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

(Verification and Validation)

Learning Guide – Information for Students

1. Description

Grade	Máster Universitario en Ingeniería del Software – European Master on Software Engineering
Module	Support processes
Area	
Subject	Verification and Validation
Type	Compulsory
ECTS credits	6
Responsible department	Lenguajes y Sistemas Informáticos e Ingeniería el Software
Major/Section/	

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

2. Faculty

NAME and SURNAME	OFFICE	email
Sira Vegas (Coord.)	5105	svegas@fi.upm.es
Natalia Juristo	5104	natalia@fi.upm.es

3. Prior knowledge required to take the subject

Passed subjects	•
Other required learning outcomes	•

4. Learning goals

SUBJECT-SPECIFIC COMPETENCES AND PROFICIENCY LEVEL		
Code	**Competence**	**Level**
SC7	Produce a verification and validation plan to coordinate and prioritize resources and activities to assess the required quality level.	S
SC8	Apply the most appropriated verification and validation techniques for a software development project, within the framework of a verification and validation plan	S

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

SUBJECT LEARNING OUTCOMES			
Code	**Learning outcome**	**Related competences**	**Profi-ciency level**
LR1	Knows and determines the most appropriate verification and validation techniques to be applied in a software development project with the aim of assuring the quality level required	GC2, GC12	S
LR2	Knows and applies quality control techniques to products and processes	GC1, GC12, SC1, SC6, SC7	S

5. Subject assessment system

ACHIEVEMENT INDICATORS		
Ref	**Indicator**	**Related to LR**
I1	Analysis of the defects found with different software evaluation techniques	LR1
I2	Application of several software evaluation techniques	LR2

(Optionally, use rubric table instead)

CONTINUOUS ASSESSMENT			
Brief description of assessable activities	**Time**	**Place**	**Weight in grade**
Application of a static evaluation technique in small scale	Week 2-4	Home	1%
Application of “ad-hoc” testing in small scale	Week 5	Home	1%
Application of a first testing technique in small scale	Week 8	Home	1%
Application of a second testing technique in small scale	Week 9	Home	1%
Application of a static evaluation technique in large scale	Week 10	Class	30%
Application of a first testing technique in large scale	Week 11	Class	30%
Application of a second testing technique in large scale	Week 12	Class	30%
Analysis of the defects found applying evaluation techniques in large scale	Week 13-16	Class/Home	5%
Class participation	Week 1-16	Class	1%
			Total: 100%

GRADING CRITERIA

The grading of students is calculated based on their performance in the different assignments:

- Application of three code evaluation techniques and “ad-hoc” testing in the small.
- Application of three code evaluation techniques in the large.
- Analysis of the defects found with the application of techniques in the large.

The participation of the student during the classes will be also taken into consideration.

6. Contents and learning activities

SPECIFIC CONTENTS		
Unit / Topic / Chapter	**Section**	**Related indicators**
Chapter 1: Introduction	1.1 Introduction to V&V	I2
	1.2 V&V and the software development process	I2
Chapter 2: Static Evaluation	2.1 Introduction to static evaluation	I1, I2
	2.2 Static evaluation techniques	I1, I2
	2.3 Reading techniques	I1, I2
Chapter 3: Dynamic Evaluation: Testing	3.1 Introduction to software testing	I1, I2
	3.2 Testing levels	I1, I2
	3.3 Software testing techniques	I1, I2
	3.4 The software testing process	I1, I2
	3.5 Testing tools	I1, I2
	3.6 The Software Verification and Validation plan	I1, I2
Chapter 4: Testing Technique Selection	4.1 Introduction	I1
	4.2 Different techniques for different fault types	I1
	4.3 Complementariness of techniques	I1

