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

Software Quality Management

Learning Guide – Information for Students

1. Description

Grade	Máster Universitario en Ingeniería de Software - European Master on Software Engineering
Module	Support Processes
Area	
Subject	Software Quality Management
Type	Mandatory
ECTS credits	4
Responsible department	Computer Languages and Systems and Software Engineering - Lenguajes y Sistemas Informáticos e Ingeniería del Software
Major/Section/	

Academic year	2012/2013
Term	2nd term
Language	English
Web site	

2. Faculty

NAME and SURNAME	OFFICE	email
Angélica de Antonio Jiménez (Coord.)	5108	angelica@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**
SC1	To perform a project plan to coordinate and prioritize resources and activities, in order to obtain the expected results within the deadlines, costs and quality required.	A
SC6	To design the unit tests and to learn to design integration and implantation tests. To perform the system integration, integration tests and implantation of the system.	S
SC7	To produce a verification and validation plan to coordinate and prioritize resources and activities to assess the required quality level.	A
SC11	To identify, control, inform and audit a system configuration and its changes.	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	Knows and applies quality models to identify and specify the quality attributes a software system must satisfy	SC1	S
LR2	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	SC7	A
LR3	Is able to identify and determine the practices needed to manage a software system configuration	SC11	S
LR4	Understands the mission of a quality system and knows the applicable standards and norms	SC10	A
LR5	Understands the interrelation between product quality and process quality	SC10	S
LR6	Knows and applies product and process quality control techniques	SC1, SC7	A

5. Subject assessment system

ACHIEVEMENT INDICATORS		
Ref	**Indicator**	**Related to LR**
I1	Understand the meaning of quality and of specific software quality attributes included in a software quality model	LR1
I2	Is able to evaluate the degree to which a system satisfies relevant quality attributes starting from collected quality metrics, and to identify significant deviations	LR1, LR6
I3	Is able to specify quality requirements for a system under development	LR1
I4	Knows and understands the range of possible verification and validation techniques to be applied in a software development project, their features, advantages and drawbacks from the point of view of quality control	LR2, LR5
I5	Is able to determine, from the results of verification and validation activities, the degree of satisfaction of relevant quality attributes and the need of corrective actions	LR2, LR6
I6	Understands the impact on quality of a proper software configuration management	LR3, LR5
I7	Is able to identify properly the software configuration items to be maintained under control in a software development project	LR3
I8	Is able to manage change requests on a software product during the development and maintenance phases, and identify their impact on the product configuration	LR3
I9	Knows the different audits to be performed along the software development process and their impact on software configuration status	LR3
I10	Knows and is able to interpret the meaning of significant product and process quality metrics and quantitatively compare quality prevention and correction activities	LR4, LR5
I11	Understands the quality assurance function and its role in a project	LR4

ACHIEVEMENT INDICATORS		
Ref	**Indicator**	**Related to LR**
I12	Understands the mission of a quality system and knows the applicable standards and norms	LR4
I13	Understands the interrelation between product quality and process quality, is able to analyze product and process metrics, and is able to identify process improvement actions that can have an significant impact on software quality	LR5, LR6

CONTINUOUS ASSESSMENT			
Brief description of assessable activities	**Time**	**Place**	**Weight in grade**
Individual exercises	Weeks 2-16	Classroom and Moodle	5%
Group exercises	Weeks 2-16	Classroom	10%
Practical work	Weeks 8-16	Group work with weekly meetings in the classroom	35%
Partial and final exams	Weeks 4,8,12 and 16	Classroom	40%
Student implication and participation	Weeks 1-16	Classroom and Moodle	10%

GRADING CRITERIA

The subject is marked following continuous assessment.

The student passes the subject only if 5 or more points on 10 are obtained at the end of the course, regarding the following criteria:

FINAL GRADE = 5% Individual exercises in the classroom and Moodle + 10% Group exercises in the classroom + 35% Practical work + 40% Exams + 10% Student participation

The final grade will be obtained from five components: (1) individual exercises and (2) group exercises performed in the classroom; (3) a practical work involving quality measurement, control and management; (4) partial and final exams to test conceptual knowledge; and (5) participation and implication of the student in the subject.

