30248 - Software Project Management

Syllabus Information

Academic Year: 2019/20
Subject: 30248 - Software Project Management
Faculty / School: 110 - Escuela de Ingeniería y Arquitectura
Degree: 439 - Bachelor's Degree in Informatics Engineering
ECTS: 6.0
Year: 4
Semester: First semester
Subject Type: ---
Module: ---

1.General information

1.1.Aims of the course

At the end of the course the students will have completed a team software project, with a theme of their choice, managed with the agile Scrum methodology. During the work they will evaluate and make technical decisions, balance the development effort with the creation of the documentation that provides the most value and analyze their management process with the aim of improving it.

1.2.Context and importance of this course in the degree

In the first two years of the degree, students acquire skills and knowledge that will enable them to develop small computer applications. The Software Engineering course has provided students with the engineering fundamentals required for the professional development of medium and large sized software systems, and the Software Project course has provided them with the integrating and supporting element that allows the process of building a software system to be carried out by a team in a controlled manner. This course examines in depth the techniques that allow the systematization of the activities and processes necessary for the effective development of any software system, adapting to the current requirements of agility, quality and continuous improvement.

1.3.Recommendations to take this course

Given the configuration of the curriculum of the degree, in the first two years of the degree the student acquires skills and knowledge that will enable him/her to develop small computer applications. Since the acquisition of knowledge and skills in this subject is carried out around the development of a small software application, but developed following software engineering techniques and with a defined development process, it is necessary to have passed the first two years of the curriculum and, at least, the subjects of Software Engineering and Software Project of the fifth and sixth four-month periods respectively.

2.Learning goals

2.1.Competences

Upon passing the course, the student will be more competent to....
Successfully address the following cross-cutting issues:

- CT1. Ability to conceive, design and develop engineering projects.
- CT2. Ability to plan, budget, organize, direct and control tasks, people and resources.
- CT4. Ability to solve problems and make decisions with initiative, creativity and critical thinking.
- CT7. Ability to analyze and assess the social and environmental impact of technical solutions acting with ethics, professional responsibility and social commitment.
- CT8. Ability to work in a multidisciplinary group and in a multilingual environment.

Successfully undertake the following tasks related to Computer Engineering in general:

- CGC2. Ability to plan, design, deploy and manage projects, services and IT systems in all areas, leading their implementation and continuous improvement and assessing their economic and social impact.
- CGC3. Ability to understand the importance of negotiation, effective work habits, leadership and communication skills in all software development environments.

- CGC4. Ability to prepare the technical specifications of a computer installation that complies with current standards and regulations.
- CGC8. Ability to analyze, design, build and maintain applications in a robust, secure and efficient manner, choosing the most appropriate paradigm and programming languages.

Successfully address the following Software Engineering related performance issues:
- CEIS2. Ability to assess customer needs and specify software requirements to meet these needs, reconciling conflicting objectives by seeking acceptable compromises within the constraints of cost, time, existing systems already developed and the organisations themselves.
- CEIS3. Ability to provide solutions to integration problems according to the available strategies, standards and technologies.
- CEIS5. Ability to identify, assess and manage potential associated risks that may arise.
- CEIS6. Ability to design appropriate solutions in one or more application domains using software engineering methods that integrate ethical, social, legal and economic aspects.

