340376 - INEP-I3O23 - Introduction to Software Engineering

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

Teaching staff
Coordinator: Josep M. Merenciano
Others: Josep M. Merenciano

Requirements
Programming methodologies

Degree competences to which the subject contributes

Specific:
3. CEFC1. Ability to design, develop, select and value applications and informatic systems affirming its reliability, security and quality corresponding to ethical principals and legislation and current rules.
6. CEFC2. Ability to plan, conceive, develop, manage informatic projects, services and systems in all areas, leading their implementation and continuous improvement assassing their economic and social repercussions.

2. CEFC6. Basic knowledge and application of algorithmic processes, informatic techniques to design solutions of problems, analyzing if proposed algorisms are apt and complex.
8. CEFC8. Ability to analyze, to design, to construct and to maintain applications in a well built, secure and efficient way choosing the most adequated paradigms and languages.
10. CETI5. Ability to select, to develop, integrate and manage information systems which satisfy organization necessities with indentified costs and quality criteria.
11. CEFC3. Ability to understand the importance of negotiation, effective work habits, leadership and communication skills in all environments for software development.

9. 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.
12. 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.
16. 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.
18. TEAMWORK - Level 2. Contributing to the consolidation of a team by planning targets and working efficiently to favor communication, task assignment and cohesion.
20. 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.
Understanding the different stages of development, and the different objectives of each stage.

Being able to model a problem and its solutions. Learn to document a development.

Being able to build up a complete specification of small size problem in a environment similar to the professional one: excess of requirements, lack of time, teamwork, telecommuting.

Know how to manage contextual pressure, and therefore know to plan, prioritize and simplify.

### Study load

<table>
<thead>
<tr>
<th><strong>Total learning time:</strong> 150h</th>
<th>Hours large group: 30h</th>
<th>20.00%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hours medium group:</td>
<td>0h</td>
<td>0.00%</td>
</tr>
<tr>
<td>Hours small group:</td>
<td>30h</td>
<td>20.00%</td>
</tr>
<tr>
<td>Guided activities:</td>
<td>0h</td>
<td>0.00%</td>
</tr>
<tr>
<td>Self study:</td>
<td>90h</td>
<td>60.00%</td>
</tr>
</tbody>
</table>
### Phases in the development of a software project

**Description:**
Generic stages should have a development: analysis, specification, design and implementation.

**Related activities:**
- Project

**Specific objectives:**
Understanding the various needs and objectives of each stage. Learn to jump from one stage to another. Distinguish when you are talking about a problem and when you are talking about a possible solution to this.

**Learning time:** 7h 30m
- Theory classes: 4h
- Self study: 3h 30m

### Modeling problems

**Description:**

**Related activities:**
- Project

**Specific objectives:**
Learn modeling problems: namely, to explain in a concise, specific and unambiguous what the problem is to solved.

**Learning time:** 125h 30m
- Theory classes: 20h
- Laboratory classes: 28h
- Guided activities: 66h 40m
- Self study: 10h 50m
### Modeling software solutions

**Description:**
Models of the solution: visibility, responsibility.

**Related activities:**
Build static model form a given dynamic model

**Specific objectives:**
Understand the difference between the model of the problem and the model of one possible solutions: similarities and differences.

- Being able to relegate every decision in the correct domain (problem or solution).
- Being able to build static solution model from dynamic solution model

<table>
<thead>
<tr>
<th>Learning time: 10h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory classes: 4h</td>
</tr>
<tr>
<td>Laboratory classes: 2h</td>
</tr>
<tr>
<td>Self study: 4h</td>
</tr>
</tbody>
</table>

### Introduction to persistence and data bases

**Description:**
See catalan version

**Related activities:**
See catalan version.

**Specific objectives:**
See catalan version

<table>
<thead>
<tr>
<th>Learning time: 7h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory classes: 1h</td>
</tr>
<tr>
<td>Self study: 6h</td>
</tr>
</tbody>
</table>

### Qualification system

See catalan version

### Regulations for carrying out activities

See castalan version
340376 - INEP-I3O23 - Introduction to Software Engineering

**Bibliography**

**Basic:**