The maximum grade for each of these components and the minimum mark needed to compensate non-passed parts are indicated in the following table.

	MAXIMUM GRADE (and correspondence over the final grade)	**MINIMUM GRADE TO COMPENSATE NON-PASSED PARTS** (and correspondence over the final grade)
Individual exercises in the classroom and Moodle (5%)	10 (0,5)	-
Group exercises in the classroom (10%)	10 (1,0)	-
Practical work (35%)	10 (3,5)	4 (1,4)
Exams (40%)	10 (4,0)	4 (1,6)
Student participation (10%)	10 (1)	-

When failed, the practical work and the exams can be repeated in the extra exam period, using the new marks together to the ones obtained in individual and group exercises in the classroom and student participation in the previous period to calculate the final grade of the subject.

6. Contents and learning activities

SPECIFIC CONTENTS		
Unit / Topic / Chapter	**Section**	**Related indicators**
Chapter 1: Introduction to Software Quality	1.1 Software Quality Definition	I1
	1.2 Software Quality Models	I1
	1.3 Usage of a Quality Model	I1, I2, I3
	1.4 Defects and Defect Density	I1, I2, I3
Chapter 2: Software Quality Control Activities	2.1 Static Controls	I4, I5
	2.2 Dynamic Controls	I4, I5
Chapter 3: Quality Metrics	3.1 Product and Process Quality Metrics	I10
	3.2 Comparison of Quality Control Activities	I4, I5, I10
Chapter 4: Software Configuration Management	4.1 Basic Concepts of Software Configuration Management	I6
	4.2 Configuration Identification	I7
	4.3 Configuration Change Control	I8
	4.4 Configuration Status Accounting	I7, I8, I9
	4.5 Configuration Audits	I9
	4.6 Configuration Management Plan	I7, I8, I9
Chapter 5: Software Quality Assurance Activities	5.1 Introduction to Quality Assurance	I11
	5.2 Quality Construction Activities	I11
	5.3 The Cost of Quality	I10, I11
	5.4 Quality Assurance Planning	I11
Chapter 6: Quality Management and Quality Systems	6.1 Introduction to Quality Management and related standards and norms	I12
	6.2 The Quality System and the Quality Manual	I12

Chapter 7: Process Quality and Product Quality	7.1 Software Development Process Maturity and Capability	I13
	7.2 Process Improvements at personal, team and organizational levels	I13

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***
X	**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***
	Case Studies	***Learning by analyzing real or simulated case studies***
X	**Exercises and Problem Solving**	***Exercise, test and practice prior knowledge***
	Problem-Based Learning (PBL)	***Develop active learning through problem solving***
X	**Project-Oriented Learning (POL)**	***Complete a problem-solving project applying acquired skills and knowledge***
X	**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	Some theoretical lectures will be arranged during the course to present basic concepts and key aspects, always supported by audiovisual resources and innovative techniques to enhance student comprehension
PROBLEM-SOLVING CLASSES	Professor and students will solve problems in the classroom to apply and fix the knowledge acquired during the theory classes
PRACTICAL WORK	Group work is complemented with practical classes in which students have to deal with problems and challenges as close as possible to real life developments
INDIVIDUAL WORK	Students will have to do individual works, some in the classroom and some out of classroom time. The professor will provide the instructions to complete them
GROUP WORK	Two kinds of group works are planned during the course: short sessions in the classroom; and a practical work to be carried out for several weeks starting from a problem. Instructions will be provided in both cases to students by the professor
PERSONAL TUTORING	Students will be able to attend personal tutoring, following the procedure established at the School

