

Academic Year: (2024 / 2025)

Review date: 25-04-2024

Department assigned to the subject: Computer Science and Engineering Department

Coordinating teacher: LLORENS MORILLO, JUAN BAUTISTA

Type: Compulsory ECTS Credits : 6.0

Year : 4 Semester : 2

REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

Software engineering (Course: second - semester: first)
 Software Development (Course: second - semester: second)
 Software development projects management (Course: fourth - semester: first, recommended)

SKILLS AND LEARNING OUTCOMES

- ¿ To understand the types of corporate information systems in relation to business models.
- ¿ Know and apply integrated project management, resource estimation and quality management.
- ¿ Know and apply change and maintenance management and audit processes.

OBJECTIVES

The subject objective is to obtain the necessary knowledge and skills to design, plan, build, deploy and operate a software system, ensuring the quality requirements in the different environments and architectures defined for its construction.

DESCRIPTION OF CONTENTS: PROGRAMME

- Basis and concepts withing the development and operation of software systems
- Planification of software systems
- Architecture and development of software systems
- Deployment of software systems
- Operation and monitoring of software systems
- Quality assurance of software systems

LEARNING ACTIVITIES AND METHODOLOGY

- Theoretical-Practical Lectures: 1 ECTS
- Review of contents before class
- Practical Lectures: 1 ECTS
- Exercise resolution
 - Partial oral presentation of the project
- Team Work: 1 ECTS
- Project development
 - Project review
- Individual Work: 1 ECTS
- Contribution to team project
 - Individual practical exercises
 - Study and preparation of theoretical exams
- Tutoring: 1ECTS
- Individual or group based tutorship sessions with the professor

ASSESSMENT SYSTEM

% end-of-term-examination:	0
% of continuous assessment (assignments, laboratory, practicals...):	100

CONTINUOUS EVALUATION (100%)

The evaluation of the subject is based on the development of a team project applying the theoretical concepts and tools used in the weekly practices to build a system including software and hardware elements.

During the execution of the course, two reviews of its development status will be carried out. The first review will require the assurance of the technology stack to be used. The second and final review will verify that the developed system is ready for operational use on the day of the exam. These verifications are evaluated using the Go/No Go technique.

On the day of the final exam, a final validation of the system will be carried out, as well as a functional demonstration under a competitive model. The system shall meet the functionality specified in the requirements specification, and must perform it without failures (adequate testing).

The evaluation is based on the following criteria:

- System verified and validated (functionally correct): passed
- Final System verification before competition 10%
- Classification in the competition 80%
- Role of each team member and 360° Evaluation between the team and the team leader. This last criterion allows establishing individual performance and discerning individual evaluation. 10%.

FINAL EVALUATION

-Final exam only if the student does not participate in the development of the project or the team leaders banes a person from the team (under the criteria clearly defined in the Project Charter). (100%)

BASIC BIBLIOGRAPHY

- Christof Ebert; Gorka Gallardo; Josune Hernantes; Nicolas Serrano DevOps, IEEE Software, 2016
- D. Farley Modern software engineering: doing what really works to build better software faster, Addison-Wesley, 2021
- G. Kim, K. Behr, and G. Spafford The phoenix project: a novel about IT, DevOps, and helping your business win, Portland, OR: IT Revolution, 2018
- J. Davis and K. Daniels Effective devOps: building a culture of collaboration, affinity, and tooling at scale, O¿Reilly, 2016
- M. Richards and N. Ford Fundamentals of software architecture: an engineering approach, O¿Reilly Media, 2020
- Martin Eigner System Lifecycle Management, Springer, 2021
- N. Ford, M. Richards, P. J. Sadalage, and Z. Dehghani Software architecture: the hard parts: modern trade-off analysis for distributed architectures, O¿Reilly Media, 2021
- N. Forsgren, J. Humble, and G. Kim Accelerate: the science behind DevOps: building and scaling high performing technology organizations, Portland, Oregon: IT Revolution, 2018
- R. C. Martin and R. C. Martin Clean architecture: a craftsman¿s guide to software structure and design, Prentice Hall, 2018
- Thomas M. Shortell INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, Willey, 2015

ADDITIONAL BIBLIOGRAPHY

- B. Beyer, C. Jones, J. Petoff, and N. R. Murphy Site reliability engineering: how Google runs production systems, O'Reilly, 2016
- C. Rosenthal and N. Jones Chaos engineering: system resiliency in practice, O'Reilly Media, 2020
- K. Morris Infrastructure as code: managing servers in the cloud, O'Reilly, 2016
- M. T. Nygard Release it! design and deploy production-ready software, Pragmatic Bookshelf, 2007

BASIC ELECTRONIC RESOURCES

- Amazon AWS . DevOps on AWS Specialization: <https://www.coursera.org/specializations/aws-devops>
- IBM . IBM DevOps and Software Engineering Professional Certificate: <https://www.coursera.org/professional-certificates/devops-and-software-engineering>
- Microsoft . Microsoft Certified: DevOps Engineer Expert: <https://learn.microsoft.com/en-us/certifications/devops-engineer/>
- Wilderness Labs . Meadow producer: <https://www.wildernesslabs.co/>