Software Development and Quality	2014/2015

Code: 43337
ECTS Credits: 6

Degree	Type	Year	Semester
4314660 Computer Engineering	OB	1	1

Contact

Name: Rodolfo Alberto Guichon Aguilar

Email: Rodolfo.Guichon@uab.cat

Use of languages

Principal working language: anglès (eng)

Some groups entirely in English: No

Some groups entirely in Catalan: Yes

Some groups entirely in Spanish: No

Prerequisites

There is no prerequisites for this subject.

Objectives and Contextualisation

Introduce the student into different aspects related to utilization of practices and tools that increase software development quality as also management of these tools. Knowlegde on planning techniques to manage these practices. Activities that allow development quality enhancing. Mesurement of the impact of these activities on the developed product. Different metodologies of software engineering that allow high quality development of software. Recommendations on the organization of the team and the mesurement of the quality of processes.

Skills

- Analyse the information needs of a particular environment and carry out all stages in the construction of an information system.
- Apply mathematical, statistical and artificial-intelligence methods to model, design and develop applications, services, intelligent systems and knowledge-based systems.
- Apply the principles of economy and management of human resources and projects, along with legislation, regulations and standards in computer science.
- Communicate and justify conclusions clearly and unambiguously to both specialised and non-specialised audiences.
- Communicate orally and in writing in English.
- Continue the learning process, to a large extent autonomously
- Direct work on computer systems, complying with current rules and guidelines and safeguarding the quality of the service.
- Display a capacity for the preparation, strategic planning, coordination and technical and financial management of projects in all areas of computer engineering, applying criteria of quality and environmental sustainability.
- Display a spirit of enterprise and innovation and a wide-ranging vision in the search for new areas to explore in a specific field of the computer engineering profession.
- Integrate knowledge and use it to make judgements in complex situations, with incomplete information, while keeping in mind social and ethical responsibilities.
- Oversee, manage, audit and certify the quality of computing developments, processes, systems, services, applications and products.

- Propose, calculate and design products, processes and installations in all areas of computer engineering.
- Responsibly manage information and knowledge when leading multidisciplinary groups and/or projects.
- Solve problems in new or little-known situations within broader (or multidisciplinary) contexts related to the field of study.
- Understand and apply ethical responsibility, legislation and codes of practice to professional activity in computer engineering.

Learning outcomes

1. Analyse the needs and impact of software engineering methodologies to solve the problem addressed in the development of high-quality software.
2. Apply the principles of economy and management of human resources and projects, along with legislation, regulations and standards in computer science.
3. Communicate and justify conclusions clearly and unambiguously to both specialised and non-specialised audiences.
4. Communicate orally and in writing in English.
5. Conceptualise alternative high-quality software solutions for engineering problems and create prototypes to show the validity of the system proposed.
6. Continue the learning process, to a large extent autonomously
7. Direct work on computer systems, complying with current rules and guidelines and safeguarding the quality of the service.
8. Display a capacity for the preparation, strategic planning, coordination and technical and financial management of projects in all areas of computer engineering, applying criteria of quality and environmental sustainability.
9. Display a spirit of enterprise and innovation and a wide-ranging vision in the search for new areas to explore in a specific field of the computer engineering profession.
10. Identify the basic problems to be solved in the development of high-quality software and the most appropriate techniques for organising development teams and measuring the quality of processes.
11. Identify the best way to organise teams and measure quality in order to solve a software development problem.
12. Integrate knowledge and use it to make judgements in complex situations, with incomplete information, while keeping in mind social and ethical responsibilities.
13. Propose, calculate and design products, processes and installations in all areas of computer engineering.
14. Responsibly manage information and knowledge when leading multidisciplinary groups and/or projects.
15. Solve problems in new or little-known situations within broader (or multidisciplinary) contexts related to the field of study.
16. Understand and apply ethical responsibility, legislation and codes of practice to professional activity in computer engineering.
17. Use quality-management techniques to oversee, manage, audit and certify a specific problem.

Content

1. Quality management principles

2. Planification

3. Inspections

4. Reviews and auditories

5. Error management

6. Improvement and metrics

7. Agile development

8. Organizational quality and CMMI

Methodology

• Lectures

• Problems based learning

• Oral presentation

• Discussions

• Complementary activities

• Work preparation

• Personal study

Activities

Title	Hours	ECTS	Learning outcomes
Type: Directed			
Theory, follow up and control	42	1.68	1, 4, 5, 9, 14, 11, 10, 12, 15, 3, 6, 16, 7, 8, 13, 2, 17
Type: Supervised			
Tutoring	15	0.6	4
Type: Autonomous			
Project	90	3.6	1, 4, 5, 9, 14, 11, 10, 12, 15, 3, 6, 16, 7, 8, 13, 2, 17

Evaluation

1. Exam: 20%

2. Project: 70% :

- Oral defense (10%)
- Active participation (10%)
- Self evaluation (10%)
- Report delivery (40%)

3. Attendance and participation: 10%

Evaluation activities

Title	Weighting	Hours	ECTS	Learning outcomes
Attendance and participation	10	0	0	4

Exam	20	3	0.12	1, 5, 11, 10, 12, 15, 3, 6, 16, 7, 8, 13, 2, 17
Project	70	0	0	1, 4, 5, 9, 14, 11, 10, 12, 15, 3, 6, 16, 7, 8, 13, 2, 17

Bibliography

1. Daniel Galin, Software Quality Assurance: From theory to implementation, Pearson, 2004.

2. Ken Schwaber and Jeff Sutherland, The scrum guide, Scrum.org, 2013.

2. Michael James, Scrum Reference card, http://scrumreferencecard.com/ScrumReferenceCard.pdf, 2012.

3. Jeff Sutherland, Scrum Handbook, Scrum Training Institute Press, 2010.

4. CMMI for Development v1.3, Software Engineering Institute, 2010.

5. Mary Beth Chrissis, Mike Konrad, Sandra Shrum,CMMI for Development 3rd edition, Adisson-Wesley, 2011

6. James Shore, Crromatic, The art fo Agile Development, O'really media, 2007