ANX-PR/CL/001-01

LEARNING GUIDE

SUBJECT

615000253 - Software Architecture & Design

DEGREE PROGRAMME

61IW – Degree in Software Engineering

ACADEMIC YEAR & SEMESTER

2022/23 - Semester 2

Index

Learning guide

1. Description

1.1. Subject details

Name of the subject	615000253 - Software Architecture & Design
No of credits	6 ECTS
Type	Mandatory
Academic year ot the programme	Third year
Semester of tuition	Semester 6
Tuition period	February - June
Tuition languages	English
Degree programme	61IW – Software Engineering Bachelor
Centre	61 - Escuela Tecnica Superior De Ingenieria De Sistemas Informaticos
Academic year	2022-23

2. Faculty

2.1. Faculty members with subject teaching role

Name and surname	Office/Room	Email	Tutoring hours *
Silvia Alba Uribe Mayoral (Coordinador/a)	4218	silviaalba.uribe@upm.es	To be confirmed.
Manuel Ottaviano	1126	manuel.ottaviano@upm.es	To be confirmed

* The tutoring schedule is indicative and subject to possible changes. Please check tutoring times with the faculty member in charge.

3. Prior knowledge recommended to take the subject

3.1. Recommended (passed) subjects

- Software Process Engineering & Construction
- Software Engineering Fundamentals
- Requirements Engineering and Modelling

3.2. Other recommended learning outcomes

- It is recommended some knowledge in Software Engineering Fundamentals, Software Engineering Processes and Requirements Engineering.

4. Skills and learning outcomes *

4.1. Competences

- CC8 - Ability to analyze, design, build, and maintain applications in a robust, secure, and efficient manner, choosing the most suitable programming languages and paradigms.
- CE2 - Ability to assess customer needs and specify software requirements to meet these needs, reconciling conflicting objectives by seeking acceptable compromises within the limitations derived from cost, time, existing developed systems, and organizational factors.
- CE4 - Ability to identify and analyze problems and design, develop, implement, verify, and document software solutions based on adequate knowledge of current theories, models, and techniques.
- CT2 - Problem-solving: Identify, analyze, and define the significant elements that constitute a problem in order to solve it effectively and with sound judgment.
- CT3 - Oral communication: Clearly and timely express one's own ideas, knowledge, and reflections through words, adapting to the characteristics of the situation and the audience to achieve understanding.

4.2. .Learning outcomes

- RA105 - Identifies and analyzes problems to provide software solutions based on adequate knowledge of current theories, models, and techniques.
- RA106 - Recognizes and applies current theories, models, and techniques for problem identification, analysis, software design, development, implementation, verification, and documentation.
- RA39 - Models and designs solutions considering efficiency and modularity commitments.
- RA83 - Performs a complex task autonomously, selecting the most suitable strategies to approach the study based on the analysis of conditions and proposed goals. Analyzes and interprets information, manages information and communication technologies (ICTs), demonstrates communication and interaction skills for collaborative learning. Values the effectiveness of task planning and makes timely decisions to achieve their purpose.
- RA47 - Capable of working as a team member to contribute to the development of projects with pragmatism and a sense of responsibility, assuming commitments and considering available resources.

Goals:

Be able to know and apply current theories, models and techniques for problem identification, analysis, software design, development, implementation, verification and documentation.

• Model and design solutions according to the efficiency, modularity, and modularity tradeoffs.

• Identify and analyse problems to solve software solutions based on an adequate knowledge of theories, models and techniques.

• Be able to perform a complex task in an autonomous way, selecting the most convenient strategies to approach the study, based on the analysis and approach the study, based on the analysis of the conditions and the proposed goal.

• Evaluate the effectiveness of the planning of tasks and make the appropriate decisions to achieve their purpose.

