



Universidad  
de Alcalá



Universidad  
de Alcalá

# TEACHING GUIDE

## Software Engineering

**Bachelor's Degree in  
Computer Engineering  
Computer Science  
Information Systems Engineering**

**University of Alcalá**

**Academic year: 2019-2020**

2nd Year – 2nd Term

## Table of contents

Module:	<b>Software Engineering</b>
Code:	<b>780015</b>
Degrees:	<b>Degree in Computer Engineering Degree in Computer Science Degree in Information Systems Engineering</b>
Department and knowledge area:	<b>Computer Science</b>
Compulsory/Optional:	<b>Obligatory</b>
ECTS:	<b>6</b>
Year and term:	<b>2nd year and 2nd term</b>
Academics:	<b>Check University's website</b>
Office Hours / Tutorships:	<b>Consult with academics</b>
Languages:	<b>English and Spanish</b>

## 1. PRESENTATION

This subject introduces the principles of sound software engineering, describing the complete software development life-cycle from feasibility study to ongoing maintenance, together with modern methods for managing the process.

Students will become aware of the differences between developing software as part of an academic course and developing software for an employer. The potential of various tools and techniques to assist in the development of on-time, within-budget software will be discussed.

### Pre-requisites and recommendations

It is recommended to have carried out *Introduction to Programming* and *Data Structures* units.

## 2. SKILLS

### **General skills:**

CG5 Ability to conceive, develop and maintain systems, services and applications using the methods of software engineering as a tool for quality assurance, according

to the knowledge acquired as provided in paragraph 5 of resolution BOE-A- 2009-12977.

CG8 Knowledge of basic materials and technologies that enable learning and development of new methods and technologies, as well as to equip them with great versatility to adapt to new situations.

CG9 Ability to solve problems with initiative, decision making, autonomy and creativity. Ability to communicate and transmit knowledge and skills of the profession of Technical Engineer.

CG12 Knowledge and application of basic elements of economics and human resource management, organization and project planning, as well as legislation, regulation and standardization in the field of IT projects, according to the knowledge acquired as provided in paragraph 5 resolution BOE-a-2009-12977.

#### **Specific skills:**

CI3 Ability to understand the importance of negotiation, effective work habits, leadership and communication skills in all software development environments.

CI4 Ability to develop the technical specifications of a computer facility that meets the standards and regulations.

CI16 Knowledge and application of the principles, methodologies and life cycles of software engineering.

CI17 Ability to design and evaluate human-computer interfaces that guarantee accessibility and usability of systems, services and applications.

#### **Learning outcomes:**

The expected learning outcomes, determined from the specific competences included in the verified report of the degree as specific competences, are the following:

- RA1. Integrate the necessary skills to perform the analysis, modelling, definition and resolution of problems in the field of Information Systems.
- RA2. Apply knowledge and skills in the management of commercial products and tools for the implementation of Information Systems.
- RA3. Know how to collect and structure information for the preparation of requirements and specifications.
- RA4. Know how to analyse the organizational systems, perform a logical design to improve them and develop and analyse alternatives that involve the implementation of packages, their customization, the construction of software or the use of CASE tools.
- RA5. Identify and know how to apply mechanisms for the rapid development of information systems, such as prototypes.
- RA6. Develop effective personal communication skills, group work and relationship with users using both classical techniques and computer tools.
- RA7. Present and use complexity and quality metrics to estimate and assess the software to be developed and maintained.

### **3. CONTENTS**

UNITs	Credits or Hours
Part 1 Introduction to Software Engineering	8
Part 2 Requirements and Analysis	8
Part 3 Design and implementation	8
Part 4 Software Testing	8
Part 5 Software Maintenance	8
Part 6 Software Processes and management	8
Part 7. Software Quality and Metrics	8

Content blocks	Lab assignments
Analysis and design	Basic UML models including Use cases, high level class diagrams, sequence and communication
Testing and maintenance	Software testing, metrics and maintenance

## 4. LEARNING METHODOLOGIES

### 4.1. Credit distribution

Tuition hours:	56 hours + 4 hours of assessment
Self-study hours:	90 hours
Total	150

### 4.2. Methodological strategies, materials and teaching

Software Engineering is part of the second year and second semester and composed of 6 ECTS (150 hours).

Learning activities include:

- Theory classes.
- Practical classes: problem solving.
- Laboratory classes
- Office hours or tutorship: individual or collective

In addition, it is possible to carry out the following works:

- Practical classes: problem solving.
- Laboratory classes
- Individual or team assignments.
- Demonstrations of assignments
- Seminars.
- On-line activities using a learning management systems such as Blackboard

#### Resources

- Bibliography
- IDEs (Integrated Development Environments) and other tools
- Computers
- Learning Management Systems
- Projectors

## 5. EVALUATION

Evaluation is regulated by the official ETCS credit system (RD 1125/2003). Students also will comply with university regulation approved on 24 March 2011 (art. 9 and 10).

Evaluation of competences will consider student attitude and interest. Students can choose between continuous evaluation with three tests during the term or a final exam.

#### Evaluation criteria:

The dimension and issues that will be assessed in the learning correspond to the acquisition of skills presented in the guide. The following general evaluation criteria will also be considered for this purpose:

Regarding the papers to be presented:

- Completeness of the work, i.e. it must describe all the sections described.
- Correct writing, respecting the rules of academic writing in terms of paragraph organization, spelling, presentation of tables and figures.
- Adaptation of format and design.
- Use of appropriate specific terminology.
- Clarity and precision of the language used.
- Use of references to support the argumentation.
- Adequacy of the bibliography used.
- Source code of the software correctly commented.

