



Requirements Engineering

Learning Guide – Information for Students

1. Description

Grade	European Master in Software Engineering
Module	N/A
Area	Software Development
Subject	Requirements Engineering
Type	Compulsory
ECTS credits	6
Responsible department	Software Engineering
Major/Section/	N/A

Academic year	2012/2013
Term	1 st term
Language	English
Web site	http://www.grise.upm.es/UPM_subjects.php?name=SRA



POLITÉCNICA

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

2. Faculty

NAME and SURNAME	OFFICE	email
Oscar Dieste (Coord.)	D-6203	odieste@fi.upm.es
Natalia Juristo	D-5104	natalia@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
CE5	Elicit, analyze and specify the clients, users and other stakeholders' needs, taking into account the possible factors affecting the development of the system.	A

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 elicit and conceptualize the customers' needs	CE5	A
LR2	The student will be able to analyze and validate software requirements	CE5	A
LR3	The student will be able to document software requirements	CE5	A

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	Identify the requirements of an existing software system	LR3
I2	Compose a standard-based software requirements specification	LR3
I3	Perform requirements inspections	LR2
I4	Elicit the requirements of a software system	LR1
I5	Analyze the requirements of a software system	LR1
I6	Create the prototype of a software system	LR1, LR2
I7	Define and apply a change control system	LR3
I8	Compose a model-based software requirements specification & perform a validation using model-checking	LR2, LR3
I9	Compose a formal software requirements specification & perform a validation by mathematical proof	LR2, LR3



CONTINUOUS ASSESSMENT			
Brief description of assessable activities	Time	Place	Weight in grade
<ul style="list-style-type: none">Submission of assignment 1 (Identify the requirements of an existing software system)	1 st Week	Moodle	5%
<ul style="list-style-type: none">Submission of assignment 2 (Compose a standard-based software requirements specification)	2 nd Week	Moodle	10%
<ul style="list-style-type: none">Submission of assignment 3 (Results of a requirements inspection meeting)	3 rd Week	Moodle	10%
<ul style="list-style-type: none">Submission of assignment 4 (Elicit the requirements of a software system)	5 th Week	Moodle	15%
<ul style="list-style-type: none">Submission of assignment 5 (Analyze the requirements of a software system)	8 th Week	Moodle & classroom	15%
<ul style="list-style-type: none">Submission of assignment 6 (Create the prototype of a software system)	9 th Week	Moodle	10%
<ul style="list-style-type: none">Submission of assignment 7 (Define and apply a change control system)	11 th Week	Moodle & classroom	15%
<ul style="list-style-type: none">Submission of assignment 8 (Compose a model-based software requirements specification & perform a validation using model-checking)	13 th Week	Moodle & classroom	10%



CONTINUOUS ASSESSMENT			
Brief description of assessable activities	Time	Place	Weight in grade
<ul style="list-style-type: none">Submission of assignment 9 (Compose a formal software requirements specification & perform a validation by mathematical proof)	15 th Week	Moodle & classroom	10%
			Total: 100%

GRADING CRITERIA
<ul style="list-style-type: none">Students will be evaluated using the assignments.The assessment of assignments will depend on (1) presentation made by the students at the classroom and (2) the correctness of the results.At the end of the term, the students will do an exam. The grade will be fail/pass. It is required to pass the exam in order to pass the course.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
Chapter 1: Introduction to Requirements engineering	1.1. Concepts & importance of requirements engineering 1.2. How to write requirements 1.3. Requirements Engineering process	11, 12
Chapter 2: Specification	2.1. Types of specification 2.2. Properties of a specification 2.3. IEEE-830-1998 2.4. Requirements tools	12
Chapter 3: Validation	3.1. Validation basics 3.2. Reviews & Inspection 3.3. Other validation techniques	13
Chapter 4: Elicitation	4.1. Problems associated to requirements elicitation 4.2. Elicitation techniques 4.3. Moving from elicitation to analysis	14
Chapter 5: Analysis	5.1. Analysis basics 5.2. Weak analysis techniques 5.3. Conceptual modeling	15
Chapter 6: Prototyping	6.1. Purpose of prototypes 6.2. Construction process 6.3. Evaluation of prototypes	16
Chapter 7: Management	7.1. Change control 7.2. Release planning	17
Chapter 8: Model-based specification	8.1. Statecharts 8.2. Model checking 8.3. Tools	18
Chapter 9: Formal	8.1. Z	19



