

Course guide 270080 - AS - Software Architecture

Last modified: 30/01/2024

Unit in charge:
Barcelona School of Informatics

Teaching unit:
747 - ESSI - Department of Service and Information System Engineering.

Degree:
BACHELOR'S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2010). (Optional subject).

Academic year: 2023
ECTS Credits: 6.0

LECTURER

Coordinating lecturer:
CRISTINA GÓMEZ SEOANE

Others: Primer quadrimestre: CLAUDIA PATRICIA AYALA MARTINEZ - 12, 13, 21, 22 CRISTINA GÓMEZ SEOANE - 11, 12, 13, 14, 22

Segon quadrimestre: PAU DURAN MANZANO - 11 CRISTINA GÓMEZ SEOANE - 11, 12, 13

PRIOR SKILLS

Knowing what is a UML/OCL formal specification of functional and nonfunctional requirements of a system. Knowing the context in which the design of a system may be performed and, in particular, what are the previous and following activities.

Knowing the object-oriented programming and (at least) an object-oriented programming language.

REQUIREMENTS

- Prerequisite IES

DEGREE COMPETENCES TO WHICH THE SUBJECT CONTRIBUTES

Specific:

CES1.1. To develop, maintain and evaluate complex and/or critical software systems and services.

CES1.4. To develop, mantain and evaluate distributed services and applications with network support.

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.

Generical:

G9. PROPER THINKING HABITS: capacity of critical, logical and mathematical reasoning. Capacity to solve problems in her study area. Abstraction capacity: capacity to create and use models that reflect real situations. Capacity to design and perform simple experiments and analyse and interpret its results. Analysis, synthesis and evaluation capacity.



TEACHING METHODOLOGY

The teaching of the course is structured in theory, exercises and laboratory sessions.

In the theory sessions teachers present some essential content of the course. Typically teachers use slides that students have available before the session. Other contents of the subject will be studied autonomously.

In the exercises sessions the contents of the subject are practiced through the exercises. In the exercises sessions proposed exercises will be solved and several solutions will be discussed. In some sessions, students deliver the exercises solved. Also, in some sessions, at the beginning of the session, the students answer a questionnaire about the contents presented and practiced and/or about the exercises solved in previous sessions.

In the laboratory classes the students work in groups of two (pair programming). During the session, the students solve the laboratory exercises proposed by the teacher. At the beginning of some sessions, the students answer a questionnaire about the contents presented and practiced and/or about the exercises proposed in the previous session.

The students must attend to the group sessions where they are enrolled.

LEARNING OBJECTIVES OF THE SUBJECT

1.To explain what is the logical and physical architecture of a software system and relations between them.

2.To explain the characteristics of a layered architecture and the design principles that govern the construction of software systems with this type of architecture.

3.To explain the characteristics of an object-oriented architecture and the design principles that govern the construction of software systems with this type of architecture.

4.To design software systems using a layered and object-oriented software using software services from the specification of requirements (functional and nonfunctional) using traditional and agile methodologies.5.To represent the design of software systems using UML.

6.To correctly apply design principles when designing software systems.

7.To design the domain (business) layer of a software system using traditional and agile methodologies.8.To design the presentation layer of a software system by identifying the design patterns to apply and applying them correctly.

- 9.To explain the characteristics of the different persistence strategies.
- 10.To design the data management layer of a software system using access or data mapper patterns.

11.To identify and correctly use existing software services when designing software systems.

12.To evaluate and critique the design of software systems and to propose alternative designs to improve their quality.

13.To explain what is softtware architecture and design and what are the different views of software architecture.

STUDY LOAD

Туре	Hours	Percentage
Guided activities	6,0	4.00
Hours large group	30,0	20.00
Hours small group	30,0	20.00
Self study	84,0	56.00

Total learning time: 150 h



CONTENTS

Introduction to the software architecture and design

Description:

Software development methodologies. Software architecture and design. Logical Architecture and Physical Architecture. Role of design patterns in software design. Software architecture and design in traditional and agile methodologies.

Layered and object oriented architectures

Description:

Layered architectural pattern. Application of layered architectural pattern to software systems and services. Layered architecture design principles. Object oriented architectural pattern. Application of object oriented architectural pattern to software systems. Object oriented architecture design principles.

Software design using traditional methodologies

Description:

Software design using traditional methodologies. Domain layer design. Domain Model and Transaction Script patterns. Domain layer design patterns. Use of software services. Presentation layer design. External design of a software system. Presentation layer internal design. Data layer design. Relational databases technology. Persistence strategies. Automatic generation of persistence. Direct design of persistence. Other architectures. Microservice architecture.

Software design using agile methodologies

Description:

Software design using agile methodologies. Extreme Programming. TDD. Tests in TDD. Coding in TDD. Pair Programming. Simple Design and Incremental design and architecture. Code Smells and Refactorings.

