

## COURSE PROGRAM

Academic Year: 2024/2025

Identification and characteristics of the course			
Code	501316	ECTS Credits	6
Course title (English)	Software Project Management		
Course title (Spanish)	Gestión de Proyectos Software		
Degree programs	Bachelor Degree in Computer Software Engineering		
Faculty/School	Escuela Politécnica (EP)		
Semester	7th	Type of course	compulsory
Module	Software Engineering Specific Technology		
Matter	Software Engineering		
Lecturer/s			
Name	Room	E-mail	Web page
José Javier Berrocal Olmeda	117 (Inf.)	jberolm@unex.es	
José Manuel García Alonso	117 (Inf)	jgaralo@unex.es	
Subject Area	Computer Languages and Systems		
Department	Computers and Telematics Systems Engineering		
Coordinator (Only if there is more than one lecturer)	José Javier Berrocal Olmeda		

Competencies*
<b>Basic Competencies</b>
<b>CB1</b> - That the students have demonstrated to possess and understand knowledge in an area of study that starts from the base of general secondary education, and is usually found at a level that, while relying on advanced textbooks, also includes some aspects that involve knowledge from the forefront of their field of study.
<b>CB2</b> - That students know how to apply their knowledge to their work or vocation in a professional manner and possess the skills that are usually demonstrated through the development and defense of arguments and problem solving within their field of study.
<b>CB3</b> - That students have the ability to gather and interpret relevant data (usually within their area of study) to make judgments that include reflection on relevant social, scientific or ethical issues.
<b>CB4</b> - Students should be able to convey information, ideas, problems and solutions to both specialized and non-specialized audiences.
<b>CB5</b> - That students should have developed those learning skills necessary to undertake further studies with a high degree of autonomy.
<b>General competencies</b>

\* The sections concerning competencies, course outline, teaching activities, teaching methodology, learning outcomes and assessment methods must conform to those included in the ANECA verified document of the degree program.

<p><b>CG04</b> - Ability to define, evaluate and select hardware and software platforms for the development and implementation of computer systems, services and applications.</p>
<p><b>CG08</b> - Knowledge of the basic subjects and technologies, which enable students to learn and develop new methods and technologies, as well as those that provide students with a great versatility to adapt to new situations.</p>
<p><b>CG09</b> - Ability to solve problems with initiative, decision-making, autonomy and creativity. Ability to know how to communicate and transmit the knowledge, skills and abilities of the profession of Technical Engineer in Computer Science.</p>
<p><b>CG10</b> - Knowledge to carry out measurements, calculations, valuations, appraisals, surveys, studies, reports, task planning and other similar IT work.</p>
<p><b>CG01</b> - Ability to conceive, write, organize, plan, develop and sign projects in the field of computer engineering that aim, the conception, development or operation of computer systems, services and applications.</p>
<p><b>CG03</b> - Ability to design, develop, evaluate and ensure the accessibility, ergonomics, usability and security of computer systems, services and applications, as well as the information they manage.</p>
<p><b>General competencies of the "Software Engineering Specific Technology" module</b></p>
<p><b>CIS01:</b> Ability to develop, maintain and evaluate software services and systems satisfying the user requirements getting them behaving in a reliable and efficient way, keeping them affordable and maintainable and making them meet the quality rules applying the Software Engineering theories, principles, methods and practices.</p>
<p><b>CIS02:</b> Ability to assess the customer needs and specify software requirements capturing them, reconciling conflicting objectives by finding acceptable compromises within the constraints arising from the cost, the time, the legacy systems and the organizations themselves.</p>
<p><b>CIS05:</b> Ability to identify, assess and manage the potential risks that may arise.</p>
<p><b>Transverse Competencies of the shared module in the field of Informatics</b></p>
<p><b>CT01:</b> Ability for organization and planning.</p>
<p><b>CT12:</b> Act with responsibility and professional ethics.</p>
<p><b>Learning objectives of the course</b></p>
<p>To properly develop the competencies assigned to this course and to achieve the proposed learning outcomes, the following specific objectives of learning are established classified according to Bloom's taxonomy in the levels of knowledge, comprehension, application and analysis.</p>
<p><b>Knowledge:</b></p> <ul style="list-style-type: none"> <li>Obj. 1. Know the relationships between the Software Project Management course and the rest of courses of the Bachelor Degree in Computer - Software Engineering (<b>CIS01</b>).</li> <li>Obj. 2. Know the main Project Management Processes and Project Management Knowledge Areas of the Project Management Body of Knowledge and the relationships between them. (<b>CIS01, CIS02, CT01</b>).</li> <li>Obj. 3. Know the main Process Development Models for Software Projects and their relations with different software development methodologies. (<b>CIS01, CIS02, CT01</b>).</li> <li>Obj. 4. Know the basic strategies for generating schedules for Software Projects (<b>CIS01, CIS05, CT01</b>).</li> <li>Obj. 5. Know tools for Software Projects Schedules management (<b>CIS01, CIS02, CT01, CT12</b>).</li> <li>Obj. 6. Know tools for Configuration Management of Software Projects (<b>CIS01, CIS02, CT01, CT12</b>).</li> <li>Obj. 7. Know tools for Continuous Integration of Software Projects (<b>CIS01, CIS02, CT01, CT12</b>).</li> <li>Obj. 8. Know tools for Product Quality Management in Software Projects (<b>CIS01, CIS02, CT01, CT12</b>).</li> <li>Obj. 9. Know tools for Testing Management in Software Projects (<b>CIS01, CIS02, CT01, CT12</b>).</li> <li>Obj. 10. Know tools for Jobs Allocation in Software Projects (<b>CIS01, CIS02, CT01, CT12</b>).</li> </ul>

