



Universidad de Oviedo



School of
Computer
Science



SOFTWARE
ARCHITECTURE

Software Architecture

Course 2018/2019

Jose E. Labra Gayo

Software Architecture

Degree: Computer Science - Software Engineering

Type: Mandatory, third year

Credits: 6

Period: 2nd Semester

Language: English/Spanish

Lecturers

Jose Emilio Labra Gayo labra@uniovi.es

Noelia Rico Pachón (noeliarico@uniovi.es)

Victor Manuel Álvarez García (victoralvarez@uniovi.es)

Pablo Suárez Otero-González (suarezgpablo@uniovi.es)

Javier Escalada Gómez (escaladajavier@uniovi.es)

Oscar Jesús Cosido Cobos (oscar.cosido@uniovi.es)

Time dedication

6 ECTS credits \approx 150 working hours

60 on-campus hours, 90 self-study

Organization (*by week*)

2h lectures (21h total)

1h seminars (7h total)

2h laboratory practice (28h total)

2h group tutorials

7,5h self-study (90h total)

Competences & learning outcomes

General competencies

Methodological skills

CG-1	Ability to design solutions to human complex problems
------	---

Specific competencies

Common to Computer Science

Com.1	Ability to design, develop, select and evaluate applications and systems, ensuring their reliability , safety and quality , according to ethical principles, laws and regulations.
Com.8	Ability to analyse , design , build and maintain applications in a robust, secure and efficient way, and choosing the most suitable paradigms and programming languages.
Com.11	Knowledge and application of features, functionality and structure of distributed systems , computer networks and the Internet , and to design and implement applications based on them.

Specific competencies

Software Engineering

ISW.1	Ability to develop, maintain and evaluate software systems and services that match all user requirements and behave reliably and efficiently , being affordable to develop and maintain and accomplishing quality standards, applying the theories, principles, methods and Software Engineering good practices.
ISW.3	Ability to solve integration problems in terms of strategies, standards and available technologies.
ISW.4	Ability to identify and analyse problems and to design, develop, implement, verify and document software solutions based on adequate knowledge of the theories, models and techniques.

Learning outcomes

RA.IS-1.	Making complex Software Engineering Projects that provide solutions to complex problems and to solve them using techniques and technologies related to manufacturing processes, including software frameworks, architectural patterns, design and integration patterns, pursuing quality software development
RA.IS-3.	To apply different construction techniques in designing low level software
RA.IS-4.	Develop design and object-oriented programming with a high level of competence
RA.IS-5.	To evolve and refactor existing designs to afford changing requirements
RA.IS-6.	Determining the degree of maintainability, reliability and efficiency of software designs
RA.IS-7	To design and implement software using different middleware technologies
RA.IS-9	To design and to carry out checks and efficient and effective inspections about validation, verification, quality and test plans.
RA.IS-10	Statistically analysing the density of defects and failure probability
RA.IS-11	Evaluating the quality of a software process from the point of view of product quality.

Evaluation & grading

3 possibilities

Continuous evaluation (by default)

Final assessment only

Differentiated assessment

Continuous evaluation (by default)

$$Final = Theory \times 40\% + Practice \times 60\%$$

where:

$$Theory = Exam \times 70\% + Seminars \times 30\%$$

$$Practice = Team \times 70\% + Individual \times 30\%$$

Requirements:

Minimum assistance (80%)

Minimum mark (theory & seminar): 3

Minimum mark (lab): 5

Only Final Evaluation (non continuous)

IF continuous evaluation fails*

Final mark

$$Final = Theory \times 40\% + Practice \times 60\%$$

where

Theory = Exam + Individual work (seminar)

Practice = Individual project

Public presentation of Individual Project

Usually after the final exam

* Penalizations will be applied to practice mark if students are not able to work in a team

Differentiated evaluation

Theory: The same as previous

Practice (2 possibilities)

1) Working in a team (minimal assistance 20%)

Mandatory: Participate in public presentation session

The mark will be: 70% team + 30% individual.

2) Working individually

During the first month the student will be assigned a project similar to the teams projects

Individual public presentation

General remark: Assignments that are not done or not delivered on time will count as 0

Seminars

Seminars

Works made by groups of 2/3 people

Subjects proposed by teachers

Analyze an open source system

Describe quality attributes, requirements, etc.

