270084 - GPS - Software Project Management

Coordinating unit: 270 - FIB - Barcelona School of Informatics
Teaching unit: 747 - ESSI - Department of Service and Information System Engineering
Academic year: 2019
Degree: BACHELOR'S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2010). (Teaching unit Optional)
ECTS credits: 6
Teaching languages: Catalan

Teaching staff

Coordinator: - Dolors Costal Costa (dolors@essi.upc.edu)
Others: - Albert Renom Vilaró (albert.renom@fib.upc.edu)
- Enric Mayol Sarroca (mayol@essi.upc.edu)
- Raimon Lapuente Ferran (raimon.lapuente@upc.edu)
- Xavier Franch Gutiérrez (franch@essi.upc.edu)

Prior skills

Students should have prior knowledge of programming and software engineering fundamentals.

Requirements

- Prerequisite EEE
- Prerequisite IES

Degree competences to which the subject contributes

Specific:

CES1.3. To identify, evaluate and manage potential risks related to software building which could arise.

CES1.7. To control the quality and design tests in the software production

CES2.1. To define and manage the requirements of a software system.

CES2.2. To design adequate solutions in one or more application domains, using software engineering methods which integrate ethical, social, legal and economical aspects.

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.

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.

CT8.6. 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:

G1. ENTREPRENEURSHIP AND INNOVATION: to know and understand the organization of a company and the sciences which govern its activity; capacity to understand the labour rules and the relation between planning, industrial and business strategies, quality and benefit. To develop creativity, entrepreneur spirit and innovation
270084 - GPS - Software Project Management

tendency.
G2. SUSTAINABILITY AND SOCIAL COMPROMISE: to know and understand the complexity of the economic and social phenomena typical of the welfare society. To be capable of analyse and evaluate the social and environmental impact.

Teaching methodology

Theory classes:
* The teacher will make a statement of issues on the subject, using audiovisual material (slides and videos) to complement the explanation.
* Teachers will discuss with students about issues raised in previous classes and readings of cases proposed.

Lab classes.
Work with project-based learning in projects that will be simulations of real activities and interpretation of roles.

Learning objectives of the subject

1. Overview: Have an overview of what Software Projects are, and what are the complexities associated with their management.
2. Types of projects: Knowing the different types of software projects, what are their characteristics, advantages and disadvantages associated. Have criteria to identify what type of project is going to be managed.
3. Factors: Knowing what factors affect the cost structure in a Software Project. Knowing what are the risk factors and the viability of a project. Being able to produce a budget.
4. Estimation: Being able to estimate the resources required to perform an activity belonging to the software development process.
5. Plan: Being able to develop the plan of a Software Project.
6. Priorization: Being able to prioritize the requirements of a Software Project in order to maximize the value they provide to its stakeholders.
7. Methods: Understand different methods of Software Project management. Knowing their advantages and disadvantages. Have criteria for selecting a method based on the project type, equipment and other contextual factors.
8. Management: Being able to perform the management and monitoring of a Software Project.
9. Project leader: Understand the role of a project leader as a human team leader and the different ways to manage teams.
10. Quality: Knowing what are the models of quality management in Software Project management, their applicability and in which cases are relevant.
11. Testing: Being able to design and implement a strategy of testing a Software Project.
12. Tools: Knowing and using software tools to support Software Project management
13. Social and environmental aspects: Being able to consider the social and environmental aspects in Software Projects.

Study load

<table>
<thead>
<tr>
<th>Total learning time: 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>6h</td>
<td>4.00%</td>
</tr>
<tr>
<td>Self study:</td>
<td>84h</td>
<td>56.00%</td>
</tr>
</tbody>
</table>
Introduction to Software Projects

**Degree competences to which the content contributes:**

**Description:**
Introduces the problem of managing projects, stakeholders, the complexities associated and the tasks involved; the project context, project types, project life cycle, software construction activities (management, requirements, analysis, software building, testing, quality, maintenance, reengineering), etc.

Classic Software Project management

**Degree competences to which the content contributes:**

**Description:**
The activities, roles and methodologies of classic Software Project management are explained. These are based on developing a predictive project plan that will guide the software development process. Rational Unified Process will be used as an example of such a methodology.