2.2.Learning goals

In order to pass this course, the student must demonstrate the following results...
- Learns about strategies and approaches to develop and manage the processes involved in obtaining a contract for a software project. This includes approximations for the definition of objectives and deliverables of a project, estimation of the cost of the project and the preparation of a budget for the project.
- Knows the bases to approach the management and optimization of the human team that integrates the project. This includes strategies for team building, tools for optimising team performance (based mainly on group dynamics), and approaches to identifying, characterising and assigning roles within a project.
- Knows the concept of risk within a software project. As well as mechanisms for the planning of its management. These mechanisms include, among other things, the identification, assessment, selection and definition of mitigation strategies.
- Knows the conceptual bases and diverse techniques for the follow-up, revision and evaluation of a software project.
- Knows procedures to carry out the closing of a software project, the implications that this has, the measurement and evaluation of a project, as well as the use of the information generated by these processes.
- Gets to know the problems associated with software maintenance.
- Knows how to manage and organize the activities involved in software maintenance.
- Knows the ethical, social, legal and economic aspects intrinsic to the development of a business software project, general and specific to the scope of one or several application domains.
- Knows an infrastructure of processes and tools necessary to develop a software project, based on the best practices of software engineering available in a software factory business environment.
- Puts into practice the knowledge acquired in the subjects of Software Engineering intensification in a specific project developed as a team: requirements, analysis, design, testing (verification and validation), project management.

2.3.Importance of learning goals

The reality of the industry shows that the development of software systems is almost always carried out within a budgetary framework, according to a schedule, and meeting a series of requirements negotiated with a client. This cannot be done without the necessary capacity to organize and structure the work, optimize the use of the resources available, systematize the tasks that can be repetitive (in order to devote more time to creative ones), and pursue a continuous improvement that results in a higher level of quality.

3.Assessment (1st and 2nd call)

3.1.Assessment tasks (description of tasks, marking system and assessment criteria)

The evaluation will consist of a test with two exercises:
- Delivery of project results (80%): the exercise consists of the delivery of results (technical report, source code and others) that reflect the work of students in a software development project that follows the agile Scrum methodology. The project will include the construction of a small-scale, open-ended computer application that will address all the tasks of software engineering, with special attention to those related to the management and support of the different development activities (all learning outcomes).
- Written questions about concepts learned in theory and problems (20%) (learning outcomes 1, 2, 3, 4, 5, 10, 16, 17).

In order to pass the course, the weighted sum of both exercises must be at least 5 out of 10.

4.Methodology, learning tasks, syllabus and resources

4.1.Methodological overview

The learning process designed for this course is based on:

1. Study and work from day one.
2. Learning concepts about the different activities in software project management, as well as about its legal framework, through classroom lectures.
3. Applying those concepts to practical cases, where the students will have an active role in the discussion and resolution of those cases.
4. Teamwork to develop a small software project, with an emphasis on the activities related to the management of the process.
5. External professionals will be invited to show the students the reality of the profession.

4.2.Learning tasks

The course includes the following learning tasks:

1. The theory will be developed in the classroom lectures.
2. The practical cases in the classroom will allow the students to apply the theoretical concepts and techniques.
3. The students will develop a software project in a team, under the supervision of the professors.

4.3.Syllabus

The course will address the following topics:

1. Agile management with Scrum: agile principles, sprints, user stories, the product backlog, estimation, velocity, planning and technical debt
2. Project management techniques: integration, scope, time, cost, quality, people and risks

4.4.Course planning and calendar

- Classroom lectures (2 hours per week)
- Practical cases in the classroom (1 hour per week)

The schedules and dates for all activities will be published on the web page of the EINA and in the Moodle page of the course.

The deadlines for the project and the exam dates will be adjusted to the academic schedule following the University regulations.

The students should work around 150 hours distributed as follows:

- 45 hours in classroom activities.
- 15 hours of study and evaluation.
- 90 hours in the project.

4.5.Bibliography and recommended resources

[BB: Bibliografía básica / BC: Bibliografía complementaria]

http://psfunizar7.unizar.es/br13/egAsignaturas.php?codigo=30248&Identificador=14713

- [BB] Rubin, Kenneth S. Essential Scrum : a practical guide to the most popular agile process / Kenneth S. Rubin . 7th print. Upper Saddle River (New Jersey) : Addison-Wesley, imp. 2015
- [BC] Larman C. Practices for Scaling Lean & Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum / Larman C, Vodde B. - 1st edition Addison-Wesley Professional, 2010