Experimental Software Engineering

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

Experimental Software Engineering

Learning Guide – Information for Students

1. Description

Grade	European Master in Software Engineering
Module	Advanced Software Engineering Aspects
Area	N/A
Subject	Experimental Software Engineering
Type	Elective
ECTS credits	6
Responsible department	Software Engineering
Major/Section/	N/A

Academic year	2010/2011
Term	2nd term
Language	English
Web site	http://www.grise.upm.es/UPM_subjects.php?name=ESE

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	To have a vision of the different specific and emergent aspects of the Software Engineering, and to go further in some of them.	S
CE14	To understand what nowadays software engineering procedures can and cannot reach, their limitations and their possible future evolution.	S

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

SUBJECT LEARNING OUTCOMES			
Code	**Learning outcome**	**Related competences**	**Proficiency level**
LR1	Given a particular software engineering field, the student will be able to design and evaluate the most adequate approach to solve some of the related problems, highlighting the technical difficulties and limits of application.	CE13, C14	S

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	Know the different elements of a software engineering experiment	LR1
I2	Understand the report of a software engineering experiment	LR1
I3	Design close and differentiated replications	LR1
I4	Apply basic statistical concepts (i.e.: t-test, ANOVA, sample size calculation)	LR1
I5	Perform a systematic literature review	LR1
I6	Apply different aggregation mechanisms to software engineering experiments (i.e.: meta-analysis)	LR1

CONTINUOUS ASSESSMENT			
Brief description of assessable activities	**Time**	**Place**	**Weight in grade**
- Presentation of assignment 1 (Understanding a software engineering experiment) in the classroom. - Submission of assignment 1 at moodle.	2nd Week	Classroom & moodle	20%
- Presentation of assignment 2 (Design a differentiated replication) in the classroom. - Submission of assignment 2 at moodle.	4th week	Classroom & moodle	20%

CONTINUOUS ASSESSMENT			
Brief description of assessable activities	**Time**	**Place**	**Weight in grade**
Presentation of assignment 3 (Data analysis) in the classroom. Submission of assignment 3 at moodle.	8th week	Classroom & moodle	20%
Presentation of assignment 4 (Perform a systematic literature review) in the classroom. Submission of assignment 4 at moodle.	10th, 11th and 12th weeks	Classroom & moodle	20%
Presentation of assignment 5 (Aggregate software engineering experiments) in the classroom. Submission of assignment 5 at moodle.	14th, 15th weeks	Classroom & moodle	20%
			Total: 100%

GRADING CRITERIA

- Students will be evaluated using the assignments only. No examination will be made.
- 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.

5. Contents and learning activities

SPECIFIC CONTENTS		
Unit / Topic / Chapter	**Section**	**Related indicators**
Chapter 1: Introduction to experimental software engineering	1.1. Basics of empiricism 1.2. Types of empirical studies 1.3. Experiments: Structure and design 1.4. Reporting experiments	I1, I2
Chapter 2: Replication of software engineering experiments	2.1. Concept of replication 2.2. Close and differentiated replications 2.3. Design of replicated experiments	I3
Chapter 3: Data analysis	3.1. Basics of inferential statistics 3.2. Procedures to compare 2 means: t, Mann-Whitney 3.3. Procedures to compare k means: ANOVA 3.4. Statistical power	I4
Chapter 4: Systematic reviews in software engineering	4.1. Purpose of systematic reviews 4.2. Stages of a review - Plan - Search - Assess quality - Extract data - Synthesis of results	I5
Chapter 5: Aggregation of software engineering experiments	5.1. Strength of evidence in software engineering experiments 5.2. Aggregation procedures - Statistical methods (meta-analysis) - Non-statistical procedures	I6

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***

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

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

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	Lectures used by the instructors to introduce the relevant concepts, provide explanations, small examples, etc. Interaction with students is restricted to brief questions (made either by the student or the instructor).
PRACTICAL WORK	Workshops carried out at the classroom where the assignments are worked out by all students together. Interaction and discussion is leaded by the instructor. The goal is to review the different solutions and reconcile the students' perspectives on the subject matter.
INDIVIDUAL WORK	Work carried out by the student individually, such as reading papers, solving exercises, etc. This work is usually performed at home.
GROUP WORK	Work carried out by the student along with some peers. The type of work is usually the same than in the individual case, but the group work helps to improve the final results and develop workgroup skills.
PERSONAL TUTORING	Students may ask for an interview with the instructors to pose questions or raise problems with the course. Those are carried out during office hours.

