

SYSTEMS ANALYSIS AND DESIGN

Code: 10800

Main Scientific Area: Information Systems and Artificial Intelligence

Lecturer: Patrícia Isabel Sousa Trindade Silva Leite

Language of Instruction: Portuguese

Regime: S1

Contact Hours: 60h Total Workload: 100h

ECTS: 6,0

Objectives

This curricular unit aims to prepare the students to understand the basic concepts about system analysis and design. It is also intended to develop in the students the capabilities of abstraction and communication, using several techniques of requirements elicitation and analysis. The students will get skills to use methods and techniques of specification and modeling, suitable to document the activities of software systems development life cycle.

Learning Outcomes

On successful completion of this curricular unit, students should be able to: Describe the process of analysis and design of software systems; Demonstrate abstraction, communication and teamwork skills in the development of software systems; Analyze and model a software system using the UML modeling language; Develop a software project, integrating a team, using adequate support tools.

Course Contents

1. Software systems development What is software? Information Systems Software engineering process
2. Requirements engineering (RE) Software requirements RE process Communication with stakeholders Requirements elicitation Business process modeling
3. Requirements analysis and specification Requirements Analysis Specification techniques Requirements validation and management Agile requirements
4. Software modeling Systemic approach Software modeling using UML

Recommended Bibliography

Unhelkar, B. (2020). Software Engineering with UML. CRC Press (ISBN: 978-0367657383) Fernandes, J. M., Machado, R. J. (2016). Requirements in engineering projects. Springer. (ISBN: 978-3-319-18596-5) Seidl, M., Scholz, M., Huemer, C. Kappel, G. (2015). UML @ Classroom: An Introduction to Object-Oriented Modeling. Springer (ISBN 978-3319127415) Sommerville, I. (2011). Software Engineering. 9th Edition, Addison-Wesley Wieggers, K., Beatty, J. (2013). Software Requirements, 3Ed. Microsoft Press (ISBN: 978-0-7356-7966-5) Leffingwell, D. (2010). Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise. Addison-Wesley (ISBN-13: 978-0-321-63584-6)

Learning and Teaching Methods

The program of the curricular unit introduces the key concepts that allow understanding the scope and context in which the system analysis and design activities are developed. To develop skills in this field, students should be familiarized with the principles and best practices of the system analysis and design, to know how to use UML, and to understand the agile software process.

Assessment Methods

Learning outcomes will be assessed through (a) a theoretical component and (b) a practical component. The theoretical component consists of two individual written tests and the practical component consists of the phased development of a project team. The practical component will be held during the lessons. The final grade (FG) is a weighted average calculated according to the expression:

$$FG = \text{Written Test 01} * 25\% + \text{Written Test 02} * 25\% + \text{Practical Component} * 35\% + \text{Project 50+10} * 15\%$$

The evaluation of the practical component includes a session of individual defence of the project, being the classification given individually to each member of the working group. Approval for the curricular unit is subject to obtaining a minimum score of 8.0 (scale from 0 to 20) for each component.

The final exam evaluates only the theoretical component, keeping, for the calculation of the final grade, the value obtained in the practical component upon attending the course unit.

The final exam grade (EG) is the weighted average calculated according to the following expression:

$$EG = \text{Exam} * 50\% + \text{Practical Component} * 35\% + \text{Project 50+10} * 15\%$$