Software Development

Academic Year: (2022 / 2023) **Review date: 25-05-2022**

Department assigned to the subject: Department of Computer Science and Engineering

Coordinating teacher: LOPEZ CUADRADO, JOSE LUIS

Type: Compulsory ECTS Credits : 6.0

Year : 2 Semester : 2

REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

Programming (Year 1 / Semester 1)
Data structures & Algorithms (Year: 1 / Semester 2)
Software Engineering (Year 2 / Semester 1)
Automata and formal language theory (Year 2 / Semester 1)

SKILLS AND LEARNING OUTCOMES

¿ Conceive, design and apply functional, structural and unit testing techniques for software verification in different software development processes.
¿ Know and become aware of the ethical and legal principles of the profession of Computer Engineering.

OBJECTIVES

The objective of the course is to allow the student to deepen in agile development practices that facilitate the achievement of high-quality software components. Functional and structural testing techniques will be learned and applied through automated unit testing in a Test-Driven Development process. The ethical and legal foundations of the Computer Engineering profession will also be introduced. Finally, students will learn code refactoring techniques and apply simple design concepts and design patterns to assign responsibilities.

DESCRIPTION OF CONTENTS: PROGRAMME

1.- Ethic and Legal Issues in the Software Engineering Profession
1.1.- The software engineering profession.
1.2.- The software engineers' code of ethics.
2.- Agile Software Development Techniques
2.1.- Software development process
2.2.- Coding Standards
2.3.- Code Collective Ownership
3.- Test Driven Development
3.1.- Principles of Test Driven Development
3.2.- Functional Testing Techniques
3.3.- Estructural Testing Techniques
3.4.- Unit Testing Automation
3.5.- Automated Continuous Integration
4.- Refactoring and Simple Design
4.1.- Refactoring
4.2.- Principles of Software Design
4.3.- Design Patterns for Responsibilities Assignment

LEARNING ACTIVITIES AND METHODOLOGY

Lectures: 1,5 ECTS, to get the specific cognitive and instrumental competences of the subject.

Exercise Classes: 1,5 ECTS, to get the specific instrumental and generic competences, as well as the attitude competences of the subject. A practical example related to the development of a software component (including estimation, specification, design, software reviews and testing activities) will be carried out, considering ethic and legal features.

Practical Work with and without professor assistance: 2 ECTS, work on specific software component development case, analyzing all the aspects considered in the theoretical part of the subject.

Tutorials: Individualised assistance (individual tutorials) or group assistance (group tutorials) to students by the lecturer.

Exam (Final Practice): 1 ECTS

ASSESSMENT SYSTEM

The exercises and exams, in addition to serving as a training activity, have the double objective of being measured by the evaluation system. The evaluation system includes the evaluation of the directed academic and practical activities according to the following weighting.

Guided Exercises (Continuous assessment) 30%
Theory Tests/Exercises (Continuous assessment) 30%
Final Practice (Exam) 40%

It is mandatory to pass, separately, each one of the parts (theory, guided exercises and final practice) to pass the whole subject

% end-of-term-examination:	40
% of continuous assessment (assigments, laboratory, practicals…):	60

BASIC BIBLIOGRAPHY

- Beck, Ken, et al.. Test-Driven Development By Example. , Three Rivers Institute., 2002
- Beck, Ken. Extreme Programming Explained., Addison-Wesley. , 2000
- Craig S. Larman Applying UML and Patterns., Pearson Education . 3er Edition, 2012
- Fowler, Martin et al.. Refactoring: Improving the Design of Existing Code. , Addison-Wesley. , 1999
- Lee Copeland. A Practitioner's Guide to Software Test Design., Artech House Publishers, 2003

ADDITIONAL BIBLIOGRAPHY

- Paul C. Jorgensen Software Testing: a craftsman's approach. , CRC.
- Roger S. Pressman. Software Engineering. A practical approach., McGraw Hill. 7ª Edición., 2009
- Spyros Xanthakis, Michel Maurice, Antonio de Amescua, Olivier Houri, Luc Griffet. Test and contro¿le des logiciels : me¿thodes, techniques and outils, EC2..