code:** 42324
Type: CORE COURSE	**ECTS credits:** 6
Degree: 347 - DEGREE PROGRAMME IN COMPUTER SCIENCE ENGINEERING (CR)	**Academic year:** 2022-23
Center: 108 - SCHOOL OF COMPUTER SCIENCE OF C. REAL	**Group(s):** 20 21 22
Year: 3	**Duration:** First semester
Main language: English	**Second language:** Spanish
Use of additional languages: Use of English as the main language for teaching in the bilingual groups, and Spanish in the rest.	**English Friendly:** N
Web site: https://campusvirtual.uclm.es	**Bilingual:** Y

Lecturer: ISMAEL CABALLERO MUÑOZ-REJA - Group(s): **20 21 22**

Building/Office	Department	Phone number	Email	Office hours
Fermín Caballero/2.16	TECNOLOGÍAS Y SISTEMAS DE INFORMACIÓN	926052259	Ismael.Caballero@uclm.es	Available at https://esi.uclm.es/index.php/grado-en-ingenieria-informatica/profesorado/

Lecturer: FERNANDO GUALO CEJUDO - Group(s): **20 22**

Building/Office	Department	Phone number	Email	Office hours
	TECNOLOGÍAS Y SISTEMAS DE INFORMACIÓN		Fernando.Gualo@uclm.es	

Lecturer: MOISES RODRIGUEZ MONJE - Group(s): **21 22**

Building/Office	Department	Phone number	Email	Office hours
Fermín Caballero/2.19	TECNOLOGÍAS Y SISTEMAS DE INFORMACIÓN	926052676	Moises.Rodriguez@uclm.es	Available at https://esi.uclm.es/index.php/grado-en-ingenieria-informatica/profesorado/

2. Pre-Requisites

Students are required to have already passed Programming Fundamentals I and II, as well as Software Engineering I and Databases to have a minimum guarantee of passing this course. Those students who have not already passed these courses and are interesting in getting enrolled in Software Engineering II, should make a major effort to acquire the necessary knowledge and experience of software design and databases.

3. Justification in the curriculum, relation to other subjects and to the profession

This subject pretends to offer a big picture of the Software Engineering processes and their relations to create work teams that develop information systems industrializing such processes, highlighting that this development is always cooperative and multidisciplinary.

For this aim, the subject introduces the concept of Software Lifecycle and describes a map of the processes necessary to develop software with a work team, explaining its enactment when different software development methodologies are used.

In addition, some important development-supporting processes such as configuration management, quality management, test management and maintenance management will be also explained. All these processes will be grounded in open ISO standards, which will provide the student with a larger global vision, and probably, they will open some doors to interesting positions in any organization.

As result, it is expected that the student will be able to achieve the abilities and knowledge necessary to work as Software Engineer.

Moreover, it is important to highlight that this subject belong to the module of Software Engineering, Information Systems and Intelligent Systems of the curriculum and provides the basis for the subjects of the specialty of Software Engineering:

- Requirements Engineering
- Software Design
- Process of Software Engineering
- Quality of Software Engineering
- Software Project Management
- Databases Development
- Enterprise Information Systems
- Security of Software Systems

4. Degree competences achieved in this course

Course competences

Code	Description
CO01	Ability to design, develop, select, and assess, applications and digital systems, guaranteeing their reliability, security, and quality, according to ethical principles and the current and common laws.
CO02	Ability to conceive, plan, develop and manage projects, services, and digital systems in any context, leading their start and applying continuous improvements, assessing their economic and social impact.
CO03	Ability to understand the important of negotiation, work efficiency, leadership, and communication abilities in every context of software development.

| CO05 CO08 | Knowledge, administration and maintenance of systems, services and digital systems.
Ability to analyse, design, build and maintain applications in a strong, safe, and efficient manner by selecting the most appropriate paradigms and programming languages. |
|---|---|
| CO16 | Knowledge and application of principles, methodologies, and life spans of software engineering. |
| PER01 | Team work abilities. |
| SIS04 | Adaptation to new scenarios. |

5. Objectives or Learning Outcomes

Course learning outcomes

Description

Knowledge and application of different types of software life cycle models.

Knowledge about tools that support the construction of software systems and the storage and processing of data.

