

## 340379 - AMEP-I4O23 - Further Software Engineering

Coordinating unit:	340 - EPSEVG - Vilanova i la Geltrú School of Engineering
Teaching unit:	723 - CS - Department of Computer Science
Academic year:	2019
Degree:	BACHELOR'S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2018). (Teaching unit Compulsory) BACHELOR'S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2010). (Teaching unit Compulsory)
ECTS credits:	6
Teaching languages:	Catalan

### Teaching staff

Coordinator: Josep M Merenciano

### Requirements

Introduction to Software Engineering

### Degree competences to which the subject contributes

Specific:

1. CEFC12. Knowledge and application of characteristics, functions and structure of data base which allow suitable use and design and analysis and implementation of applications based on them.
2. CEFC13. Knowledge and application of necessary tools for storage, processing and access to informatic systems, including the ones based on webs.
3. CEFC16. Knowledge and application of principals, methodologies, life cycles of programming engineering.
4. CEFC6. Basic knowledge and application of algorithmic processes, informatic techniques to design solutions of problems, analyzing if proposed algorisms are apt and complex.
5. CEIS1. Ability to develop, to maintain and avaluate programming services and systems which satisfy all requirements of user having a reliable and efficient behavior, being comprehensible to develop and maintain and observe to current rules, applying theory, principals, methods, practices of programming engineering.
6. CETI2. Ability to select, design, develop, integrate, value, construct, tmanage, exploit and maintain technologies of machines, programming and nets, keeping suitable costs and quality parameters.

Transversal:

7. SELF-DIRECTED LEARNING - Level 1. Completing set tasks within established deadlines. Working with recommended information sources according to the guidelines set by lecturers.
8. SELF-DIRECTED LEARNING - Level 2: Completing set tasks based on the guidelines set by lecturers. Devoting the time needed to complete each task, including personal contributions and expanding on the recommended information sources.
9. SELF-DIRECTED LEARNING. Detecting gaps in one's knowledge and overcoming them through critical self-appraisal. Choosing the best path for broadening one's knowledge.
10. EFFICIENT ORAL AND WRITTEN COMMUNICATION - Level 1. Planning oral communication, answering questions properly and writing straightforward texts that are spelt correctly and are grammatically coherent.
11. EFFICIENT ORAL AND WRITTEN COMMUNICATION - Level 2. Using strategies for preparing and giving oral presentations. Writing texts and documents whose content is coherent, well structured and free of spelling and grammatical errors.
12. EFFICIENT ORAL AND WRITTEN COMMUNICATION. Communicating verbally and in writing about learning outcomes, thought-building and decision-making. Taking part in debates about issues related to the own field of specialization.
13. THIRD LANGUAGE. Learning a third language, preferably English, to a degree of oral and written fluency that fits in with the future needs of the graduates of each course.
14. TEAMWORK - Level 1. Working in a team and making positive contributions once the aims and group and individual responsibilities have been defined. Reaching joint decisions on the strategy to be followed.

## 340379 - AMEP-I4O23 - Further Software Engineering

15. TEAMWORK - Level 2. Contributing to the consolidation of a team by planning targets and working efficiently to favor communication, task assignment and cohesion.

16. TEAMWORK. Being able to work as a team player, either as a member or as a leader. Contributing to projects pragmatically and responsibly, by reaching commitments in accordance to the resources that are available.

### Teaching methodology

A case study that conveys the course.

Material in the form of notes, summarized in the form of projections, for the study and review of content.

Exercises and problems that students must solve on their own as a means of understanding the contents of the subject.

Done exercises as a mechanism of self-evaluation.

A project in pairs as a fundamental mechanism for learning and evaluation.

Class sessions where they discussed the hard issues of content, or where it is discussed in detail how they arrived at a certain solution. The basic definitions and concepts of self are related by 80%.

Tutoring lab sessions of work done.

### Learning objectives of the subject

See catalan version

### Study load

Total learning time: 150h	Hours large group:	30h	20.00%
	Hours medium group:	0h	0.00%
	Hours small group:	30h	20.00%
	Guided activities:	0h	0.00%
	Self study:	90h	60.00%

## 340379 - AMEP-I4O23 - Further Software Engineering

### Content

<p>Application architecture</p>	<p>Learning time: 4h 10m Theory classes: 2h 30m Self study : 1h 40m</p>
<p>Description: Three-tier architecture: presentation, domain and persistence.</p> <p>Distributed architecture: distribution of layers, distribution of processes, data distribution.</p> <p>Specific objectives: Understand information systems as a set of layers interactions. Being able to work in a single layer.</p>	
<p>Responsabilites-Driven Development</p>	<p>Learning time: 185h 40m Theory classes: 35h Laboratory classes: 15h Self study : 135h 40m</p>
<p>Description: Mechanisms to move from a problem model to a solution model.</p> <p>Principles of responsibilities assignment.</p> <p>Specific objectives: Prototyping a solution from the model of the problem.</p> <p>Learn to document development. Understand the role of documentation ..</p>	

### Qualification system

See catalan version

### Regulations for carrying out activities

See catalan version

## 340379 - AMEP-I4O23 - Further Software Engineering

### Bibliography

#### Basic:

Larman, Craig. Applying UML and patterns : an introduction to object-oriented analysis and design and iterative development. 3th ed. Upper Saddle River, N.J: Prentice Hall PTR, 2005. ISBN 0131489062.

#### Complementary:

Gamma, Erich. Design patterns : elements of reusable object-oriented software. Reading, MA [etc.]: Addison-Wesley, 1995. ISBN 9780201633610.