DEGREE CURRICULUM

SOFTWARE ENGINEERING

Coordination: GIMENO ILLA, JUAN MANUEL

Academic year 2016-17

SOFTWARE ENGINEERING 2016-17

Subject's general information

<table>
<tr><td>Subject name</td><td>SOFTWARE ENGINEERING</td></tr>
<tr><td>Code</td><td>102018</td></tr>
<tr><td>Semester</td><td>1st Q(SEMESTER) CONTINUED EVALUATION</td></tr>
<tr><td>Typology</td><td>
<table>
<tr><th>Degree</th><th>Course</th><th>Typology</th><th>Modality</th></tr>
<tr><td>Double bachelor's degree: Degree in Computer Engineering and Degree in Business Administration and Management</td><td>3</td><td>COMPULSORY</td><td>Attendance-based</td></tr>
<tr><td>Bachelor's Degree in Computer Engineering</td><td>3</td><td>COMPULSORY</td><td>Attendance-based</td></tr>
<tr><td>Master's Degree in Informatics Engineering</td><td></td><td>COMPLEMENTARY TRAINING</td><td>Attendance-based</td></tr>
</table>
</td></tr>
<tr><td>ECTS credits</td><td>6</td></tr>
<tr><td>Groups</td><td>1GG,2GM</td></tr>
<tr><td>Theoretical credits</td><td>3</td></tr>
<tr><td>Practical credits</td><td>3</td></tr>
<tr><td>Coordination</td><td>GIMENO ILLA, JUAN MANUEL</td></tr>
<tr><td>Department</td><td>INFORMATICA I ENGINYERIA INDUSTRIAL</td></tr>
<tr><td>Teaching load distribution between lectures and independent student work</td><td>40% Presential (equivalent to 60h)
60% Autonomous work (equivalent to 90h)</td></tr>
<tr><td>Important information on data processing</td><td>Consult this link for more information.</td></tr>
<tr><td>Language</td><td>Preferably Catalan (Spanish if any student shows dificulties with Catalan).</td></tr>
<tr><td>Distribution of credits</td><td>Juan Manuel Gimeno Illa 4.5
Montserrat Sendin Veloso 4.5</td></tr>
<tr><td>Office and hour of attention</td><td>Juan Manuel Gimeno (3.20 EPS wednesday at 1pm; others by appointment)
Monserrat Sendín (3.20 EPS by appointment)</td></tr>
</table>

SOFTWARE ENGINEERING 2016-17

Professor/a (s/es)	Adreça electrònica professor/a (s/es)	Crèdits	Horari de tutoria/lloc
GIMENO ILLA, JUAN MANUEL	jmgimeno@diei.udl.cat	4,5	
SENDÍN VELOSO, MONTSERRAT	msendin@diei.udl.cat	4,5	

Subject's extra information

Compulsory subject of 3rd year (1st quarter) that belongs to the common studies in the computer science branch.

Matter: Analysis and Design of Applications.

RECOMMENDATIONS: We assume the student knows the concepts about object-oriented programming and data structures teached in Programming II and Data Structures.

Learning objectives

- Knowing the conceptual basis and the different aspects of the discipline, among other the software lifecycle process model
- Apply the Use Case technique
- Specifying in a textual way the functional and non functional needs for a certain software system planned by means of a statement and/or other inputs from the user
- Developing the classes diagram for a certain software system following the Object Oriented Modeling principles
- Be familiar with a UML-based modeling tool
- Understanding the concept of code as a something that evolves over time
- Be able to program basic unit tests
- Understanding the object oriented design fundamental principles
- Recognizing the concept of responsibility as a fundamental one when planning an object oriented design

Competences

Cross-disciplinary competences

- **EPS-11:** Capacity to understand the needs of the user expressed in a no technical language

Specific competences

- **GII-CRI2:** Capacity to plan, conceive, deploy and direct projects, services and computer systems in all the fields, leading his set up and his continuous improvement and evaluation his economic and social impact
- **GII-CRI12:** Knowledge and application of the characteristics, functionalities and structure of the databases, that allow their suitable use, and the design and the analysis and implementation of applications based in them
- **GII-CRI13:** Knowledge and application of the necessary tools for the storage, processing and access to the Systems of information, including those based in web
- **GII-CRI16:** Knowledge and application of the principles, methodologies and life cycle of the software engineering
- **GII-CRI17:** Capacity to design and evaluate person-computer interfaces that guarantee the accessibility and

usability of systems, services and computer applications.