ACTIVITIES

C2

Description:

Test 2 (theory and exercises) where part of the objectives of chapter 3 are evaluated

Specific objectives:

4, 6, 8, 9, 10, 11

Related competencies :

G9. PROPER THINKING HABITS: capacity of critical, logical and mathematical reasoning. Capacity to solve problems in her study area. Abstraction capacity: capacity to create and use models that reflect real situations. Capacity to design and perform simple experiments and analyse and interpret its results. Analysis, synthesis and evaluation capacity.

Full-or-part-time: 12h Guided activities: 2h Self study: 10h



C1

Description:

Test 1 (theory and exercises) where all objectives of chapters from 1 to part of 3 are evaluated

Specific objectives: 1, 2, 3, 4, 5, 6, 7, 13

Related competencies :

G9. PROPER THINKING HABITS: capacity of critical, logical and mathematical reasoning. Capacity to solve problems in her study area. Abstraction capacity: capacity to create and use models that reflect real situations. Capacity to design and perform simple experiments and analyse and interpret its results. Analysis, synthesis and evaluation capacity.

Full-or-part-time: 12h Guided activities: 2h Self study: 10h

Introduction to software architecture

Description:

Listen carefully, take notes and ask questions if necessary. In addition, the student studies the contents that the teacher indicates

Specific objectives: 1, 13

Full-or-part-time: 3h Theory classes: 2h Self study: 1h

Studying the layered architecture

Description:

Listen carefully, take notes and ask questions if necessary. In addition, the student studies the contents that the teacher indicates

Specific objectives: 2

Full-or-part-time: 3h Laboratory classes: 1h Self study: 2h



Styding the object-oriented architecture

Description:

Listen carefully, take notes and ask questions if necessary. In addition, the student studies the contents and does the exercises that the teacher indicates

Specific objectives:

3, 5, 12

Related competencies :

G9. PROPER THINKING HABITS: capacity of critical, logical and mathematical reasoning. Capacity to solve problems in her study area. Abstraction capacity: capacity to create and use models that reflect real situations. Capacity to design and perform simple experiments and analyse and interpret its results. Analysis, synthesis and evaluation capacity.

Full-or-part-time: 14h

Laboratory classes: 6h Self study: 8h

Studying software design using traditional methodologies

Description:

Listen carefully, take notes and ask questions if necessary. In addition, the student studies the contents and does the exercises that the teacher indicates

Specific objectives:

4, 5, 6, 7, 8, 9, 10, 11

Related competencies :

G9. PROPER THINKING HABITS: capacity of critical, logical and mathematical reasoning. Capacity to solve problems in her study area. Abstraction capacity: capacity to create and use models that reflect real situations. Capacity to design and perform simple experiments and analyse and interpret its results. Analysis, synthesis and evaluation capacity.

Full-or-part-time: 56h

Theory classes: 14h Laboratory classes: 14h Self study: 28h

Studying software design usign agile methodologies

Description:

Listen carefully, take notes and ask questions if necessary. In addition, the student studies the contents and does the exercises that the teacher indicates

Specific objectives:

4, 6, 7, 12

Related competencies :

G9. PROPER THINKING HABITS: capacity of critical, logical and mathematical reasoning. Capacity to solve problems in her study area. Abstraction capacity: capacity to create and use models that reflect real situations. Capacity to design and perform simple experiments and analyse and interpret its results. Analysis, synthesis and evaluation capacity.

Full-or-part-time: 32h

Theory classes: 10h Laboratory classes: 7h Self study: 15h



С3

Description:

Test 3 (laboratory) where all objectives of chapter 4 are evaluated

Specific objectives: 6, 7, 12

Related competencies :

G9. PROPER THINKING HABITS: capacity of critical, logical and mathematical reasoning. Capacity to solve problems in her study area. Abstraction capacity: capacity to create and use models that reflect real situations. Capacity to design and perform simple experiments and analyse and interpret its results. Analysis, synthesis and evaluation capacity.

Full-or-part-time: 12h Guided activities: 2h Self study: 10h

Review of C1 (Control 1) with formative feedback (solution delivery with group feedback)

Description:

Review of C1 (Control 1) with formative feedback (solution delivery) in groups of 10 students (via meet or with assigned classroom, depending on the availability of classrooms).

Specific objectives:

1, 2, 3, 4, 5, 6, 7

Related competencies :

G9. PROPER THINKING HABITS: capacity of critical, logical and mathematical reasoning. Capacity to solve problems in her study area. Abstraction capacity: capacity to create and use models that reflect real situations. Capacity to design and perform simple experiments and analyse and interpret its results. Analysis, synthesis and evaluation capacity.