Obj. 11. Know tools for the Issues Management in Software Projects (**CIS01, CIS02, CT01, CT12**).

**Understanding:**

- Obj. 12. Understand the need for the different disciplines of Software Project Management (**CIS01, CIS02, CT01, CT12**).
- Obj. 13. Relate the technical development skills acquired in previous courses with their application to the industry and business (**CIS01, CIS02, CT01, CT12**).
- Obj. 14. Differentiate each of the disciplines and Knowledge areas involved in Software Projects Management (**CIS01, CIS02, CT01, CT12**).
- Obj. 15. Differentiate the different Software Process Development Models and to understand the differences between the sequential and the iterative time management strategies as well as understand the advantages and disadvantages of the Agile Time Management (**CIS01, CIS02, CT01, CT12**).
- Obj. 16. Understand the need of producing time schedules as roadmaps for the Software Project Management and to understand their utility as elements to keep development time and costs under control (**CIS01, CIS02, CT01, CT12**).
- Obj. 17. Understand the problem raised by different working teams contributing to the same Software Project Development and how it needs some tools and procedures to help avoiding interferences between developers (**CIS01, CIS02, CT01, CT12**).
- Obj. 18. Understand the difficulties of generating integrated images of software projects when there are several working teams contributing simultaneously and the need of software tools providing automated support to such task (**CIS01, CIS02, CT01, CT12**).
- Obj. 19. Understand the complexity of the testing process during Software Project Development and to differentiate the different kind of tests that can be carried out. To relate the above task with the Issues Management (**CIS01, CIS02, CT01, CT12**).
- Obj. 20. Understand the need of defining quality criteria for the code resulting from Software Projects and how automated tools can help forcing criteria (**CIS01, CIS02, CT01, CT12**).
- Obj. 21. Understand the need of logging man ours in Software Projects and relate them with the time schedules and their control (**CIS01, CIS02, CT01, CT12**).

**Application:**

- Obj. 22. Build project time schedules based on a iterative and incremental process model (**CIS01, CIS02, CT01, CT12**).
- Obj. 23. Set up and use a Configuration Management tool for a software project development team (**CIS01, CIS02**).
- Obj. 24. Set up and use a continuous integration tool for a software project development team (**CIS01, CIS02**).
- Obj. 25. Set up and use a code quality management tool for software projects (**CIS01, CIS02**).
- Obj. 26. Set up and use a test management tool for software projects (**CIS01, CIS02**).
- Obj. 27. Set up and use an issues management tool for software projects (**CIS01, CIS02**).
- Obj. 28. Set up and use a man hours logging tool for software projects (**CIS01, CIS02**).

**Analysis:**

- Obj. 29. Relate the man hours logging discipline with those of time management and cost management in Software Projects (**CIS01, CIS02, CT07**).
- Obj. 30. Relate the man hours logging discipline with those of time and cost management in Software Projects (**CIS01, CIS02, CT07**).