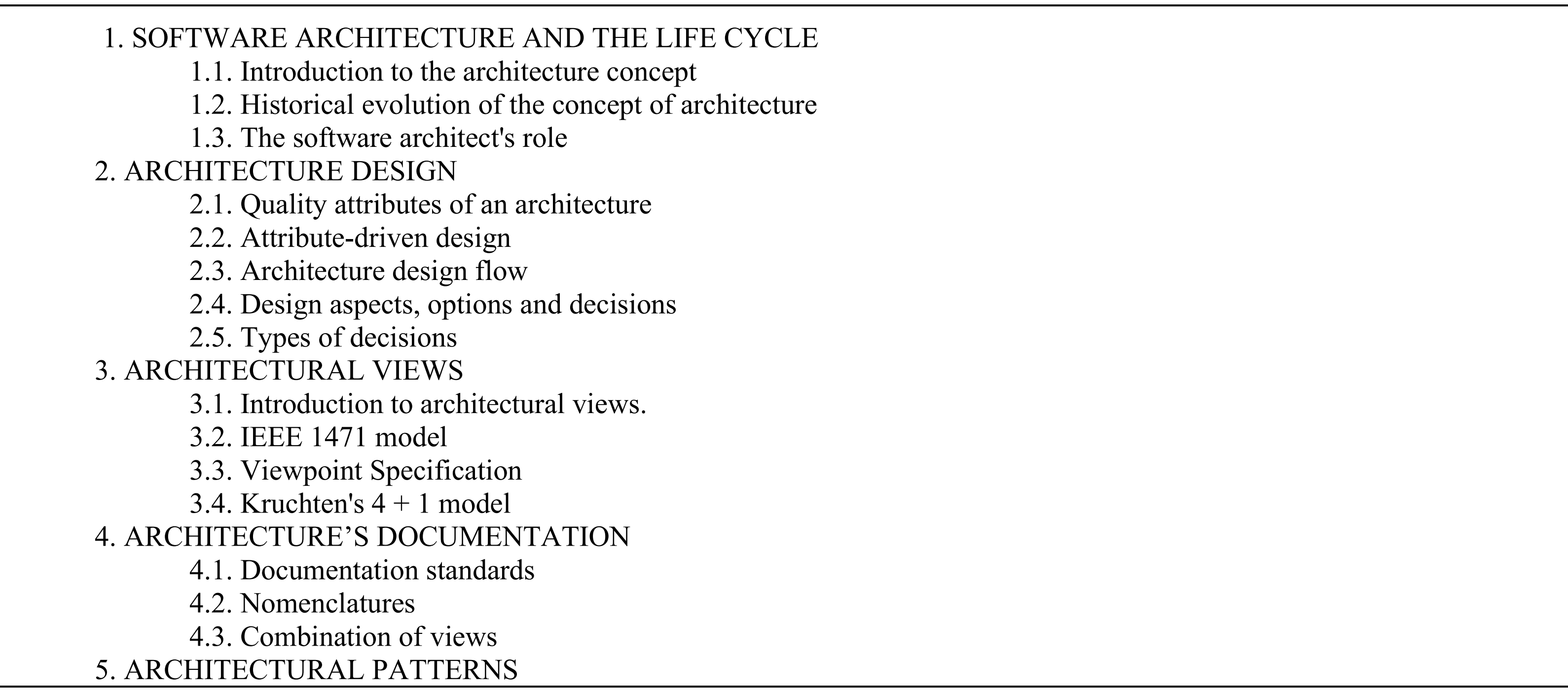
• Be able to work as a member of a team with the purpose of contributing to the development of projects with pragmatism and sense of responsibility, assuming commitments and taking into account the available resources. available.

5. Brief description of the subject and syllabus

5.1. Brief description of the subject

The software architecture translates the requirements into a more detailed model representing of the software solution, which will typically include the architecture design specifications and detailed design specifications. Alternatively, in software engineering, it is the process of defining the software architecture (the structure), components, modules, interfaces, test methods, and data for the software system to satisfy given requirements. For the design of a software architecture it will be necessary to use appropriate methods and select the architectural patterns that are considered more suitable for the resulting system such that it meets the required quality attributes.

Syllabus



1. SOFTWARE ARCHITECTURE AND THE LIFE CYCLE
    1.1. Introduction to the architecture concept
    1.2. Historical evolution of the concept of architecture
    1.3. The software architect's role
2. ARCHITECTURE DESIGN
    2.1. Quality attributes of an architecture
    2.2. Attribute-driven design
    2.3. Architecture design flow
    2.4. Design aspects, options and decisions
    2.5. Types of decisions
3. ARCHITECTURAL VIEWS
    3.1. Introduction to architectural views.
    3.2. IEEE 1471 model
    3.3. Viewpoint Specification
    3.4. Kruchten's 4 + 1 model
4. ARCHITECTURE’S DOCUMENTATION
    4.1. Documentation standards
    4.2. Nomenclatures
    4.3. Combination of views
5. ARCHITECTURAL PATTERNS