Knowledge and use of the technologies that support the construction and use of information systems.

Implementation and maintenance of applications according to the analysis and design activities previously carried out.

Consideration of the aspects of quality in software development such as usability, accessibility, security, reliability, etc.

Additional outcomes

To achieve professional preparation for the insertion of computer scientists in teams oriented to the development and operation of computer applications

Understand the need for software version control and change requests as a self-protection mechanism in the software development process and apply these principles correctly.

Select and apply the appropriate testing techniques according to the characteristics of the product and the development process.

Plan and properly document the testing process according to the specific needs of the project.

Have a general vision of the quality in software development as part of the success of the project.

Have a general vision and knowledge of the international standards related to the quality of a software product.

Have a general vision and knowledge of international standards related to the maturity of software processes.

Know and apply appropriately the existing techniques for software maintenance.

6. Units / Contents

Unit 1: Methodologies and Software Development Processes

Unit 2: Software Configuration Management

Unit 3: Quality of Software Products

Unit 4: Verification and Validation of the Software

Unit 5: Software Maintenance

Unit 6: Lab sessions

ADDITIONAL COMMENTS, REMARKS

Description of the Lab Sessions:

- P1. Use of the Unified Development Process to a given software development project, software configuration management, testing and maintenance

7. Activities, Units/Modules and Methodology

Training Activity	Methodology	Related Competences (only degrees before RD 822/2021)	ECTS	Hours	As	Com	Description
Class Attendance (theory) [ON-SITE]	Lectures	CO01 CO02 CO03 CO05 CO08 CO16	0.72	18	N	-	Teaching of the subject matter by lecturer (MAG)
Individual tutoring sessions [ON-SITE]		CO01 CO02 CO03 CO05 CO08	0.18	4.5	N	-	Individual or small group tutoring in lecturer's office, classroom or laboratory (TUT)
Problem solving and/or case studies [ON-SITE]	Problem solving and exercises	CO01 CO02 CO05 CO08 CO16	0.6	15	Y	N	Worked example problems and cases resolution by the lecturer and the students (PRO)
Study and Exam Preparation [OFF-SITE]	Self-study	CO01 CO02 CO03 CO05 CO08 CO16	2.1	52.5	N	-	Self-study (EST)
Other off-site activity [OFF-SITE]	Practical or hands-on activities	CO01 CO02 CO03 CO05 CO08 CO16 PER01 SIS04	0.6	15	N	-	Lab practical preparation (PLAB)
Writing of reports or projects [OFF-SITE]	Self-study	CO01 CO02 CO05 CO08 PER01	0.9	22.5	Y	N	Preparation of essays on topics proposed by lecturer (RES)
Laboratory practice or sessions [ON-SITE]	Practical or hands-on activities	CO01 CO02 CO03 CO05 CO08 CO16 PER01	0.6	15	Y	Y	Realization of practicals in laboratory /computing room (LAB)
Final test [ON-SITE]	Assessment tests	CO02 CO03 CO08 CO16	0.3	7.5	Y	Y	Final test of the complete syllabus of the subject (EVA)
		Total:	6	150			
		Total credits of in-class work: 2.4					Total class time hours: 60
		Total credits of out of class work: 3.6					Total hours of out of class work: 90

As: Assessable training activity

Com: Training activity of compulsory overcoming (It will be essential to overcome both continuous and non-continuous assessment).

8. Evaluation criteria and Grading System

Evaluation System	Continuous assessment	Non-continuous evaluation*	Description

Laboratory sessions	25.00%	25.00%	Compulsory activity that can be retaken. To be carried out during lab sessions. The students of non-continuous modality will be evaluated of this activity through an alternative system in the ordinary call
Final test	50.00%	50.00%	Compulsory activity that can be retaken (rescheduling) to be carried out within the planned exam dates of the final exam call (convocatoria ordinaria).
Theoretical papers assessment	15.00%	15.00%	Non-compulsory activity that can be retaken. To be carried out before end of teaching period
Assessment of active participation	10.00%	10.00%	Non-compulsory activity that can be retaken (rescheduling). To be carried out in the theory/laboratory sessions for the students of the continuous modality. The students of non-continuous modality will be evaluated of this activity through an alternative system in the ordinary call
Total:	**100.00%**	**100.00%**	

According to art. 4 of the UCLM Student Evaluation Regulations, it must be provided to students who cannot regularly attend face-to-face training activities the passing of the subject, having the right (art. 12.2) to be globally graded, in 2 annual calls per subject , an ordinary and an extraordinary one (evaluating 100% of the competences).