<b>Contents</b>
<b>Course outline*</b>
<p>This course focuses on with the Project Planning discipline in the field of Software Engineering. it will deal with the following concepts: Project planning, management and monitoring; Quality control: testing techniques and metrics; Software quality support tools; Risk control.</p> <p>It also addresses the preparation of the appropriate environment for the development and execution of projects in the field of Software Engineering with the aim of a correct management and deployment of the developed product, introducing a set of tools that will help to manage the project in a satisfactory way.</p> <p>The aim of the subject is to provide to the student with a global view of what Software Project Management consists of, providing him/her with the necessary competences to prepare the whole project execution environment in order to facilitate its adequate management.</p>
<b>Course contents</b>
<p>Theme 1 Title: Introduction            Theme 1 Contents: Introduction to project management in ICT engineering, existing standards and certifications, introduction to project management processes.</p>
<p>Theme 2 Title: Software development process models and planning techniques.            Theme 2 Contents: Introduction to the development processes that can be followed during project management. The most important milestones and activities that must be taken into account during the planning and execution of a project. Introduction to the techniques and tools for planning tasks, their dependencies and scheduling.</p> <p>Description of the practical activities of theme 2: Initial project planning</p>
<p>Theme 3 Title: Software project progress management            Theme 3 Contents: During this theme the student will be shown what project progress management tools are, their most important concepts, and how they should be used to monitor and control the progress of a project.</p> <p>Description of practical activities for theme 3: Planning control</p>
<p>Theme 4 Title: Software configuration and continuous integration management            Theme 4 Contents: Methodologies for managing the development and implementation of a product. Use of tools for versioning management, collaborative work, and project deployment.</p> <p>Description of practical activities for theme 4: Configuration management, development and continuous integration</p>
<p>Theme 5 Title: Software quality management            Theme 5 Contents: During this theme students will see different tools to control the quality of software during its development. They will also see tools for continuous quality management.</p> <p>Description of the practical activities of topic 5: Software quality analysis</p>
<b>Description of practical activities</b>

Please, note that 5 hours of the practical activity P5 will be used to also introduce English expressions on project management and software quality.

### Educational activities \*

Student workload (hours per lesson)		Lectures	Practical sessions				Monitoring activity	Homework
Lesson	Total	L	HI	LAB	COM	SEM	SGT	PS
I	4	2						2
II	46	8		8				30
III	18	4		4			1	9
IV	46	8		8			1	29
V	32	6		8			1	17
<b>Assessment **</b>	4	2		2				
<b>TOTAL ECTS</b>	150	30		30			3	87

L: Lectures (85 students)

HI: Hospital internships (7 students)

LAB: Lab sessions or field practice (15 students)

COM: Computer room or language laboratory practice (30 students)

SEM: Problem-solving classes, seminars or case studies (40 students)

SGT: Scheduled group tutorials (educational monitoring, ECTS type tutorials)

PS: Personal study, individual or group work and reading of bibliography

### Teaching Methodology\*

The "learning by doing" and "problem-based learning" will be used in this subject. Therefore, to achieve the learning objectives of the course, the following training activities will arise:

Large groups in classroom:

- Explanation of concepts lessons
- Exercises lessons
- Exercises resolution
- Exercises resolution by groups
- Presentation of solutions by groups
- Multiple choice exercises

Classes in laboratory

- Guided exercises
- Sample software already built
- Implementing programming exercises
- Detecting errors in programs
- Use of External Libraries
- Building software projects
- Use of data structures
- Executing test
- Use of the virtual campus

Non-classroom activities

- Study of the course themes

\*\* Insert as many rows as necessary. For instance, you can include one row for a partial exam and another for the final exam.

- Personal work using the development environment
- Search of information (books, Internet, etc.)
- Working groups meetings
- Software Programming
- Building documentation for the practical activities
- Use of the course forums
- Solving problems proposed in classes

### Learning outcomes \*

#### Learning outcomes of the general competencies:

- Masters the basic concepts of Software Project Management, as well as techniques and tools to measure productivity and quality of a software project being able to analyse the risks that may affect the development of a project.
- Knows in detail the standards to ensure the quality of software systems and their development process.

