Universidad de Alcalá

TEACHING GUIDE

Software engineering

Degree in
Information System Engineering (GISI)
Computer Engineering (GIC)
Computer Science Engineering (GII)
Computer Science Engineering and Business Management and Administration (GII-ADE)

Universidad de Alcalá

Academic Year 2024/2025

2nd Year - 2nd Semester (GISI+GIC+GII)

1st Year - 2nd Semester (GII-ADE)

Approved by the EPS School Board on June 14th, 2024

TEACHING GUIDE

Course Name:	**Software engineering**
Code:	**780015 (GISI+GIC+GII+GII-ADE)**
Degree in:	**Information System Engineering (GISI)** **Computer Engineering (GIC)** **Computer Science Engineering (GII)** **Computer Science Engineering and Business Management and Administration (GII-ADE)**
Department and area:	**Ciencias de la Computación** **Computer Science**
Type:	**Compulsory (GISI+GIC+GII+GII-ADE)**
ECTS Credits:	**6.0**
Year and semester:	**2nd Year - 2nd Semester (GISI+GIC+GII)** **1st Year - 2nd Semester (GII-ADE)**
Teachers:	Check the course Website at the University.
Tutoring schedule:	Check with the academics involved
Language:	English

1. COURSE SUMMARY

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.

This subject introduces the principles of 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 software will be discussed.

Prerequisites and recommendations
It is desirable for the student to have been previously enrolled in the following subjects: Fundamentals of Programming and Data Structures.

2. SKILLS

Basic, Generic and Cross Curricular Skills.

This course contributes to acquire the following basic, generic and corss curricular skills:

en_CG5 - Ability to conceive, develop and maintain computer systems, services and applications using software engineering methods as an instrument for quality assurance, in accordance with the knowledge acquired as established in section 5, annex 2, of the resolution BOE-A-2009-12977.

en_CG8 - Knowledge of the basic subjects and technologies, which enable them to learn and develop new methods and technologies, as well as those that provide them with great versatility to adapt to new situations.

en_CG9 - Ability to solve problems with initiative, decision making, autonomy and creativity. Ability to know how to communicate and transmit the knowledge, skills and abilities of the profession of Computer Engineering Engineer.

en_CG12 - Knowledge and application of basic elements of economics and human resources management, organization and planning of projects, as well as legislation, regulation and standardization in the field of computer projects, in accordance with the knowledge acquired as established in section 5 , annex 2, of resolution BOE-A-2009-12977.

en_CB1 - That students have demonstrated to possess and understand knowledge in an area of study that is based on general secondary education, and is usually found at a level that, although supported by advanced textbooks, also includes some aspects that involve knowledge from the forefront of their field of study.

en_CB2 - That the students know how to apply their knowledge to their work or vocation in a professional manner and possess the competencies that are usually demonstrated through the elaboration and defense of arguments and the resolution of problems within their area of study.

en_CB3 - That students have the ability to gather and interpret relevant data (usually within their area of study) to make judgments that include a reflection on relevant social, scientific or ethical issues.

en_CB4 - That students can transmit information, ideas, problems and solutions to both a specialized and non-specialized public.

en_CB5 - That the students have developed those learning skills necessary to undertake further studies with a high degree of autonomy.

en_TRU1 - Capacity of analysis and synthesis.

en_TRU2 - Oral and written competencies.

en_TRU3 - Ability to manage information.

en_TRU4 - Autonomous learning skills.

en_TRU5 - Team work.

Specificl Skills

This course contributes to acquire the following specific skills:

en_CI3 - Ability to understand the importance of negotiation, effective work habits, leadership and communication skills in all software development environments.

en_CI4 - Ability to prepare the technical specifications of a computer installation that meets current standards and regulations.

en_CI16 - Knowledge and application of the principles, methodologies and life cycles of software engineering.

en_CI17 - Ability to design and evaluate computer person interfaces that guarantee accessibility and usability to computer 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

Contents Blocks	Total number of hours
Introduction to Software Engineering	8 hours
Requirements and Analysis	8 hours
Design and implementation	8 hours
Software Testing	8 hours
Software Maintenance	8 hours
Software Processes and management	8 hours
Software Quality and Metrics	8 hours

Laboratory assignments will include the following blocks.

Contents Blocks	Lab Assignment
Requirements, 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. TEACHING - LEARNING METHODOLOGIES. FORMATIVE ACTIVITIES.

4.1. Credits Distribution

Number of on-site hours:	60 hours (56 hours on-site +4 exams hours)
Number of hours of student work:	90
Total hours	150

4.2. Methodological strategies, teaching materials and resources

The teaching strategy of the course is divided into 3 sections: classroom learning, learning in small groups and finally the working sessions in the laboratory.

Sessions of large group in the classroom:

Working sessions in the classroom, in large groups, will consist of lectures where the main concepts of the theory of circuits will be presented. The aim is to introduce students to the theoretical foundations of circuit analysis in a guided and reflective way. The understanding of these concepts will culminate with the use of them in both the laboratory and the problem-solving sessions in small groups.

