Course guide
270015 - IES - Introduction to Software Engineering

Last modified: 30/01/2024

Unit in charge: Barcelona School of Informatics
Teaching unit: 747 - ESSI - Department of Service and Information System Engineering.

Degree: BACHELOR'S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2010). (Compulsory subject).

Academic year: 2023 **ECTS Credits:** 6.0 **Languages:** Catalan

LECTURER

Coordinating lecturer: ERNEST TENIENTE LOPEZ

Others:

Primer quadrimestre:
MARIA DOLORS COSTAL COSTA - 43
JOSE FRANCISCO CRESPO SANJUSTO - 12, 13
MONTSERRAT ESTAÑOL LAMARCA - 41, 42
MANUEL RELLO SALTOR - 11, 12, 13, 41, 42, 43
ERNEST TENIENTE LOPEZ - 11

Segon quadrimestre:
CLAUDIA PATRICIA AYALA MARTINEZ - 14, 41, 42, 43
JORDI CASANOVAS MUÑOZ - 22
MARIA DOLORS COSTAL COSTA - 31, 34
JOSE FRANCISCO CRESPO SANJUSTO - 11, 32
MONTSERRAT ESTAÑOL LAMARCA - 12, 13
SERGIO MORALES GARCIA - 41, 42
XAVIER ORIOL HILARI - 21, 22, 23, 43
MANUEL RELLO SALTOR - 11, 12, 13, 14, 31, 32, 33, 34
ERNEST TENIENTE LOPEZ - 21, 33

PRIOR SKILLS

Students are expected to be sufficiently knowledgeable about data structures, algorithms and programming to study software engineering.

REQUIREMENTS

- Prerequisite BD
- Pre-Corequisite EDA

DEGREE COMPETENCES TO WHICH THE SUBJECT CONTRIBUTES

Specific:
CT2.1. To demonstrate knowledge and capacity to apply the principles, methodologies and life cycles of software engineering.
CT2.3. To design, develop, select and evaluate computer applications, systems and services and, at the same time, ensure its reliability, security and quality in function of ethical principles and the current legislation and normative.
CT2.4. To demonstrate knowledge and capacity to apply the needed tools for storage, processing and access to the information system, even if they are web-based systems.
CT3.3. To be able to find and interpret basic information for evaluating the economic environment of the organization.
CT5.5. To use the tools of a software development environment to create and develop applications.
CT6.1. To demonstrate knowledge and capacity to manage and maintain computer systems, services and applications.
CT7.1. To demonstrate knowledge about metrics of quality and be able to use them.
CT8.1. To identify current and emerging technologies and evaluate if they are applicable, to satisfy the users needs.
CT8.2. To assume the roles and functions of the project manager and apply, in the organizations field, the techniques for managing the timing, cost, financial aspects, human resources and risk.
CT8.3. To demonstrate knowledge and be able to apply appropriate techniques for modelling and analysing different kinds of decisions.
CT8.4. To elaborate the list of technical conditions for a computers installation fulfilling all the current standards and normative.
CT8.5. To manage and solve problems and conflicts using the capacity to generate alternatives or future scenarios analysed properly, integrating the uncertainty aspects and the multiple objectives to consider.
CT8.6. To demonstrate the comprehension of the importance of the negotiation, effective working habits, leadership and communication skills in all the software development environments.
CT8.7. To control project versions and configurations.

Generical:
G4. EFFECTIVE ORAL AND WRITTEN communication: To communicate with other people knowledge, procedures, results and ideas orally and in a written way. To participate in discussions about topics related to the activity of a technical informatics engineer.

TEACHING METHODOLOGY

The subject is structured around theory and problem-solving classes.

In the theory classes the lecturer will explain the main subject content. Lecturers typically use slides that students should obtain before class.

In problem-solving classes, course content (whether presented in class or studied independently) will be studied by completing problems. This will sometimes require problems to be resolved (or at least attempted) before class, so that the best solutions can be collectively analysed and discussed in class. On other occasions, the problem will be both set and resolved in class.

LEARNING OBJECTIVES OF THE SUBJECT

1.Students should be able to provide an overview of the software engineering process
2.Students should be able to understand the requirements of a software system and relate these to the different parts of the specifications
3.Students should be able to write specifications for a UML software system
4.Students should be able to understand the desirable properties of specifications for a software system.
5.Students should be able to analyse the completeness and consistency of the specifications.
6.Students should be able to understand the general principles of software architecture and object-oriented design in UML
7.Students should be able to understand the logic structure in layers of an information system: presentation, domain and data management layers
8.Students should be able to transform a UML model into a design specification
9.Students should be able to understand the concept of design template and use some of the better known templates.
10.Students should be able to understand the basic concepts of software testing

STUDY LOAD

Type	Hours	Percentage
Hours large group	30,0	20.00
Hours medium group	30,0	20.00
Guided activities	6,0	4.00
Self study	84,0	56.00

Total learning time: 150 h

CONTENTS

Introduction to software engineering

Software requirements and software specifications

UML use case models

UML software system specification

Description:
Class diagrams, behaviour diagrams and state diagrams.

Introduction to software design

Object-oriented design with UML

ACTIVITIES

Introduction to software engineering

Specific objectives:
1

Full-or-part-time: 4h
Theory classes: 2h
Self study: 2h

Software requirements and software specifications

Specific objectives:
1, 2

Full-or-part-time: 3h
Theory classes: 1h
Self study: 2h

UML use case models

Specific objectives:
2, 3

Related competencies :
G4. EFFECTIVE ORAL AND WRITTEN communication: To communicate with other people knowledge, procedures, results and ideas orally and in a written way. To participate in discussions about topics related to the activity of a technical informatics engineer.

