

## BASIC DETAILS:

<b>Subject:</b>	CALIDAD DEL SOFTWARE		
<b>Id.:</b>	30492		
<b>Programme:</b>	GRADUADO EN INGENIERÍA INFORMÁTICA (ITINERARIO DE ADAPTACIÓN)		
<b>Module:</b>	INGENIERÍA DEL SOFTWARE		
<b>Subject type:</b>	OBLIGATORIA		
<b>Year:</b>	3	<b>Teaching period:</b>	Segundo Cuatrimestre
<b>Credits:</b>	3	<b>Total hours:</b>	75
<b>Classroom activities:</b>	11,5	<b>Individual study:</b>	63,5
<b>Main teaching language:</b>	Castellano	<b>Secondary teaching language:</b>	Inglés
<b>Lecturer:</b>		<b>Email:</b>	

## PRESENTATION:

Uno de los problemas más acuciantes que afronta actualmente el mundo de la ingeniería informática es la producción de software de calidad. Desde la década de los 70 este tema ha sido motivo de preocupación para especialistas, ingenieros, investigadores y fabricantes de software, los cuales han realizado gran cantidad de investigaciones al respecto con el objetivo fundamental de obtener un software de calidad a través de una serie de procesos basados en el establecimiento de métricas, métodos de evaluación, seguimiento y realimentación en el ciclo de vida de desarrollo.

Se puede definir la calidad del software como el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad. La calidad del software es medible y varía de un sistema a otro o de un programa a otro. Un software elaborado para el control de naves espaciales debe ser confiable al nivel de "cero fallos"; un software hecho para ejecutarse una sola vez no requiere el mismo nivel de calidad; mientras que un producto de software para ser explotado durante un largo período (10 años o más), necesita ser confiable, mantenible y flexible para disminuir los costes de mantenimiento y perfeccionamiento durante el tiempo de explotación.

La obtención de un *software* con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del *software* que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, menor mantenimiento y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del *software*. El objetivo de la asignatura *Calidad del Software* es dotar al alumno de las competencias necesarias para medir, controlar y garantizar el desarrollo de software bajo unos estándares de calidad preestablecidos.

## PROFESSIONAL COMPETENCES ACQUIRED IN THE SUBJECT:

<b>General programme competences</b>	G02	Capacidad innovadora para proponer y encontrar formas nuevas y eficaces de realizar cualquier tarea y/ o función dentro de su entorno profesional con una elevada motivación por la calidad
	G03	Capacidad para trabajar dentro de equipos multidisciplinares para conseguir metas comunes, anteponiendo los intereses colectivos a los personales
	G06	Capacidad para analizar y resolver los problemas o imprevistos complejos que puedan surgir durante la actividad profesional dentro de cualquier tipo de organización socio-económica
	G07	Capacidad para trabajar con flexibilidad y versatilidad para adaptarse a las necesidades y exigencias de su entorno profesional
	G09	Capacidad para tomar decisiones de manera imparcial y desde un punto de vista racional
	G11	Habilidad para desenvolverse en un entorno multicultural e internacional, relacionándose con personas de distintas nacionalidades, idiomas y culturas
<b>Specific programme competences</b>	E05	Capacidad para evaluar los aspectos económicos y empresariales de las actividades de la ingeniería
	E06	Capacidad para aplicar procesos de aseguramiento de la calidad en procesos y productos
	E12	Capacidad para gestionar la complejidad a través de la abstracción, el modelado, las "best practices", los patrones, los estándares y el uso de herramientas apropiadas
	E21	Capacidad para realizar pruebas que verifiquen la validez del proyecto (funcional, integridad

		de los datos, rendimiento de las aplicaciones informáticas, equipos, comunicaciones, etc.)
	E22	Capacidad para desarrollar labores de implementación en las que se requiera un alto grado de conocimientos técnicos en diversos ámbitos (programación, configuración de equipos hardware y de comunicaciones, etc.).
	E26	Capacidad para definir y gestionar políticas de calidad para sistemas informáticos y de comunicaciones aplicando principios cuantitativos basados en métricas y estadísticas
<b>Learning outcomes</b>	R01	Trabajar de forma metódica
	R02	Automatizar tareas repetitivas
	R03	Anticiparse a los problemas
	R04	Analizar riesgos
	R05	Aplicar procesos de calidad rigurosamente
	R06	Diseñar soluciones a problemas
	R07	Dominar metodologías para la construcción organizada de software
	R08	Interaccionar en inglés en un escenario de trabajo
	R09	Trabajar productivamente en equipo
	R10	Aplicar técnicas matemáticas a la ingeniería