#### Learning outcomes of the transverse competencies:

- Knows and apply the transversal competencies in advanced activities

### Assessment methods \*

The course offers two alternative evaluation itineraries. During the first quarter of the course, students will be allowed to choose between any of the below detailed itineraries for their evaluation.

**Itinerary A: Continuous assessment.** The continuous assessment itinerary requires from the student the attendance to all large group, laboratory and monitoring activities. Besides. Students are required to sit the five practical activities proposed along the course (P1 to P5). Practical activities are evaluated independently of each other. They require a minimum grade of 5. The overall grade of the practical activities is obtained as a weighted average of the grades obtained for each activity using the following formula:

$$\text{PA Grade} = (25\% \text{ P1} + 20\% \text{ P2} + 30\% \text{ P3} + 25\% \text{ P4})$$

Students must also sit an exam about the theoretical content of the course. The final grade for the course is then obtained using the following formula:

$$\text{Final Grade} = 60\% \text{ PA Grade} + 30\% \text{ Exam Grade} + 10\% \text{ Presentation}$$

To apply the above formula and to pass the course, the student must obtain a mark equal to or greater than 5 in every practical activity and in the theoretical exam. Both the practical activities (as a whole) and the theoretical exam can be recovered in the normal call. Partial grades are not save for other calls.

*Continuous assessment process.* The development of the practical activities in the continuous assessment itinerary will be carried out in groups and, if the student agrees on it, in coordination with the course Software Architectures for Enterprise Environments

(7th semester). Therefore, it is recommended to take both courses at the same time. The development of a joint project allows students to get a more complete training, providing them with both a software architecture and a project management vision for the development of the same product. Furthermore, this collaboration reduces the effort required to develop the software product for both courses, since it is the same one.

Students who do not take both courses in the same semester, or those not agreeing on integrating both courses, will be organized into groups that will develop the software product only from the perspective of this Software Project Management. Although, they will need to carry out a basic development in order to be able to complete the activities and apply the learnt knowledge.

**Itinerary B: Global assessment.** This itinerary is for those students that decided not to use the Continuous Assessment. Students must sit the exam about the theoretical contents of the course. Besides, they must also sit an exam about the practical contents of the course. The final grade is then obtained using the following formula:

$$\text{Final Grade} = 70\% \text{ Practical Exam Grade} + 30\% \text{ Theoretical Exam Grade}$$

To apply the above formula and to pass the course, the student must obtain a mark equal to or greater than 5 in each of the practical contents of the course and in the theoretical part of the exam.

### Bibliography (basic and complementary)

#### Basic bibliography

1. A Guide to the Project Management Body of Knowledge: (Pmbok Guide). Project Management Institute (Corporate Author). Project Management Institute; 4 edition. 2008. ISBN-13: 978-1933890517

#### Complementary bibliography

2. The Unified Software Development Process. Ivar Jacobson, Grady Booch and James Rumbaugh. ADDISON WESLEY. ISBN-13: 978-8478290369.
3. Succeeding with Agile: Software Development Using Scrum. Mike Cohn. Addison-Wesley. ISBN-13: 978-0321822000

#### Other resources

4. Microsoft Project 2013 Step by Step. Carl Chatfield, Timothy Johnson. Microsoft Press. ISBN-13: 978-0735669116
5. JIRA. <http://www.atlassian.com/software/jira/overview>
6. Git. <http://git-scm.com/book>
7. Sonar. <http://www.sonarsource.org/>
8. Jenkins. <http://jenkins-ci.org/>
9. JUnit. <http://junit.org/>

### Other resources and complementary materials

1. Mantis. <http://www.mantisbt.org/>
2. IntelliJ. <https://www.jetbrains.com/idea/>
3. Eclipse. <https://eclipse.org/>
4. Software Configuration Management Patterns: Effective Teamwork, practical integration. Berczuk, S.P. y Appleton, B. Addison-Wesley. 2002
5. Software Measurement and Estimation: A Practical Approach. Laird, L.M. y Brennan, M. C. Wiley-IEEE Computer Society. 2006.
6. Rapid Development. McConnell, S. Microsoft Press. 1996.
7. Software Estimation: Demystifying the Black Art. McConnell, S. Microsoft Press. 2006.
8. Suárez Suárez, A. Decisiones óptimas de inversión y financiación en la empresa. Pirámide. 1994.