Full-or-part-time: 5h
Theory classes: 1h
Self study: 4h

UML data structural diagram

Specific objectives:
2, 3, 4, 5

Related competencies :
G4. EFFECTIVE ORAL AND WRITTEN communication: To communicate with other people knowledge, procedures, results and ideas orally and in a written way. To participate in discussions about topics related to the activity of a technical informatics engineer.

Full-or-part-time: 34h
Theory classes: 4h
Practical classes: 10h
Self study: 20h

UML behaviour diagram

Specific objectives:
2, 3, 4, 5

Related competencies :
G4. EFFECTIVE ORAL AND WRITTEN communication: To communicate with other people knowledge, procedures, results and ideas orally and in a written way. To participate in discussions about topics related to the activity of a technical informatics engineer.

Full-or-part-time: 28h
Theory classes: 4h
Practical classes: 8h
Self study: 16h

Introduction to software design

Specific objectives:
6, 7

Full-or-part-time: 4h
Theory classes: 2h
Self study: 2h

Software design in UML

Specific objectives:
6, 8, 9

Related competencies :
G4. EFFECTIVE ORAL AND WRITTEN communication: To communicate with other people knowledge, procedures, results and ideas orally and in a written way. To participate in discussions about topics related to the activity of a technical informatics engineer.

Full-or-part-time: 48h
Theory classes: 8h
Practical classes: 10h
Self study: 30h

UML structural model exercise

Description:
Parts of learning objectives 3 to 5 will be assessed

Specific objectives:
3, 4, 5

Related competencies :
G4. EFFECTIVE ORAL AND WRITTEN communication: To communicate with other people knowledge, procedures, results and ideas orally and in a written way. To participate in discussions about topics related to the activity of a technical informatics engineer.

Full-or-part-time: 5h
Guided activities: 2h
Self study: 3h

UML structural model test

Description:
Parts of learning objectives 1 to 5 will be assessed

Specific objectives:
1, 2, 3, 4, 5

Related competencies :
G4. EFFECTIVE ORAL AND WRITTEN communication: To communicate with other people knowledge, procedures, results and ideas orally and in a written way. To participate in discussions about topics related to the activity of a technical informatics engineer.

Full-or-part-time: 5h
Guided activities: 2h
Self study: 3h

UML behavioural model test

Description:
Parts of learning objectives 1 to 5 will be assessed

Specific objectives:
2, 3, 4, 5

Related competencies :
G4. EFFECTIVE ORAL AND WRITTEN communication: To communicate with other people knowledge, procedures, results and ideas orally and in a written way. To participate in discussions about topics related to the activity of a technical informatics engineer.

Full-or-part-time: 5h
Guided activities: 2h
Self study: 3h

Software design exercise

Description:
Part of learning objectives 8 and 9 will be assessed

Specific objectives:
8, 9

Related competencies :
G4. EFFECTIVE ORAL AND WRITTEN communication: To communicate with other people knowledge, procedures, results and ideas orally and in a written way. To participate in discussions about topics related to the activity of a technical informatics engineer.

Full-or-part-time: 5h
Guided activities: 2h
Self study: 3h

Software design test

Description:
Part of learning objectives 6 to 10 will be assessed

Specific objectives:
6, 7, 8, 9, 10

Related competencies :
G4. EFFECTIVE ORAL AND WRITTEN communication: To communicate with other people knowledge, procedures, results and ideas orally and in a written way. To participate in discussions about topics related to the activity of a technical informatics engineer.

Full-or-part-time: 4h
Guided activities: 2h
Self study: 2h

GRADING SYSTEM

The final mark will be based on five assessment activities (C1, C2, C3, FHC1, FHC2) issued throughout the course and on participation in course activities. The final mark is calculated as:

Final mark = 10% C1 + 30% FHC1 + 15% C2 + 10% C3 + 30% FHC2 + 5% participation.

taking into account that:
The control C3 is optional. If a student does not attend C3, then the weight of his/her FHC2 becomes automatically 40%
The mark of participation is achieved by delivering at least the 75% of the exercises proposed during the course and by actively participating at class.
Final Mark = NP if the student does not show to FHC2 and his/her mark according to the previous formula is
There is no final exam.

In addition to a subject mark, a generic competency mark will be awarded with the score A (excellent), B (good), C (satisfactory) , D (fail) or NA (Not evaluated).

BIBLIOGRAPHY

Basic:
- Larman, C. Applying UML and patterns: an introduction to object-oriented analysis and design and iterative development. 3rd ed. Prentice Hall PTR, 2005. ISBN 0131489062.
- Pressman, R.S.; Maxim, B.R. Software engineering: a practitioner's approach. 9th ed. New York: McGraw Hill Higher Education, 2020. ISBN 9781260548006.
- Rumbaugh, J.; Jacobson, I.; Booch, G. The unified modeling language reference manual. 2nd ed. Addison-Wesley, 2005. ISBN 0321245628.
- Professorat d'IES. Transparències d'Introducció a l'Enginyeria del Software.
- Professorat d'IES. Exercicis i exercicis resolts d'Introducció a l'Enginyeria del Software.

Complementary:
- Olivé, A. Conceptual modeling of information systems. Springer, 2007. ISBN 9783540393894.
- Olivé, A. Conceptual modeling of information systems. Berlin: Springer, 2007. ISBN 9783540393894.
- Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. Design patterns: elements of reusable object-oriented software. Addison-Wesley, 1995. ISBN 0201633612.
- Binder, R.V. Testing object-oriented systems: models, patterns, and tools. Addison-Wesley, 2000. ISBN 9780321700674.

RESOURCES

Hyperlink:
- http://www.uml.org/- http://www.omg.org/- http://hillside.net/patterns/