8. Teaching resources

TEACHING RESOURCES	
RECOMMENDED READING	Gordon Schulmeyer, G. (2007) Handbook of Software Quality Assurance, Artech House Publishers, 4th ed.
	A.F. Ackerman, L.S. Buchwald, F.H. Lewski, Software Inspections: An Effective Verification Process, IEEE Software, Vol. 6, Nº 3, pp. 31-36, Mayo, 1989
	J.R. Evans, W.M. Lindsay (2001) The Management and Control of Quality, South-Western, 5th ed.
	M.E. Fagan, Design and Code Inspections to Reduce Errors in Program Development, IBM Systems Journal, Vol. 15, Nº 3, pp. 182-210, 1976
	C.A. Cianfrani, J.J. Tsiakals, J.E. West (2009) ISO 9001:2008 Explained, ASQ Quality Press
	S.H. Kan (2002) Metrics and Models in Software Quality Engineering, Addison-Wesley, 2nd ed.
	D. Galin (2003) Software Quality Assurance: From Theory to Implementation, Addison-Wesley
	J.W. Horch (1996) Practical Guide to Software Quality Management, Artech House Publishers
	J.A. McCall, P.K. Richards,.G.F. Walters, Factors in Software Quality, RADC-TR-77-369, Rome Air Development Center, United States Air Force, 1977
	Weinberg, G.M., Freedman, D.P., Reviews, Walkthroughs and Inspections, IEEE Transactions on Software Engineering, Vol. SE-10, nº 1, pp. 68-72, Ene. 1984
	[ESA, PSS-05-11], European Space Agency, ESA Guide to Software Quality Assurance, ESA Software Engineering Standards Issue 2, Febrero 1991
	[ESA, PSS-05-10], European Space Agency, ESA Guide to Software Verification and Validation, ESA Software Engineering Standards Issue 2, Febrero 1991

[IEEE, 1298-92], IEEE Standard Software Quality Management System. Part 1: requirements, ANSI/IEEE std. 1298-1992, IEEE Computer Society, Software Engineering Technical Committee, Software Engineering Standards Subcommittee, 1992

[IEEE, 1044-93], IEEE Standard Classification for Software Anomalies, ANSI/IEEE Std.1044-1993, IEEE Computer Society, Software Engineering Technical Committee, Software Engineering Standards Subcommittee, 1993

[IEEE, 1044.1-95], IEEE Guide to Classification for Software Anomalies, ANSI/IEEE Std.1044.1-1995, IEEE Computer Society, Software Engineering Technical Committee, Software Engineering Standards Subcommittee, 1995

[IEEE, 1061-92], IEEE Standard for a Software Quality Metrics Methodology, ANSI/IEEE std. 1061-1992, IEEE Computer Society, Software Engineering Technical Committee, Software Engineering Standards Subcommittee ,1992

[IEEE, 1045-92], IEEE Standard for a Software Productivity Metrics, IEEE Std 1045-1992, IEEE Computer Society, Software Engineering Technical Committee, Software Engineering Standards Subcommittee,1992

[IEEE,730-89], IEEE Standard for Software Quality Assurance Plans, ANSI/IEEE std. 730-1989, IEEE Computer Society, Software Engineering Technical Committee, 1989

[IEEE, 1028-88], IEEE Standard for Software Reviews and Audits, ANSI/IEEE std. 1028-1988, IEEE Computer Society, Software Engineering Technical Committee, Software Engineering Standards Subcommittee, 1988

[IEEE, 1008-87], IEEE Standard for Software Unit Testing, ANSI/IEEE std. 1008-1987, IEEE Computer Society, Software Engineering Technical Committee, 1986

[IEEE, 1012-86], IEEE Standard for Software Verification and Validation Plans, ANSI/IEEE std. 1012-1986, IEEE Computer Society, Software Engineering Technical Committee, 1986

[IEEE, 983-86], IEEE Guide for Software Quality Assurance Planning, ANSI/IEEE std. 983-1986, IEEE Computer Society, Software Engineering Technical Committee, Software Engineering Standards Subcommittee, 1986

	[IEEE, 829-83], IEEE Standard for Software Test Documentation, ANSI/IEEE std. 829-1983, IEEE Computer Society, Software Engineering Technical Committee, 1983
	ISO/IEC 9126-1:2001, Software engineering -- Product quality -- Part 1: Quality model
	ISO/IEC TR 9126-2:2003, Software engineering -- Product quality -- Part 2: External metrics
	ISO/IEC TR 9126-3:2003, Software engineering -- Product quality -- Part 3: Internal metrics
	ISO/IEC TR 9126-4:2004, Software engineering -- Product quality -- Part 4: Quality in use metrics
WEB RESOURCES	Subject Moodle site (http://moodle.upm.es/titulaciones/oficiales/course/view.php?id=2999
EQUIPMENT	Room 6201