Agile Software Project management

**Degree competences to which the content contributes:**

**Description:**
The agile approach to Software Project management is explained. It is based on the Manifesto for Agile Software Development, that proposes an adaptive approach focused on value contributed to the project stakeholders and people involved in it. Scrum, XP and Kanban are used as examples of agile methodologies.

Other Software Project contexts

**Degree competences to which the content contributes:**

**Description:**
There are contexts that are quite specific and require particular considerations and techniques. We will discuss the following cases: open source; start-ups (lean); call for tenders; outsourcing and offshoring.
### 270084 - GPS - Software Project Management

#### Planning of activities

<table>
<thead>
<tr>
<th>Activity</th>
<th>Hours</th>
<th>Description</th>
<th>Specific objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Study of Introduction to Software Projects</strong></td>
<td>6h</td>
<td>The student will participate in the classroom raising questions, giving opinions and discussing the topics proposed for discussion</td>
<td>3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14</td>
</tr>
<tr>
<td><strong>Study of classic Software Project management</strong></td>
<td>15h</td>
<td>The student will participate in the classroom raising questions, giving opinions and discussing the topics proposed for discussion</td>
<td>3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14</td>
</tr>
<tr>
<td><strong>Study of agile Project management</strong></td>
<td>18h</td>
<td>The teacher will participate in the classroom raising questions, giving opinions and discussing the topics proposed for discussion</td>
<td>3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14</td>
</tr>
<tr>
<td><strong>Study of other Software Project contexts</strong></td>
<td>5h</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Project of classic Software Project management

**Description:**
The student will participate in the classroom raising questions, giving opinions and discussing the topics proposed for discussion.

**Specific objectives:**
3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14

<table>
<thead>
<tr>
<th>Hours</th>
<th>Theory classes: 0h</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Practical classes: 0h</td>
</tr>
<tr>
<td></td>
<td>Laboratory classes: 14h</td>
</tr>
<tr>
<td></td>
<td>Guided activities: 0h</td>
</tr>
<tr>
<td></td>
<td>Self study: 34h</td>
</tr>
</tbody>
</table>

### Project of agile Software Project management

**Description:**
First, the student will be part of a team (formed by 4-5 students) and will inform the teacher about the composition. The student will carry out the project with his/her team colleagues and will deliver to the teacher.

**Specific objectives:**
3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14

<table>
<thead>
<tr>
<th>Hours</th>
<th>Theory classes: 0h</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Practical classes: 0h</td>
</tr>
<tr>
<td></td>
<td>Laboratory classes: 16h</td>
</tr>
<tr>
<td></td>
<td>Guided activities: 0h</td>
</tr>
<tr>
<td></td>
<td>Self study: 35h</td>
</tr>
</tbody>
</table>

### Delivery of the project on classic Software Project management

**Description:**
The team of students deliver the project at due time. The teacher evaluates it according to the established criteria.

**Specific objectives:**
3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14

<table>
<thead>
<tr>
<th>Hours</th>
<th>Guided activities: 0h</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Self study: 0h</td>
</tr>
</tbody>
</table>
### Delivery of the project on agile Software Project management

<table>
<thead>
<tr>
<th>Description:</th>
<th>Hours: 0h</th>
</tr>
</thead>
</table>
| The team of students deliver the project at due time. The teacher evaluates it according to the established criteria. | Guided activities: 0h  
Self study: 0h |

<table>
<thead>
<tr>
<th>Specific objectives:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14</td>
<td></td>
</tr>
</tbody>
</table>

### Course exam

<table>
<thead>
<tr>
<th>Description:</th>
<th>Hours: 7h</th>
</tr>
</thead>
</table>
| Exam in the last course’s hour with the aim of validating the knowledge taught along the course | Guided activities: 1h  
Self study: 6h |

<table>
<thead>
<tr>
<th>Specific objectives:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14</td>
<td></td>
</tr>
</tbody>
</table>

### Qualification system

**TO-DO**
Bibliography

Basic:


Complementary:


Others resources:

Hyperlink

https://www.scrum.org/scrum-guide

http://agilemanifesto.org