With respect to the exposition of the assignments:

- Explanatory and expressive clarity.
- Spontaneity and verbal fluency.
- Adaptation to the time foreseen for the exhibition.
- Use of the time used.
- Appropriate use of support material.
- Security in the answers.
- Adequate arguments based on the work presented.

In relation to the attitude in face-to-face classes:

- Respect for the teacher and other classmates.
- Participation in group activities.

- Collaboration in the development of practical cases.
- Interest shown in face-to-face sessions.

The following evaluation criteria specific to the subject are established:

- CE1 The student is able to understand the basic terminology related to Software Engineering, its history and origin.
- CE2 The student values the importance of documentation of computer systems and consider it as an integral part of the software.
- CE3 The student knows how to apply requirements engineering and understands its usefulness as an initial step in the analysis and design of the software.
- CE4 The student knows the objectives, structure and diagrams that compose the modelling of systems with UML.
- CE5. The student applies correctly analysis and structured design techniques, both for data and for processes, and knows how to use UML diagrams in the analysis and design of object-oriented applications.
- CE6 The student knows the importance of consistency between the different elements obtained in the analysis and in the design (data and processes) and between the results and inputs of both processes.
- CE7 The student is able to execute the verification and validation activities in a rigorous way throughout the software development and maintenance process.
- CE8 The student has acquired the concept of test case and its importance in maintenance tasks.
- CE9. The student knows the main existing methods based on the test techniques and the benefits and disadvantages of using each of them.
- CE10 The student understands and is able to explain the different software maintenance techniques.
- CE11. The student has assimilated the responsibility of the software engineers, assuming their ethical code.
- CE12. The student understands the quality process and its importance in software development.

## Evaluation instruments and marking

### 1) Continuous Evaluation

Continuous Assessment Tests (CATs or PEI) will be carried out during theory classes. Laboratory tests (Continuous Assessment Laboratory –CALs or PL–) will be assignments that might be demonstrated. Also further evidences can be collected from individual works or laboratory sessions. The following tables show the tests, assignments and percentages of the marks for the continuous evaluation.

#### Evaluation Instruments

- PEI1: Theoretical-practical knowledge test on topics 1 and 2.
- PEI2: Theoretical-practical knowledge test on topics 3, 4 and 5.
- PEI3: Theoretical-practical knowledge test on topics 6 and 7.
- PL1: Group or individual work on laboratory practices (CAL1).
- PL2: Group or individual work on laboratory practices (CAL2).

Competencies		Learning Outcome	Evaluation Criteria	Evaluation Instrument	Marks Percentage
CG5,CG8, CG9,CG12	CI3, CI4	RA3, RA4	CE1, CE3	PEI1	15
CG5,CG8, CG9,CG12	CI17	RA1, RA4, RA5, RA7	CE4, CE8, CE10, CE9	PEI2	25
CG5,CG8, CG9,CG12	CI16	RA1, RA4	CE11 CE12	PEI3	10
CG5,CG8, CG9,CG12	CI3, CI4	RA1, RA2, RA4, RA6	CE2, CE3, CE5, CE6	PL1	25
CG5,CG8, CG9,CG12	CI16, CI17	RA1, RA2, RA6, RA7	CE2, CE7, CE8	PL2	25

## 2) Final evaluation:

The following evaluation tools will be used:

- PEF: theoretical-practical knowledge test on parts 1, 2, 3, 4, 5, 6 and 7.
- PL1: Group or individual work on laboratory practices (requirements analysis and design).
- PL2: Group or individual work on laboratory practices: Testing, metrics and Maintenance.

The examination of the theory part will be carried out according to the calendar established by the Polytechnic School. It will consist of a series of questions to be developed by the student, exercises and/or multiple choice questions.

The evaluation of the laboratory will consist of two practical works that the student will have to deliver and defend before the professor of the matter in a session specifically destined for it.

The following table indicates the weight in the grade (between 0 and 100) of each test, and its relation with the evaluation criteria, learning results and general competences:

Competencies		Learning Outcome	Evaluation Criteria	Evaluation Instrument	Marks Percentage
CG5,CG8, CG9,CG12	CI3, CI4 CI17 CI16	RA3, RA4 RA1, RA4, RA5, RA7 RA1, RA4	CE1, CE3 CE4, CE8, CE10, CE9 CE11 CE12	PEF	50
CG5,CG8, CG9,CG12	CI3, CI4	RA1, RA2, RA4, RA6	CE2, CE3, CE5, CE6	PL1	25
CG5,CG8, CG9,CG12	CI16, CI17	RA1, RA2, RA6, RA7	CE2, CE7, CE8	PL2	25

## Results:

Same as the final evaluation.

Failing individual tests/assignment does not imply to fail the whole unit.

Evaluation of competences include

- Knowledge of contents and concepts

- Problem solving skills
- Practical application of theoretical concepts
- Interest and attitude while carrying out the work

Those students not opting for continuous evaluation, there will be a final exam covering all materials and a single assignment also covering all practical parts.

## 6. BIBLIOGRAFY

- Sommerville, I., *Software Engineering, 10th Edt*, Addison Wesley, 2015
- Pfleeger, S.L., Atlee, J., *Software Engineering: Theory and Practice*, 4 Edt, Prentice Hall, 2010
- van Vliet, H., *Software Engineering: Principles and Practice*, 3rd Edt., Wiley, 2008
- Fenton, N.E., Bieman, J., *Software Metrics: A Rigorous and Practical Approach (3rd Edt.)*, PWS, 2015
- Pressman, R.S., Maxim, B., *Software Engineering: a practitioner's approach*, 8th Edt, McGraw- Hill, 2015

Further bibliography and journal articles will cover specific areas of knowledge.