POLITÉCNICA

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

specification	8.2. Formal proof 8.3. Tools	
----------------------	---------------------------------	--



6. Brief description of organizational modalities and teaching methods

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



TEACHING METHODS		
	Method	Purpose
	Explanation/Lecture	<i>Transfer information and activate student cognitive processes</i>
	Case Studies	<i>Learning by analyzing real or simulated case studies</i>
	Exercises and Problem Solving	<i>Exercise, test and practice prior knowledge</i>
	Problem-Based Learning (PBL)	<i>Develop active learning through problem solving</i>
	Project-Oriented Learning (POL)	<i>Complete a problem-solving project applying acquired skills and knowledge</i>
	Cooperative Learning	<i>Develop active and meaningful learning through cooperation</i>
	Learning Contract	<i>Develop independent learning</i>

Known as explanation, this teaching method involves the “*presentation of a logically structured topic with the aim of providing information organized according to criteria suited for the purpose*”. This methodology, also known as *lecture*, mainly focuses on the verbal exposition by the teacher of contents on the subject under study. The term *master class* is often used to refer to a special type of lecture taught by a professor on special occasions

Intensive and exhaustive analysis of a real fact, problem or event for the purpose of understanding, interpreting or solving the problem, generating hypotheses, comparing data, thinking, learning or diagnosis and, sometimes, training in possible alternative problem-solving procedures.

Situations where students are asked to develop the suitable or correct solutions by exercising routines, applying formulae or running algorithms, applying information processing procedures and interpreting the results. It is often used to supplement lectures.

Teaching and learning method whose starting point is a problem, designed by the teacher, that the student has to solve to develop a number of previously defined competences.

Teaching and learning method where have a set time to develop a project to solve a problem or perform a task by planning, designing and completing a series of activities. The whole thing is based on developing and applying what they have learned and making effective use of resources.

Interactive approach to the organization of classroom work where students are responsible for their own and their peers' learning as part of a co-responsibility strategy for achieving group goals and incentives. This is both one of a number of methods for use and an overall teaching approach, or philosophy.

An agreement between the teacher and student on the achievement of learning outcomes through an independent work proposal, supervised by the teacher, and to be accomplished within a set period. The essential points of a learning contract are that it is a written agreement, stating required work and reward, requiring personal involvement and having a time frame for accomplishment.



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	Gerald Kotonya, Ian Sommerville; Requirements Engineering: Processes and Techniques. Wiley 1998.
	Alan Davis; Requirements Engineering: Objects, Functions and States. Prentice-Hall, 2 nd Ed., 1993.
WEB RESOURCES	Subject web site http://www.grise.upm.es/UPM_subjects.php?name=SRA
	Subject Moodle site http://moodle.upm.es/titulaciones/oficiales/course/view.php?id=1574
EQUIPMENT	Laboratory: TBD
	Room: TBD
	Group work room: TBD



8. Subject schedule

Week	Classroom activities	Lab activities	Individual work	Group work	Assessment activities	Others
1	<ul style="list-style-type: none"> Lecture: Chapter 1 (3 hours) 	-	<ul style="list-style-type: none"> Assignment 1: Identify the requirements of an existing software system 	-	<ul style="list-style-type: none"> Grading of assessment 1 	-
2	<ul style="list-style-type: none"> Lecture: Chapter 2 (3 hours) 	-	<ul style="list-style-type: none"> Assignment 2: Compose a standard-based software requirements specification 	-	<ul style="list-style-type: none"> Grading of assessment 2 	-
3	<ul style="list-style-type: none"> Lecture: Chapter 3 (3 hours) 	<ul style="list-style-type: none"> Simulation of a requirements engineering project: Validation (3 hours) 	<ul style="list-style-type: none"> Assignment 3: Results of a requirements inspection meeting (3 hours) 	-	<ul style="list-style-type: none"> Grading of assessment 3 	-
4	<ul style="list-style-type: none"> Lecture: Chapter 4 (3 hours) 	<ul style="list-style-type: none"> Simulation of a requirements engineering project: Elicitation (3 hours) 	<ul style="list-style-type: none"> Assignment 4: Elicit the requirements of a software system (6 hours) 	-	-	-