7. Brief description of organizational modalities and teaching methods

TEACHING ORGANIZATION		
Scenario	**Organizational Modality**	**Purpose**
X	**Theory Classes**	***Talk to students***
	Seminars/Workshops	***Construct knowledge through student interaction and activity***
X	**Practical Classes**	***Show students what to do***
	Placements	***Round out student training in a professional setting***
	Personal Tutoring	***Give students personalized attention***
X	**Group Work**	***Get students to learn from each other***
X	**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**
X	**Explanation/Lecture**	***Transfer information and activate student cognitive processes***
X	**Case Studies**	***Learning by analyzing real or simulated case studies***
X	**Exercises and Problem Solving**	***Exercise, test and practice prior knowledge***
X	**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***
X	**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 will have a theoretical part, where main concepts of the course are presented by the professor
PROBLEM-SOLVING CLASSES	The student will be asked to exercise these main concepts by means of problems that will be solved in the class. This will help to reinforce the concepts and help the student to get acquainted with them.
PRACTICAL WORK	The student will be asked to perform several assignments during the course.
INDIVIDUAL WORK	Theory classes have to be complemented with individual study from the student side. Additionally, some of the practical work will have to be performed individually.
GROUP WORK	There will be some group work in the last assignment (defect analysis), where the students have to discuss about their findings with other students.
PERSONAL TUTORING	At any moment, the student will be monitored by the professors, giving any required mentoring related to the topics that are discussed in the course.

8. Teaching resources

TEACHING RESOURCES	
RECOMMENDED READING	1. B. Beizer. "Software Testing Techniques" 2ª Edición. 1990
	2. G. J. Myers. "The Art of Software Testing" 2ª Edición. Wiley. 2004.
	3. R.G. Pressman. Software Engineering: A practitioner's approach. Quinta Edición. McGrawHill. 2004.
	4. I. Sommerville. Software Engineering. Octava edición. Addison-Wesley. 2006.
	5. P.C. Jorgensen. Software Testing. A Craftsman's Approach. CRC Press, 1995.
	6. C. Kaner, J. Falk, H.Q. Nguyen. Testing Computer Software. Wiley, 1999.
	7. W.E. Perry. Effective methods for software testing. Tercera edición. Wiley. 2006.
	8. S.L. Pfleeger. Ingeniería de software: teoría y práctica. Segunda edición. Prentice Hall. 2002.
	9. IEEE standards: [IEEE-1012, 1998], [IEEE-1008, 1987] [IEEE-829, 1998] [IEEE-1044, 1993] [IEEE-1044.1, 1995]
WEB RESOURCES	Subject web site (http://)
	Subject Moodle site (http://)
EQUIPMENT	Laboratory
	Room XXXX
	Group work room

9. Subject schedule

Week	Classroom activities	Lab activities	Individual work	Group work	Assessment activities	Others
Week 1 (6 hours)	• Introduction (3 hours)	•	• Individual study (3 hours)	•	•	•
Week 2 (9 hours)	• Static techniques (3 hours)	•	• Individual study (3 hours) • Exercises and problem solving (3 hours)	•	•	•
Week 3 (6 hours)	• Static techniques (3 hours)	•	• Individual study (3 hours)	•	•	•
Week 4 (12 hours)	• Static techniques (3 hours)	•	• Individual study (3 hours) • Exercises and problem solving (6 hours)	•	•	•
Week 5 (12 hours)	• Dynamic techniques (3 hours)	•	• Individual study (3 hours) • Exercises and problem solving (6 hours)	•	•	•
Week 6 (9 hours)	• Dynamic techniques (3 hours)	•	• Individual study (6 hours)	•	•	•
Week 7 (9 hours)	• Dynamic techniques (3 hours)	•	• Individual study (6 hours) •	•	•	•

Week	Classroom activities	Lab activities	Individual work	Group work	Assessment activities	Others
Week 8 (12 hours)	• Dynamic techniques (3 hours)	•	• Individual study (3 hours) • Exercises and problem solving (6 hours)	•	•	•
Week 9 (12 hours)	• Dynamic techniques (3 hours)	•	• Individual study (3 hours) • Exercises and problem solving (6 hours)	•	•	•
Week 10 (13 hours)	•	•	• Individual study (10 hours)	•	• Application of techniques (3 hours)	•
Week 11 (13 hours)	•	•	• Individual study (10 hours)	•	• Application of techniques (3 hours)	•
Week 12 (13 hours)	•	•	• Individual study (10 hours)	•	• Application of techniques (3 hours)	•
Week 13 (12 hours)	• Technique selection (3 hours)	•	• Individual study (3 hours) • Learning contract (6 hours)	•	•	•
Week 14 (12 hours)	• Technique selection (3 hours)	•	• Individual study (3 hours) • Learning contract (6 hours)	•	•	•
Week 15 (3 hours)	• Technique selection (2,75 hours)	•	•	•	• Presentation (15 min.)	•
Week 16 (3 hours)	• Technique selection (2,75 hours)	•	•	•	• Presentation (15 min.) •	•

Note: Student workload specified for each activity in hours