Full-or-part-time: 2h

Guided activities: 2h

Review of C2 (Control 2) with formative feedback (solution delivery with group feedback)

Description:

Review of C1 (Control 1) with formative feedback (solution delivery) in groups of 10 students (via meet or with assigned classroom, depending on the availability of classrooms).

Specific objectives:

4, 6, 8, 9, 10, 11

Related competencies :

G9. PROPER THINKING HABITS: capacity of critical, logical and mathematical reasoning. Capacity to solve problems in her study area. Abstraction capacity: capacity to create and use models that reflect real situations. Capacity to design and perform simple experiments and analyse and interpret its results. Analysis, synthesis and evaluation capacity.

Full-or-part-time: 2h

Guided activities: 2h



Review of C3 (Control 3) with formative feedback (solution delivery with group feedback)

Description:

Review of C3 (Control 3) with formative feedback (solution delivery) in groups of 10 students (via meet or with assigned classroom, depending on the availability of classrooms).

Specific objectives:

6, 7, 12

Related competencies :

G9. PROPER THINKING HABITS: capacity of critical, logical and mathematical reasoning. Capacity to solve problems in her study area. Abstraction capacity: capacity to create and use models that reflect real situations. Capacity to design and perform simple experiments and analyse and interpret its results. Analysis, synthesis and evaluation capacity.

Full-or-part-time: 2h

Guided activities: 2h

GRADING SYSTEM

The final grade (NF) of the technical skills of the course is the following:

NF= 0,2*C1 + 0,25*C2 + 0,4*C3 + 0,1*NT + NPP

C1 is the first test mark

C2 is the second test mark

C3 is the third test mark

NT is the mark of the tasks done (delivery of exercises before the class and questionnaires made at the beginning of the classes of exercises and laboratory).

NPP is the mark of the delivered exercises and class participation.

Assesment of NT:

- During the course the students may take 5 tasks.

- If the student delivers 4 or less tasks, NT will be equal to the sum of the marks of all tasks delivered divided by 4.

- If the student delivers the 5 tasks, NT will be equal to the sum of the 4 better marks divided by 4.

Assessment of NPP: This grade will have a maximum value of 0.5. To obtain this grade, the following aspects will be considered:

- The delivery of 75% or more of the problems proposed to be delivered.

- Active participation in class.

Students will be only marked if they attend to the group sessions where they are enrolled.

Final mark of transversal skill is evaluated by one question in the exams where the students must demostrate their critical thinking evaluating the proposed designs. Final mark is the average of the marks obtained in these questions and have an assessment of:

A is the mark is between 8.5 and 10 $\,$

B is the mark is between 7 and 8.4 $\,$

C is the mark is between 5 and 6.9 $\,$

D is the mark is between 0 and 4.9



BIBLIOGRAPHY

Basic:

- Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. Design patterns: elements of reusable object-oriented software. Addison-Wesley, 1995. ISBN 0201633612.

- Larman, C. Applying UML and patterns: an introduction to object-oriented analysis and design and iterative development. 3rd ed. Prentice Hall PTR, 2005. ISBN 0131489062.

- Fowler, M. Patterns of enterprise application architecture. Addison-Wesley, 2003. ISBN 0321127420.

- Rumbaugh, J.; Jacobson, I.; Booch, G. The unified modeling language reference manual. 2nd ed. Addison-Wesley, 2005. ISBN 0321245628.

- Shore, J.; Warden, S. The art of agile development. O'Reilly Media, 2008. ISBN 9780596527679.

- Christensen, H.B. Flexible, reliable software: using patterns and agile development. CRC Press, 2010. ISBN 9781420093629.
- Meszaros, G. XUnit test patterns: refactoring test code. Addison-Wesley, 2007. ISBN 9780131495050.

Complementary:

- Pressman, R.S.; Maxim, B.R. Software engineering: a practitioner's approach. 9th ed. New York: McGraw Hill Higher Education, 2020. ISBN 9781260548006.

- Sommerville, I. Software engineering. 10th ed., global ed. Pearson Education Limited, 2016. ISBN 9781292096148.

- Martin, R.C. Agile software development: principles, patterns, and practices. First edition. Pearson Education Limited, 2014. ISBN 9781292038360.

- Buschmann, F.; Meunier, R.; Rohnert, H.; Sommerlad, P.; Stal, M. Pattern-oriented software architecture: Vol. 1: a system of patterns. Wiley, 1996. ISBN 0471958697.

- Beck, K. Test-driven development: by example. Addison-Wesley, 2003. ISBN 0321146530.

- Professorat de l'assignatura. Material d'elaboració pròpia de l'assignatura.

RESOURCES

Hyperlink:

- <u>http://msdn.microsoft.com/en-us/library/ff650706.aspx</u>- <u>http://martinfowler.com/eaaCatalog/-</u> https://industriallogic.com/xp/refactoring/catalog.html- <u>http://www.uml.org/-</u> <u>http://refactoring.com/catalog/</u>