Public presentations during the seminars

Assessment:

Report delivered + Presentation and questions

About the practical assignment

Assessment

70% team mark+ 30% individual mark

Team mark: Presentation days

Final presentation = Mandatory (like an exam)

Teachers select the person(s) that will do the presentation

Other team members can participate

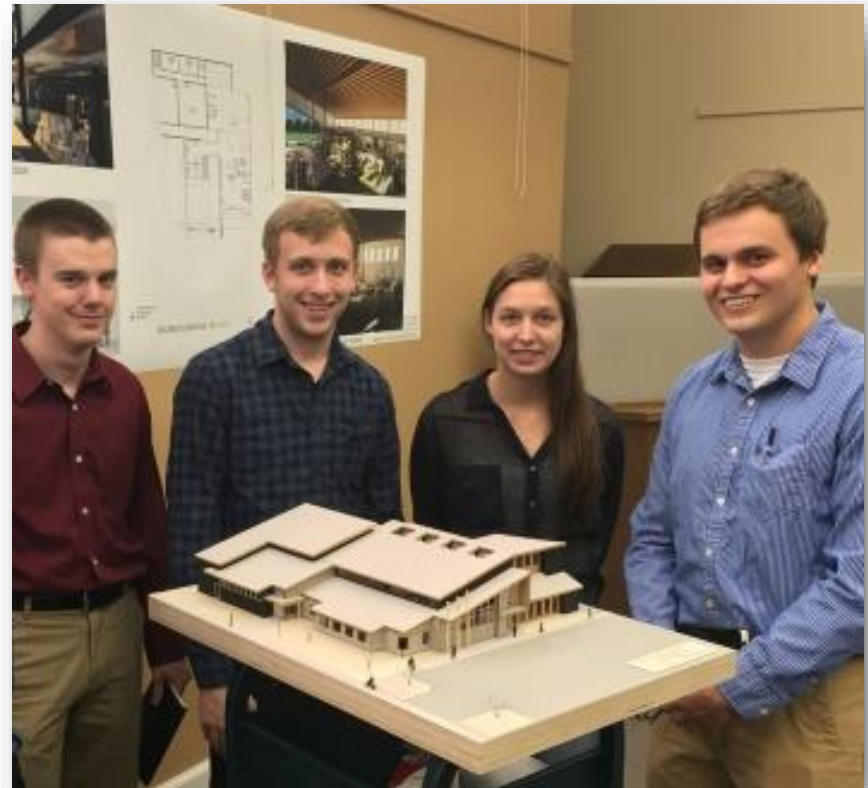
Individual mark: github contributions

Project management tool: github.com

Important: Create your github account
If possible, use a login name that resembles your first name/last name...

Team project

1. Design and document a software architecture
2. Implement prototype
3. Public presentation



About the teams

Teams created initially by teachers

Size: 5-8 people

Teams will work together the whole year

Being able to work in a team is very important!

Members that abandon will present individually

Possible penalization to the individual and the team

Lab sessions

13 lab sessions (except Mondays, only 12)

During the lab sessions (2 hours)

- 1.- A teacher will explain some concept (1 hour approx.)
- 2.- The team will work on the assignment
That hour counts as a team meeting

Team meetings

Every lab session

You can also organize your own team meetings

Mandatory: Keep record of all team meetings

One person must write the minutes

Advice: Rotate the role of scribe

Minutes must be maintained in the project wiki

General structure of minutes:

- Date/time/place of meeting

- Participants

- Decisions taken

- Actions

4 Deliverables

Checkpoint at every deliverable

1st deliverable - Week 4 (19 Feb - 25 Feb.)

Documentation 0.1

2nd deliverable - Week 7 (12 March - 18 March)

Prototype version 0.1

3rd deliverable - Week 10 (2 April - 8 April)

Prototype version 1.0 + Documentation 1.0

4th deliverable - Week 13 (29 April - 3 May)

Prototype version 1.1 + Documentation 1.1

Public presentation

Public presentation

Last week (29 April - 3 May)

It acts as a Practical Exam

Participation is mandatory

Each group will present their project to the teachers

The teachers select the presenter(s)



Material to follow the course

Web page: Slides and public information

<https://arquisoft.github.io/>

Virtual campus (internal information)

Learning guide

Forum

Bibliography

Other material

Manuals, Tutorilas, Videos, etc...

This year's assignment



DeChat - Decentralized Chat

<https://labra.solid.community/public/SoftwareArchitecture/AssignmentDescription/index.html>

Inrupt challenge (<https://www.inrupt.com/>)

Inrupt has shown interest in the projects

Certificate of acknowledgement for the best project

Voluntary participation

Event will be on 12th June

It must be based on the Solid platform

About SOLID



SOLID (SOcial LInked Data)

Goal: Decentralized Social Web

Separate personal data from apps

Project started at MIT

It uses several W3C specifications

WebId

Web Access Control

Linked Data Platform

...

You must read/learn about that by yourself

Lots of materials available

If you have questions...

About the course...

Deadlines, exams, mandatory tasks, etc.

Please use the Campus Virtual forum

The message will arrive to the rest of the students

Every one can see the question and the answer

Every one can even answer

About technical matters...

Use public places

Stackoverflow (general): <https://stackoverflow.com/>

Solid forum (about solid): <https://forum.solidproject.org/>

About personal problems or similar questions

Send me an email

Important dates

Assignment deadlines

1st deliverable (19 Feb - 25 Feb.)

2nd deliverable (12 March - 18 March)

3rd deliverable (2 April - 8 April)

4th deliverable **MANDATORY** (29 April - 3 May)

Theory exams:

Ordinary: 30/05/2019, 12:00 – 13:00

Extraordinary: 01/07/2019, 12:00 – 13:00