9. Subject schedule

Week	Classroom activities	Lab activities	Individual work	Group work	Assessment activities	Others
Week 1 (5 hours)	Chapter 1 1.1 Software Quality Definition 1.2 Software Quality Models (2 hours)		• Individual study(1 hour)	• Exercise on quality attributes (2 hours)	• Class participation	•
Week 2 (6 hours)	Chapter 1 1.3 Usage of a Quality Model (1 hour)	•	• Individual study(2 hours) • Individual exercise on quality model usage (2 hours)	•	• Class participation • Presentation of group work (1 hour) • Evaluation of individual exercise	•
Week 3 (7 hours)	Chapter 1 1.4 Defects and Defect Density (1 hour)	•	• Individual study(2 hours)	• Practical group work (3 hours)	• Class participation • Presentation of group work (1 hour)	•
Week 4 (7 hours)	Chapter 2 2.1 Static Controls (1 hour)	•	• Individual study (1 hour)	• Exercise on inspections (3 hours) • Practical group work (2 hours)	• Class participation • Evaluation of group exercise	•

Week 5 (7 hours)	Chapter 2 2.1 Static Controls (1 hour)	•	• Individual study (1 hour)	• Exercise on walkthroughs (3 hours) • Practical group work (2 hours)	• Class participation • Evaluation of group exercise	•
Week 6 (7 hours)	Chapter 2 2.2 Dynamic Controls (2 hours)	•	• Individual study (2 hour) • Individual exercises on testing (3 hours)	•	• Class participation • Evaluation of individual exercises	•
Week 7 (7 hours)	Chapter 3 3.1 Basic Concepts of Software Configuration Management 3.2 Configuration Identification (2 hours)	•	• Individual study(1 hour) • Individual exercises on configuration identification (1 hour)	• Practical group work (3 hours)	• Class participation • Evaluation of individual exercises	•
Week 8 (7 hours) (	Chapter 3 3.3 Configuration Change Control 3.4 Configuration State Reports 3.5 Configuration Audits 3.6 Configuration Management Plan (2 hours)	•	• Individual study(2 hours)	• Practical group work (3 hours)	• Class participation	•
Week 9 (7 hours)	Partial exam (2 hours)	•	• Individual study (5 hours)	•	• Partial exam	•

Week 10 (7 hours)	Chapter 4 4.1 Product and Process Quality Metrics 4.2 Comparison of Quality Control Activities (2 hours)	•	• Individual study (1 hour) • Individual exercises on metrics (1 hour)	• Practical group work (3 hours)	• Class participation • Evaluation of individual exercises	•
Week 11 (8 hours)	Chapter 5 5.1 Introduction to Quality Assurance 5.2 Quality Construction Activities 5.3 The Cost of Quality 5.4 Quality Assurance Planning (2 hours)	•	• Individual study(2 hours)	• Practical group work (4 hours)	• Class participation	•
Week 12 (6 hours)	Chapter 6 6.1 Introduction to Quality Management and related standards and norms 6.2 The Quality System and the Quality Manual (1 hour)	•	• Individual study (2 hours)	• Group exercise on quality standards and norms (2 hours)	• Class participation • Presentation of group work (1 hour)	•

Week 13 (7 hours)	Chapter 7 7.1 Software Development Process Maturity and Capability (2 hours)	•	• Individual study (1 hours)	• Practical group work (4 hours)	• Class participation	•
Week 14 (6 hours)	Chapter 7 7.2 Process Improvements at personal, team and organizational levels (2 hours)	•	• Individual study (1 hour)	• Group exercise on process improvement (3 hours)	• Class participation • Evaluation of group exercise	•
Week 15 (7 hours)	• Final exam (2 hours)	•	• Individual study (5 hours)	•	• Final exam	•

Note: Student workload specified for each activity in hours