5.1. Introduction to Architectural Patterns
5.2. Description of architectural patterns
6. ARCHITECTURE EVALUATION AND REFACTORING
6.1. Introduction to Architecture Evaluation
6.2. Architectural Analysis Techniques
6.3. ATAM model of architectures evaluation
6.4. Refactoring
7. REFERENCE ARCHITECTURES
7.1. Concept of reference architecture
7.2. Examples of reference architectures
8. PRODUCT LINES
8.1. Examples of reference architectures
8.2. Feature models

6. Schedule

6.1. Subject schedule*

Week	Classroom activities	Laboratory activities	Distant / On-line	Assessment activities
1	**Theoretical lesson** Lecture			
2	**Theoretical lesson** Lecture	**Practical lesson** Duration: 02:00		
3	**Theoretical lesson**	**Practical lesson** Duration: 02:00		
4	**Theoretical lesson**	**Practical lesson** Duration: 02:00		
5	**Theoretical lesson**	**Practical lesson** Duration: 02:00		
6	**Theoretical lesson**	**Practical lesson** Duration: 02:00		
7	**Theoretical lesson**	**Practical lesson** Duration: 02:00		
8	**Project development part I** Laboratory lesson Duration: 02:00	**Project development part I** Duration: 02:00		Written test of theoretical and practical knowledge. (RA39, RA105, RA106) EX: Evaluation Technique: Written Exam Continuous Assessment Face to face Duration: 02:00 hours
9	**Theoretical lesson**			Assignment 1 Submission. (RA39, RA105, RA106, RA83, RA47, RA65) TG: Evaluation Technique: Group Work Continuous Assessment with only a final test Non-face-to-face Duration: 00:00 (Not applicable)
10	**Theoretical lesson**	**Project development part II** Duration: 02:00		
11	**Theoretical lesson** Lecture **Theoretical lesson** Lecture			

12	**Theoretical lesson**	**Project development part II** Duration: 02:00		
13	**Theoretical lesson**	**Project development part II** Duration: 02:00		
14	**Theoretical lesson**	**Project development part II** Duration: 02:00		
15	**Theoretical lesson**	**Project development part II** Duration: 02:00		
16				Written test of theoretical and practical knowledge. (RA39, RA105, RA106) EX: Evaluation technique: Written Exam Continuous Assessment In-person Duration: 02:00 Assignment 2 Submission. (RA39, RA105, RA106, RA83, RA47, RA65) TG: Evaluation technique: Group Work Continuous Assessment with only a final test Non-face-to-face Duration: 00:00 (Not applicable) Evaluation of in-class teamwork assignments. (RA106, RA65, RA47) TG: Evaluation technique: Group Work Continuous Assessment In-person Duration: 00:00 (Not applicable) Written test of theoretical and practical knowledge. (RA39, RA105, RA106) EX: Evaluation technique: Written Exam Assessment based only on the final test In-person Duration: 02:00

7. Activities and assessment criteria

7.1. Assessment activities

7.1.1. Assessment for progressive attendance

Week	Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
8	Written Exam	Written exam	Face to face	02:00	25%	4/10	CE4 CT2 CE2
9	Submission of Project Part I	Workteam	Non-presential	00:00	30%	3/10	CE4 CT2 CC8 CT3 CE2
17	Written Exam	Written exam	Face to face	02:00	15%	4/10	CE4 CT2 CE2
17	Submission of Project Part II	Workteam	Non-presential	00:00	20%	3/10	CE4 CT2 CC8 CT3 CE2

17	Evaluation of in-class assignments as teamwork: Evaluation technique: Group Work Continuous Assessment In-person Duration: Variable (dependent on the specific in-class assignment)	Teamwork	Non-presential	00:00	10%	0 / 10	CT3

7.1.2. Global examination

Week	Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
9	Submission of Project Part I	Workteam	Non-presential	00:00	30%	3/10	CE4 CT2 CC8 CT3 CE2
17	Submission of Project Part II	Workteam	Non-presential	00:00	20%	3/10	CE4 CT2 CC8 CT3 CE2

7.1.3. Referred (re-sit) examination

Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
"Final exam"	Written exam	Face to face	02:30	50%	5/ 10	CC8 CE4 CE2 CT2
Practical Evaluation 1.	Workteam	Face to face	00:00	30%	3/ 10	CE2 CT2 CC8 CT3 CE4
Practical Evaluation 2.	Workteam	Face to face	00:00	20%	3/ 10	CE2 CT2 CC8 CE4

7.2. Assessment criteria

1. ASSESSMENT CRITERIA

The evaluation (progressive) will consist of the tests described above. The theoretical-practical exams and the Practical Project have a minimum grade requirement. In order to pass the course, it is mandatory to have passed the minimum grades in all evaluation activities that have them, as well as to have a final weighted grade equal to or higher than 5, according to the weights assigned to each part.