5	-	<ul style="list-style-type: none">• Simulation of a requirements engineering project: Elicitation (3 hours)	<ul style="list-style-type: none">• Assignment 4: Elicit the requirements of a software system (6 hours)	-	<ul style="list-style-type: none">• Grading of assessment 4	-
6	<ul style="list-style-type: none">• Lecture: Chapter 5 (3 hours)	<ul style="list-style-type: none">• Simulation of a requirements engineering project: Analysis (3 hours)	<ul style="list-style-type: none">• Assignment 5: Analyze the requirements of a software system (3 hours)	-	-	-
7	-	<ul style="list-style-type: none">• Simulation of a requirements engineering project: Analysis (3 hours)	<ul style="list-style-type: none">• Assignment 5: Analyze the requirements of a software system (3 hours)	-	-	-



8	-	<ul style="list-style-type: none">• Presentation of assignment 5 (3 hours)• Simulation of a requirements engineering project: Elicitation (3 hours)	<ul style="list-style-type: none">• Improvement of assignment 4 (3 hours)	-	<ul style="list-style-type: none">• Grading of assessment 5	-
9	<ul style="list-style-type: none">• Lecture: Chapter 6 (3 hours)	<ul style="list-style-type: none">• Simulation of a requirements engineering project: Prototyping (3 hours)	<ul style="list-style-type: none">• Assignment 6: Create the prototype of a software system (3 hours)	-	<ul style="list-style-type: none">• Grading of assessment 6	-
10	<ul style="list-style-type: none">• Lecture: Chapter 7 (3 hours)	<ul style="list-style-type: none">• Simulation of a requirements engineering project: Management (3 hours)	<ul style="list-style-type: none">• Assignment 7: Define and apply a change control system (6 hours)	-	-	-



11	-	<ul style="list-style-type: none"> • Presentation of assignment 7 (3 hours) 	<ul style="list-style-type: none"> • Extra work -- Definition of a Software Requirements Specification for the simulated software system (6 hours) 	<ul style="list-style-type: none"> • Extra work -- Validation of the Software Requirements Specification for the simulated software system (6 hours) 	<ul style="list-style-type: none"> • Grading of assessment 7 	-
12	<ul style="list-style-type: none"> • Lecture: Chapter 8 (3 hours) 	-	<ul style="list-style-type: none"> • Assignment 8: Compose a model-based software requirements specification & perform a validation using model-checking (6 hours) 	<ul style="list-style-type: none"> • Perform the validation of the specification (6 hours) 	-	-
13	-	<ul style="list-style-type: none"> • Presentation of assignment 8 (3 hours) 	-	-	<ul style="list-style-type: none"> • Grading of assessment 8 	-
14	Lecture: Chapter 9 (3 hours)	-	<ul style="list-style-type: none"> • Assignment 9: Compose a formal software requirements specification & perform a validation by mathematical proof (6 hours) 	<ul style="list-style-type: none"> • Perform the validation of the specification (6 hours) 	-	-
15	-	<ul style="list-style-type: none"> • Presentation of assignment 9 (3 hours) 	-	-	<ul style="list-style-type: none"> • Grading of assessment 9 	-
16	<ul style="list-style-type: none"> • Wrap up (2 hours) 	-	-	-	<ul style="list-style-type: none"> • Final test (6 hours) 	-

Note: Student workload specified for each activity in hours