7. Teaching resources

TEACHING RESOURCES	
RECOMMENDED READING	Natalia Juristo, Ana Moreno; Basics of software engineering experimentation. Kluwer 2001.
	Claes Wohlin et al.; Experimentation in software engineering: An introduction. Kluwer 2000.
WEB RESOURCES	Subject web site http://www.grise.upm.es/UPM_subjects.php?name=ESE
	Subject Moodle site http://moodle.upm.es/titulaciones/oficiales/course/view.php?id=1575
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 (6 hours)	□ Lecture: Chapter 1 (3 hours)	-	-	□ Assignment 1: Understanding a software engineering experiment (3 hours)	-	-
2 (15 hours)	□ Lecture: Chapter 1 (3 hours)	-	□ Assignment 1: Understanding a software engineering experiment (9 hours)	□ Assignment 1: Understanding a software engineering experiment (3 hours)	-	
3 (6 hours)	□ Discussion: assignment 1 (3 hours)	-	□ Assignment 1: Amendment (3 hours)		□ Presentation of assignment 1	-
4 (15 hours)	□ Lecture: Chapter 2 (3 hours)	-	□ Assignment 2: Design a differentiated replication (6 hours)	□ Assignment 2: Design a differentiated replication (6 hours)	-	-

5 (6 hours)	☐ Discussion: assignment 2 (3 hours)	-	☐ Assignment 2: Amendment (3 hours)		☐ Presentation of assignment 2	-
6 (9 hours)	☐ Lecture: Chapter 3 (3 hours)	-	☐ Assignment 3: Data analysis part 1 (6 hours)	-	-	-
7 (9 hours)	☐ Lecture: Chapter 3 (3 hours)	-	☐ Assignment 3: Data analysis part 2 (6 hours)	-	-	-
8 (3 hours)	☐ Discussion: assignment 3 (3 hours)	-	-		☐ Presentation of assignment 3	-
9 (15 hours)	☐ Lecture: Chapter 4 (3 hours)	-	☐ Assignment 4: Perform a systematic literature review part 1 (9 hours)	☐ Assignment 4: Perform a systematic literature review part 1 (3 hours)	-	-
10 (18 hours)	☐ Discussion: assignment 4 part 1 (3 hours)	-	☐ Assignment 4: Perform a systematic literature review part 2 (12 hours)	☐ Assignment 4: Perform a systematic literature review part 2 (3 hours)	☐ Presentation of assignment 4 part 1	-

11 (12 hours)	☐ Discussion: assignment 4 part 2 (3 hours)	-	☐ Assignment 4: Perform a systematic literature review part 3 (6 hours)	☐ Assignment 4: Perform a systematic literature review part 3 (3 hours)	☐ Presentation of assignment 4 part 2	-
12 (3 hours)	☐ Discussion: assignment 4 part 3 (3 hours)	-	-	-	☐ Presentation of assignment 4 part 3	-
13 (12 hours)	☐ Lecture: Chapter 5 (3 hours)	-	☐ Assignment 5: Aggregate software engineering experiments part 1 (6 hours)	☐ Assignment 5: Aggregate software engineering experiments part 1 (3 hours)	-	-
14 (9 hours)	☐ Discussion: assignment 5 part 1 (3 hours)	-	☐ Assignment 5: Aggregate software engineering experiments part 2 (6 hours)	-	☐ Presentation of assignment 5 part 1	-
15 (3 hours)	☐ Discussion: assignment 5 part 2 (3 hours)	-	-		☐ Presentation of assignment 5 part 2	-
16 (3 hours)	☐ Wrap-up (3 hours)	-	-	-	-	-
144 total hours						