Evaluation criteria for the final exam:

Continuous assessment:

In compulsory activities, a minimum mark of 40% is required in order to pass that activity and have the possibility to therefore pass the entire subject. The evaluation of the activities will be global and therefore must be quantified by means of a single mark. In the case of the activities that may be retaken (i.e., rescheduling), an alternative activity or test will be offered in the resit/retake exam call (convocatoria extraordinaria).

The final exam will be common for all the theory/laboratory groups of the subject and will be evaluated by the lecturers of the subject in a serial way, i.e., each part of the final exam will be evaluated by the same lecturer for all the students.

A student is considered to pass the subject if she/he obtains a minimum of 50 points out of 100, taking into account the points obtained in all the evaluable activities, and also has passed all the compulsory activities.

For students who do not pass the subject in the final exam call (convocatoria ordinaria), the marks of activities already passed will be conserved for the resit/retake examcall (convocatoria extraordinaria). If an activity is not recoverable, its assessment will be preserved for the resit/retake exam call (convocatoria extraordinaria) even if it has not been passed. In the case of the passed recoverable activities, the student will have the opportunity to receive an alternative evaluation of those activities in the resit/retake exam call and, in that case, the final grade of the activity will correspond to the latter grade obtained.

The mark of the passed activities in any call, except for the final exam, will be conserved for the subsequent academic year at the request of the student, provided that mark is equal or greater than 50% and that the activities and evaluation criteria of the subject remain unchanged prior to the beginning of that academic year.

The failure of a student to attend the final exam will automatically result in her/him receiving a "Failure to attend" (no presentado). If the student has not passed any compulsory evaluation activity, the maximum final grade will be 40%.

Non-continuous evaluation:

Students may apply at the beginning of the semester for the non-continuous assessment mode. In the same way, the student may change to the non-continuous evaluation mode as long as she/he has not participated during the teaching period in evaluable activities that together account for at least 50% of the total mark of the subject. If a student has reached this 50% of the total obtainable mark or the teaching period is over, she/he will be considered in continuous assessment without the possibility of changing to non-continuous evaluation mode.

Students who take the non-continuous evaluation mode will be globally graded, in 2 annual calls per subject, an ordinary and an extraordinary one (evaluating 100% of the competences), through the assessment systems indicated in the column "Non-continuous evaluation".

In the "non-continuous evaluation" mode, it is not compulsory to keep the mark obtained by the student in the activities or tests (progress test or partial test) taken in the continuous assessment mode.

Specifications for the resit/retake exam:

Evaluation tests will be conducted for all recoverable activities.

Specifications for the second resit / retake exam:

Same characteristics as the resit/retake exam call.

9. Assignments, course calendar and important dates	
Not related to the syllabus/contents	
Hours	hours

General comments about the planning: The subject is taught in 3 x 1,5 hour sessions per week.

Unit 1 (de 6): Methodologies and Software Development Processes	
Activities	Hours
Class Attendance (theory) [PRESENCIAL][Lectures]	3
Individual tutoring sessions [PRESENCIAL][]	5
Individual tutoring sessions [PRESENCIAL][]	.9
Study and Exam Preparation [AUTÓNOMA][Self-study]	8.8
Other off-site activity [AUTÓNOMA][Practical or hands-on activities]	2
Other off-site activity [AUTÓNOMA][Practical or hands-on activities]	7.5
Final test [PRESENCIAL][Assessment tests]	1.5

Unit 2 (de 6): Software Configuration Management	
Activities	Hours
Class Attendance (theory) [PRESENCIAL][Lectures]	3
Individual tutoring sessions [PRESENCIAL][]	2.5
Individual tutoring sessions [PRESENCIAL][]	.9
Study and Exam Preparation [AUTÓNOMA][Self-study]	8.8
Other off-site activity [AUTÓNOMA][Practical or hands-on activities]	5
Other off-site activity [AUTÓNOMA][Practical or hands-on activities]	2.5
Final test [PRESENCIAL][Assessment tests]	1.5

