

Academic Year: (2024 / 2025)

Review date: 26-04-2024

Department assigned to the subject: Computer Science and Engineering Department

Coordinating teacher: ALVAREZ RODRIGUEZ, JOSE MARIA

Type: Compulsory ECTS Credits : 6.0

Year : 2 Semester : 1

REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

Writing and Communication Skills (1st course, 1st semester)

SKILLS AND LEARNING OUTCOMES

- ¿ Know how to obtain, describe and manage requirements for the functional and non-functional description of a software application.
- ¿ Design and develop conceptual and architectural models of a software application.

DESCRIPTION OF CONTENTS: PROGRAMME

Block I. Requirements engineering

Unit 1. Introduction to requirements engineering

Unit 2. Elicitation, description and management of requirements

Unit 3. Properties, attributes and organization of requirements

Unit 4. Types of requirements

Block II. Conceptual modeling with UML

Unit 5. Introduction to conceptual modeling

Unit 6. Conceptual modeling: classes and objects

Unit 7. Conceptual modeling: associations

Unit 8. Conceptual modeling: hierarchies

Block III. Architectural modeling with UML

Unit 9. Introduction to architectural modeling

Unit 10. Architectural modeling: components

Unit 11. Architectural modeling: interfaces

Unit 12. Architectural modeling: design by contracts

LEARNING ACTIVITIES AND METHODOLOGY

Theoretical-Practical Lectures: 1 ECTS

- Review of contents before class

Practical Lectures: 1 ECTS

- Exercise resolution
- General tutoring
- Partial oral expositions of the project

Team Work: 3 ECTS

- Proposal of project statement
- Project development
- Project peer review
- Proposal of questions for the theoretical exam

Individual Work: 1 ECTS

- Contribution to team project
- Individual practical exercises
- Study and preparation of final exam

ASSESSMENT SYSTEM

% end-of-term-examination:	55
% of continuous assessment (assignments, laboratory, practicals...):	45

CONTINUOUS EVALUATION (45%)

- Week 5: Activities, participation, and exercises in the classroom (Theory, 5%)
- Week 12: Exercises in groups (Practice, 10%)
- Week 13: Activities, participation and exercises in the classroom (Theory, 5%)
- Week 13: Final project (team) delivery and monthly reviews (Practice, 15%)
- Week 14: Individual practical exercises (Practice, 10%)

FINAL EVALUATION (55%)

- Final project: technical report: 25%
- Final exam: 30%

A minimum grade of 5 is required in both the final project and the final exam to pass the course. The final exam will comprise exercises related to the application of all contents of the course.

BASIC BIBLIOGRAPHY

- Alexander, I. and Stevens, R. Writing Better Requirements, Addison-Wesley, 2002
- Arlow, J. and Neustadt, I. UML and the Unified Process. Practical Object-Oriented Analysis & Design, Addison-Wesley, 2002
- Braude, E. Software Engineering. An Object-Oriented Perspective., John Wiley & Sons, 2001..
- ESA Board for Software Standardisation and Control (BSSC). ESA Software Engineering Standards., European Space Agency, February 1991..
- Fowler, M. and Scott, K. UML Distilled. A Brief Guide to the Standard Object Modeling Language, Addison-Wesley, 2004
- Larman, C. Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design and the Unified Process, Prentice Hall, 1998
- Mark Richards, Neal Ford Fundamentals of Software Architecture, O'Reilly Media, Inc, 2020
- Martin Fowler Patterns of Enterprise Application Architecture, Addison-Wesley , 2002
- Martin Fowler Refactoring, Addison-Wesley , 2018
- Pressman, Roger S. Ingeniería del software: un enfoque práctico, 6ª ed., McGraw-Hill, 2006..
- Robert Martin Clean Code: A Handbook of Agile Software Craftsmanship, Prentice Hall, 2008
- Robert Martin Clean Architecture, Prentice-Hall, 2017
- Sommerville, I. Ingeniería del Software., Pearson-Addison Wesley, 2005..
- Sommerville, I. and Sawyer, P. Requirements Engineering: A Good Practice Guide, John Wiley & Sons, 1997

- Stevens, P. and Pooley, R. Using UML, Software Engineering with Objects and Components, Addison-Wesley, 2000

- Wiegers, K. and Beaty, J. Software Requirements, Microsoft Press, 2013

ADDITIONAL BIBLIOGRAPHY

- ISO ISO/IEC/IEEE 15288:2023 Systems and software engineering System life cycle processes, ISO, 2023

- N. Ford, M. Richards, P. J. Sadalage, and Z. Dehghani Software Architecture: The Hard Parts: Modern Trade-Off Analyses for Distributed Architectures , O'Reilly, 2021

- Titus Winters, Tom Manshreck, Hyrum Wright Software Engineering at Google, O'Reilly Media, Inc., 2022

BASIC ELECTRONIC RESOURCES

- IEEE Software Special Issue . 50 Years of Software Engineering: <https://ieeexplore.ieee.org/document/8474511>

- Ian Sommerville . Software Engineering, Tenth Edition: <https://iansommerville.com/software-engineering-book/>

- Jose María Álvarez Rodríguez . Lista de lectura de DevOps: <https://learning.oreilly.com/playlists/bc1031bc-6189-42b8-85b7-789a7e59b0bd>

- Jose María Álvarez Rodríguez . Lista de lectura de Arquitectura de Software: <https://learning.oreilly.com/playlists/80425411-f3fe-4b47-8415-35cfffce0f>

- UC3M . The Software Architect Code: Building the Digital World: <https://www.edx.org/course/software-architect-code-building-digital-uc3mx-inf-1x>

- edX Platform . List of courses on Software Engineering: https://www.edx.org/course?search_query=SOFTWARE+ENGINEERING