Teaching materials will be essential to create reflective learning environments, where students and teachers can undertake a critical analysis that allows the student to autonomously relate concepts.

The order of presentation of the contents will evolve from the simple to the complex, in order to avoid a high degree of abstraction that might cause a student lack of interest in the course. In any case, it is very convenient, during the working sessions in the classroom, to establish linkages with other subjects in the curriculum, and to provide possible experience on the contents, which will help to attract students' attention and will encourage their interest in the subject.

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

Face-to-face activities:

1. In the classroom: Presentation and discussion of the basic knowledge of the subject. Theoretical approach and resolution of exercises and related assumptions. Activities (readings, discussions, cases, etc.) oriented to the teaching of the specific competences of the subject.
2. In the laboratory: Presentation, development and solution of practical exercises using tools, techniques and methods object of study of the subject, contributing to the development of the capacity of analysis, critical reasoning and understanding of the practices used.

Non-face-to-face activities:

1. Analysis and assimilation of the contents of the subject, problem-solving, bibliographical consultation, preparation of individual and/or group work, performance of face-to-face exams and self-evaluations. Especially oriented to the development of methods for the organization and planning of individual and team work.
2. Tutorials: individual and group course tutoring during the teaching-learning process, either in person or remotely.

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

5. ASSESSMENT: procedures, evaluation and grading criteria

Preferably, students will be offered a continuous assessment model that has characteristics of formative assessment in a way that serves as feedback in the teaching-learning process.

5.1. PROCEDURES

The evaluation must be inspired by the criteria of continuous evaluation (Learning Assesment Guidelines, LAG, art 3). However, in compliance with the regulations of the University of Alcalá, an alternative process of final evaluation is made available to the student in accordance with the Learning Assesment Guidelines as indicated in Article 10, students will have a period of fifteen days from the start of the course to request in writing to the Director of the Polytechnic School their intention to take the non-continuous evaluation model adducing the reasons that they deem convenient. The evaluation of the learning process of all students who do not apply for it or are denied it will be done, by default, according to the continuous assessment model. The student has two calls to pass the subject, one ordinary and one extraordinary.

The assessment must be inspired by the criteria of continuous assessment (Learning Assessment Regulations, NEA, art 3).

However, in accordance with the regulations of the University of Alcalá, an alternative final assessment process is an alternative final assessment process is available to the student in accordance with the Learning Assessment Regulations (last modified by the Governing Council on 31 October 2019) as indicated in its Article 10, students will have a period of fifteen days from the start of the course to apply in writing to the Director of Polytechnic School their intention to opt for the non-continuous assessment, stating the reasons they deem appropriate. The assessment of the learning process of all students who do not submit a request or whose request is denied will be carried out by default in accordance with the continuous assessment model. The student has two exams to pass the course, one in the ordinary call and one extraordinary one. Students will be subject to the assessment procedures in accordance with the provisions of Title 2 (art. 9 and 10) of the Learning Assessment Regulations of the UAH.

Continuous assessment will consist of a series of theory and laboratory tests that will be carried out throughout the course.

5.2. EVALUATION

EVALUATION CRITERIA

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

Regarding the works 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.

Ordinary Call

Continuous Assessment:

Continuous Assessment Tests (CATs or PEI) are carried out during theory classes. Laboratory tests (Continuous Assessment Laboratory –CALs or PL–) will be assignments that might require to 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 blocks 1 and 2
- PEI2: Theoretical-practical knowledge test on blocks 3, 4 and 5
- PEI3: Theoretical-practical knowledge test on blocks 6 and 7

Laboratory assessments:

- PL1: Group or individual work on laboratory practices.
- PL2: Group or individual work on laboratory practices.

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	30
CG5,CG8, CG9,CG12	CI16, CI17	RA1, RA2, RA6, RA7	CE2, CE7, CE8	PL2	20

Assessment through final exam:

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	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	30
CG5,CG8, CG9,CG12	CI16, CI17	RA1, RA2, RA6, RA7	CE2, CE7, CE8	PL2	20

Extraordinary call (Resit)

In the case of the extraordinary call exam, the same percentages that have been established in the case of the evaluation by means of a final exam will be maintained, giving the option of making the PL or maintaining the mark obtained in the EL (continuous evaluation) or in the PEF (final evaluation), according to the student's decision. In any case, the PL will be made by those students who have not done it in the final exam option in the ordinary call.

6. BIBLIOGRAPHY

6.1. Basic Bibliography

- 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

6.2. Additional Bibliography

Further relevant bibliography and articles from digital libraries (ACM, IEEE, Springer, Elsevier, etc.) will be provided.

Disclosure Note

During the evaluation tests, the guidelines set out in the Regulations establishing the Rules of Coexistence of the University of Alcalá must be followed, as well as the possible implications of the irregularities committed during said tests, including the consequences for committing academic fraud according to the Regulation of Disciplinary Regime of the Students of the University of Alcalá.