Unit 3 (de 6): Quality of Software Products	

Activities	Hours
Class Attendance (theory) [PRESENCIAL][Lectures]	2
Individual tutoring sessions [PRESENCIAL][]	.9
Study and Exam Preparation [AUTÓNOMA][Self-study]	5.8
Other off-site activity [AUTÓNOMA][Practical or hands-on activities]	2.5
Final test [PRESENCIAL][Assessment tests]	1.5

Unit 4 (de 6): Verification and Validation of the Software

Activities	Hours
Class Attendance (theory) [PRESENCIAL][Lectures]	6
Individual tutoring sessions [PRESENCIAL][]	4
Individual tutoring sessions [PRESENCIAL][]	.9
Study and Exam Preparation [AUTÓNOMA][Self-study]	17.5
Other off-site activity [AUTÓNOMA][Practical or hands-on activities]	5
Other off-site activity [AUTÓNOMA][Practical or hands-on activities]	7.5
Final test [PRESENCIAL][Assessment tests]	1.5

Unit 5 (de 6): Software Maintenance

Activities	Hours
Class Attendance (theory) [PRESENCIAL][Lectures]	4
Individual tutoring sessions [PRESENCIAL][]	3.5
Individual tutoring sessions [PRESENCIAL][]	.9
Study and Exam Preparation [AUTÓNOMA][Self-study]	11.6
Other off-site activity [AUTÓNOMA][Practical or hands-on activities]	3
Other off-site activity [AUTÓNOMA][Practical or hands-on activities]	2.5
Final test [PRESENCIAL][Assessment tests]	1.5

Unit 6 (de 6): Lab sessions

Activities	Hours
Laboratory practice or sessions [PRESENCIAL][Practical or hands-on activities]	15

Global activity

Activities	hours
Individual tutoring sessions [PRESENCIAL][]	15.9
Study and Exam Preparation [AUTÓNOMA][Self-study]	52.5
Other off-site activity [AUTÓNOMA][Practical or hands-on activities]	17.5
Laboratory practice or sessions [PRESENCIAL][Practical or hands-on activities]	15
Class Attendance (theory) [PRESENCIAL][Lectures]	18
Final test [PRESENCIAL][Assessment tests]	7.5
Total horas:	**126.4**

10. Bibliography and Sources

Author(s)	Title/Link	Publishing house	City	ISBN	Year	Description
Bourque, P., Fairley, R.E.	Guide to the Software Engineering Body of Knowledge (SWEBOOK v3.0) https://www.computer.org/education/bodies-of-knowledge/software-engineering	IEEE Computer Society		0-7695-5166-1	2014	Cuerpo de conocimiento de los procesos software descritos en la asignatura
Arlow, Jim	UML 2 and the unified process : practical object-oriented an	Addison-Wesley		978-0-321-32127-5	2005	Libro de referencia del tema 1 para entender cómo utilizar el lenguaje UML para crear los distintos artefactos generados durante la ejecución de un proyecto siguiendo el Proceso Unificado de Desarrollo
PIATTINI, MARIO, CALVO-MANZANO, JOSÉ A.,CERVERA, JOAQUÍN, FERNANDEZ,	ANALISIS Y DISEÑO DE APLICACIONES INFORMATICAS DE GESTION http://www.ra-ma.es/libros/ANALISIS-Y-DISENO-DE-APLICACIONES-INFORMATICAS-DE-GESTION-CFGS-INCLUYE-CD-ROM/115/978-84-7897-587-7	RA-MA		78-84-7897-587-7	2003	Libro de Referencia sobre conceptos básicos relacionados con los distintos procesos software
RICARDO PÉREZ DEL CASTILLO, FRANCISCO / RUÍZ GONZÁLEZ, IGNACIO / RODRÍGUEZ, MACARIO POLO / PIATTINI VELTHUIS, MARIO G	MANTENIMIENTO Y EVOLUCIÓN DE SISTEMAS DE INFORMACIÓN http://www.ra-ma.es/libros/MANTENIMIENTO-Y-EVOLUCION-DE-SISTEMAS-DE-INFORMACION/99623/978-84-9964-759-3	RA-MA	Madrid	978-84-9964-759-3	2018	Libro de referencia para el tema de Mantenimiento