UNIVERSIDAD POLITÉCNICA DE MADRID
FACULTAD DE INFORMÁTICA
Campus de Montegancedo
Boadilla del Monte. 28660 Madrid

Distributed and Outsourced Software Engineering

Learning Guide – Information for Students

1. Description

Grade	European Master in Software Engineering
Module	N/A
Area	Advanced Software Engineering Aspects
Subject	Distributed and Outsourced Software Engineering
Type	Elective
ECTS credits	4 ECTS
Responsible department	Software Engineering
Major/Section/	N/A

Academic year	2012/2013
Term	1st term
Language	English
Web site	N/A

2. Faculty

NAME and SURNAME	OFFICE	email
Natalia Juristo (Coord.)	D-5104	natalia@fi.upm.es
Oscar Dieste	D-6203	odieste@fi.upm.es

3. Prior knowledge required to take the subject

Passed subjects	N/A
Other required learning outcomes	N/A

4. Learning goals

SUBJECT-SPECIFIC COMPETENCES AND PROFICIENCY LEVEL		
Code	**Competence**	**Level**
CE13	Get exposed to Software Engineering's emergent aspects, and gain deep knowledge of some of them	K
CE14	Understand what actual Software Engineering practices can and cannot achieve, their limitations and potential future evolution	K

Proficiency level: knowledge (K), comprehension (C), application (A), and analysis and synthesis (S)

SUBJECT LEARNING OUTCOMES			
Code	**Learning outcome**	**Related competences**	**Proficiency level**
LR1	The student will be able to plan, manage and execute a distributed software development project	CE13	K
LR2	The student will be able to assess the pros and cons of several technologies for distributed development	CE14	K

Proficiency level: knowledge (K), comprehension (C), application (A), and analysis and synthesis (S)

5. Subject assessment system

ACHIEVEMENT INDICATORS		
Ref	**Indicator**	**Related to LR**
I1	Create a vision document	LR1
I2	Create a SRS document	LR1
I3	Create an API/contract document	LR1
I4	Develop a non-functional version of a software system	LR1
I5	Develop a functional (/tested) version of a software system	LR1
I6	Report, by means of an oral presentation, the project challenges and the way they were solved	LR2

CONTINUOUS ASSESSMENT			
Brief description of assessable activities	**Time**	**Place**	**Weight in grade**
Creation of a vision document	4th Week	Moodle & classroom	10%
Creation of SRS document	6th Week	Moodle & classroom	15%
Creation of an API/contract document	7th Week	Moodle & classroom	25%
Development of a non-functional version of a software system	11th Week	Moodle & classroom	10%
Development of a functional (/tested) version of a software system	13th Week	Moodle & classroom	25%
Reporting the project challenges and the way they were solved	14th – 15th Week	Moodle & classroom	15%
			Total: 100%

GRADING CRITERIA

- Students will be evaluated using the assignments only. Those assignments are particular deliverables of a course's software development project in which students participate.
- The assessment of assignments will depend on (1) presentation made by the students at the classroom and (2) the correctness of the results.
- The final grade will be calculated using a weighted average as described before (for the students who passed the exam).

5. Contents and learning activities

SPECIFIC CONTENTS		
Unit / Topic / Chapter	**Section**	**Related indicators**
Introduction to distributed software engineering	This unit introduces distributed software development, highlighting the differences with traditional development	I6
Requirements for distributed software engineering	This unit introduces the tools and techniques to create requirements specifications created by distributed teams	I1, I2
Design by contract	This unit introduces the approach of “Design by contract”, which eases the development of software created by distributed teams	I3
Development with Eiffel	This unit describes the Eiffel language. Eiffel supports the “Design by contract” approach. This language will be used for the development of the course’s software development project	I5
QA for distributed software engineering	This unit introduces the tools and techniques to perform QA on software systems created by distributed teams	I4

6. Brief description of organizational modalities and teaching methods

TEACHING ORGANIZATION		
Scenario	**Organizational Modality**	**Purpose**
	Theory Classes	***Talk to students***
	Seminars/Workshops	***Construct knowledge through student interaction and activity***
	Practical Classes	***Show students what to do***
	Placements	***Round out student training in a professional setting***
	Personal Tutoring	***Give students personalized attention***
	Group Work	***Get students to learn from each other***
	Independent Work	***Develop self-learning ability***

BRIEF DESCRIPTION OF THE ORGANIZATIONAL MODALITIES AND TEACHING METHODS	
THEORY CLASSES	…
PROBLEM-SOLVING CLASSES	…
PRACTICAL WORK	…
INDIVIDUAL WORK	…
GROUP WORK	…
PERSONAL TUTORING	…

7. Teaching resources

TEACHING RESOURCES	
RECOMMENDED READING	Distributed and outsourced software engineering is a novel topic and no one book provides a good coverage of the topic. Specific readings will be published progressively at moodle.
WEB RESOURCES	Moodle (TBD)
EQUIPMENT	Laboratory: N/A
	Room: 6202
	Group work room: School facilities

8. Subject schedule

Week	Classroom activities	Lab activities	Individual work	Group work	Assessment activities	Others
1-3	Lectures: Introduction to distributed software engineering (6 hours)	-	-	-	-	-
4-6	Lectures: Requirements for distributed software engineering (4 hours) Students: presentation (2 hours)	-	-	Create the vision document (4 hours) Create the SRS document (8 hours)	Presentation in the classroom	-
7	Lectures: Design by contract (2 hours) Students: presentation (2 hours)	-	-	Create the API/contract document (10 hours)	Presentation in the classroom	-
8-11	Lectures: Development with Eiffel (6 hours) Students: presentation (2 hours)	-	-	Develop a non-functional version of a software system (30 hours)	Presentation in the classroom	-
12-13	Lectures: QA for distributed software engineering (4 hours) Students: presentation (2 hours)	-	-	Develop a functional (/tested) version of a software system (15 hours)	Presentation in the classroom	-
14-15	Students: report the project challenges and the way they were solved (4 hours)	-	-	Prepare a presentation (6 hours)	Presentation in the classroom	-
16	Wrap up (1 hours)	-	-	-	-	-

Note: Student workload specified for each activity in hours