Subject contents

Theme I - *Introductory aspects*

1.1. Initial qüestions about the Software Engineering

1.2. A little of history

1.3. Software development process

1.4. Software process models

1.5. Conclusions

Theme II - *Requirements Analisys*

2.1. Requirements specification

2.2. The Use Cases technique

2.3. A step more in the specification: the System Sequence Diagram

2.4. Conclusions

Theme III - *Domain Analisys*

3.1. Analisys Classes Diagram

3.2. A step more in the domain analysis: the contracts of the operations

3.3. Conclusions

Theme IV - *Introduction to design and unit testing*

4.1. Code as something which evolves

4.2. The JUnit framework

Tema V - *The SOLID principles*

5.1. Single responsibiliti principle

5.2. Open-closed principle

5.3. Liskov substitution principle

5.4. Interface segregation principle

5.5. Dependency inversion principle

Tema VI - *Responsibility based design*

6.1. The concept of responsibility

6.2. The GRASP patterns of responsibility assignement

Methodology

Big-size Groups: Masterly Classes (3 credits)

- Theorical part: Supported by snapshots and/or specific notes
- Practival application part: Always working over examples.A **problems collection** is available. In class concrete problems are being solved. The solutions are being delivered along the semester

Medium-size Groups: Laboratory Classes (3 credits)

- Guided classes and personalized monitoring
- UML Modeling tool usage: **ArgoUML** and/or **Visual Paradigm**
- Progressive work regarding a certain **practical statement**, which will simulate the software project development

Autonomous work (non presential):

- Practical work will be completed during **no presential** hours
- **Highly recommended** to the student: the problem solving from the **problems collection**, in order to get feedback from the teacher

The **avaluation system** (detailed in el corresponding section) is composed of: 1) writen tests (l2 partial exams); and 2) practices (to develop individually or in groups depending on each case).

Development plan

Week	Theory (GG)	Laboratory (LG)	Autonomous Work
1	Subject presentation T1: Introductory aspects	T1: Introductory aspects	Study
2	T1: Introductory aspects	T1: Introductory aspects	Study
3	T2: Requirements analysis Requirements specification	Requirements analysis application	Study and problems solving (Problems collection)
4	T2: Requirements analysis. The Use Cases technique. Problems	UML Modeling usage Use Cases technique practical application	Study and problems solving (Problems collection)
5	T2: Requirements analysis System Sequence Diagrams Problems	Use Cases technique practical application	
6	T3: Domain analysis Object Oriented Modeling technique	UML Modeling usage SSD practical application	Study, problems solving (Problems collection) and practice defelopment

7	T3: Domain analysis Problems	UML Modeling usage Domain Model practical application	Study, problems solving (Problems collection) and practice defelopment
8	T3: Domain analysis Contracts of operations	Contracts practical application	Study, problems solving (Problems collection) and practice defelopment
9	First midterm		
10	T4: Introduction to Design Test concept	Git usage	Study
11	T4: Junit Substitute objects	Git usage	Study and problems solving
12	T5: Principis SOLID Intro, OCP & LSP	Simple testing problems	Study and testing practical
13	T5: Principis SOLID SRP, ISP & DIP	T6: GRASP Patrons Responsability concept	Study, problems solving and testing practical
14	T6: Patrons GRASP Expert, creador, baix acoblament	Testing with substitutions	Study, problems solving and testing practical
15	T6: GRASP Patrons High cohesion, controller	Testing with substitutions	Study, problems solving and testing practical
16	Second midterm		
17	Second midterm		
18	Tutorization		
19	Recovery		

Evaluation

Activt.	Description	Weight	Mínimum Grade	In group	Presential	Mandatory	Recoverable
Part1	First midterm Basic concepts	25%	3,0	No	Yes	Yes	Sí
Part2	Second midterm Basic concepts	25%	3,0	No	Yes	Yes	Sí
Actv1	Requirements Analysis	15%	No	Sí	No	Yes	No
Actv2	Domain Model and Contracts	15%	No	Yes	No	Yes	No
Actv3	Unitary testing	20%	No	No	No	Yes	No

Final grade = 0,25 * Part1 + 0,25 * Part2 + 0,15 * Actv1 + 0,15 * Actv2 + 0,20 * Actv3

- Subject is passed if **final grade** is greater or equal than **5,0** and all midterms arr above the minimum required.

Other considerations:

- Type of exams: concept fixation and problems solving
- For all activities: programmed deliveries, unmovable dates

Bibliography

Basic bibliography

- C. Larman, Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. Prentice-Hall, 2005 (3ª ed.)
- Robert C. Martin: Agile Software Development: Principles, Patterns, and Practices, Prentice-Hall, 2002.

Complementary bibliography

- G. Kotonya, I. Sommerville: Requirements Engineering: Processes and Techniques. Wiley, 1998
- P.Tahchiev et al.: Junit in Action (2nd edition). Manning, 2011.