For the progressive evaluation, in the second written test, students who did not reach the minimum grade in the first test will have the opportunity to recover it through an additional test related to that first part.

The evaluation with only the final test consists of a different number of tests, and each test has different weights, as indicated in the evaluation activities.

The completion of the Practical assignments is mandatory and must be submitted on time for all evaluations, as it is not possible to assess the cross-cutting competencies without them. The assignments should be submitted at the specified times.

It is not possible to pass the course in parts. To pass the course, it is necessary to obtain a grade equal to or higher than 5 in any of the exam sessions, taking into account the grades from the different tests, provided that the minimum grade requirement has been met in all applicable assessments.

7.3. Teaching resources for the subject

Name	Type	Notes
Moodle UPM	Web resource	The whole pack of documentation and examples used in class by the teacher. It is documentation elaborated by the teacher
Software Architecture: Foundations, Theory, and Practice. R. N. Taylor ,N. Medvidovic , E. M. Dashofy.Wiley, 2009. ISBN-10: 0470167742. ISBN-13: 978-0470167748	Bibliography	Main book.
Software Systems Architecture:Working with Stakeholders Using Viewpoints and Perspectives, Second Edition. Nick Rozanski; EoinWoods. Addison-Wesley Professional. October 25, 2011. ISBN-10: 0-321-71833-X. ISBN-13: 978-0-321-71833-4	Bibliography	Main book
Applied SOA: Service-Oriented Architecture And Design Strategies. Michael Rosen; Boris Lublinsky; Kevin T. Smith; Marc J. Balcer. John Wiley & Sons. 2008. ISBN: 978-0-470-22365-9	Bibliography	Complementary book
Documenting Software Architectures: Views and Beyond, Second Edition. By: Paul Clements; Felix Bachmann; Len Bass; David Garlan; James Ivers; Reed Little; Paulo Merson;Robert Nord; Judith Stafford. Addison-Wesley Professional. 05-OCT-2010	Bibliography	Complementary book.
Essential Software Architecture.Ian Gorton. Springer-Verlag Berlin and Heidelberg GmbH & Co. K; Edición: 2nd ed. 2011. ISBN-10: 3642191754.ISBN-13: 978-3642191756	Bibliography	Complementary book.
Software Architecture in Practice, Third Edition. Len Bass; Paul Clements; Rick Kazman. Addison-Wesley Professional. 2015. Colección Safari	Bibliography	Complementary book.
Beyond Software Architecture: Creating and Sustaining Winning Solutions. Luke Hohmann. Addison-Wesley Professional. 2003. Colección Safari	Bibliography	Complementary book.
Software Engineering: Principles and Practice. Hans van Vliet. John Wiley & Sons. 2008. Safari	Bibliography	Complementary book.
Software Architecture: A Case Based Approach. Vasudeva Varma. Pearson Education India. 2009. ISBN-10: 81-3170-749-0. ISBN-13: 978-8-131-70749-4	Bibliography	Complementary book.
Scaling Software Agility: Best Practices for Large Enterprises. Dean Leffingwell. Addison-Wesley Professional . 2007. Safari	Bibliography	Complementary book.

8. Other information

The subject of Software Architecture and Design contributes to the Sustainable Development Goals (SDGs) by addressing the following practices in the course:

1. Objective 3: Good Health and Well-being - Be He@lthy, Be Mobile Initiative, direct patient interaction, health informatics, and telemedicine.
2. Objective 11: Sustainable Cities and Communities - Smart and sustainable cities, intelligent transportation systems, 5G technology, and the Internet of Things.

By incorporating these topics into the course's practices, students are exposed to the importance of leveraging technology to improve healthcare delivery and enhance the sustainability and efficiency of urban environments. This aligns with the broader agenda of the SDGs, promoting a more sustainable and inclusive future.