### PRE-REQUISITES:

Asignatura “Ingeniería del Software”.

### SUBJECT PROGRAMME:

#### Subject contents:

<b>1 - Principios de la Calidad del Software</b>
1.1 - Introducción: ¿Qué es la calidad software?
1.2 - Actividades para garantizar la calidad del software
1.3 - Ingeniería de la Calidad
<b>2 - Métricas para medir la calidad del software</b>
2.1 - Métricas del producto
2.2 - Métricas del Proceso
2.3 - Métricas en Programación Orientada a Objetos
<b>3 - 3.Gestión de la Configuración y del Software</b>
3.1 - Principios Básicos
3.2 - Patrones
3.3 - Control de Versiones en Práctica
3.4 - Integración Continua
<b>4 - Pruebas</b>
4.1 - Conceptos Básicos
4.2 - Actividades esenciales, gestión de los tests y automatización
4.3 - Cobertura
4.4 - Partición del dominio de parámetros de entrada y pruebas de límites.
4.5 - Control de flujo, dependencias e interacciones.
4.6 - Patrones
4.7 - Frameworks
4.8 - Desarrollo Orientado a Tests
<b>5 - Más allá de las pruebas</b>
5.1 - Prevención de defectos y mejora de procesos
5.2 - Inspecciones de Código y Verificación Formal
5.3 - Assertion Driven Development y Diseño por Contrato
5.4 - Tolerancia a Errores y Contención de Fallos

Subject planning could be modified due unforeseen circumstances (group performance, availability of resources, changes to academic calendar etc.) and should not, therefore, be considered to be definitive.

### TEACHING AND LEARNING METHODOLOGIES AND ACTIVITIES:

### Teaching and learning methodologies and activities applied:

El profesor seguirá las siguientes metodologías de enseñanza:

- Antes de cada sesión presencial el profesor indicará el material a preparar y las tareas a realizar
- Los alumnos de manera autónoma deberán trabajar esa materia y realizar los ejercicios correspondientes (si los hubiere).
- Sesiones Presenciales: El profesor utilizará esas sesiones para resolver dudas sobre la materia que ya ha debido ser estudiado , trabajos a entregar así como introducir el material para las siguientes semanas.
- Sesiones de tutoría: La adquisición de competencias se facilita mediante la observación del trabajo de expertos en la materia y la discusión interactiva sobre los asuntos prácticos. De este modo se dedicarán sesiones en las que los alumnos observarán al profesor durante el proceso de resolución de un problema para que luego ellos puedan enfrentarse a problemas similares de modo autónomo.
- Estudio de casos.
- Aprendizaje basado en casos y problemas: Análisis y evaluación de casos reales relacionados con la calidad del software.

Para optimizar la adopción de las competencias fijadas para esta asignatura se recomienda la siguiente metodología de aprendizaje:

- Seguimiento y estudio al día de la materia, según el calendario previsto por el profesor.
- Resolución de posibles dudas sobre la materia de cada semana durante las tutorías de esa semana o de la siguiente.
- Participación activa en las sesiones presenciales para poder así resolver aquellas dudas más importantes.
- Realización de los ejercicios y cuestiones prácticas propuestas por el profesor semanalmente. Que ciertas tareas no sean obligatorias no significa que no sean importantes. El trabajo en problemas prácticos ayuda a asimilar los conceptos teóricos y apreciar su utilidad y finalidad.

### Student work load:

Teaching mode	Teaching methods	Estimated hours
<b>Classroom activities</b>	Master classes	5
	Practical exercises	2
	Practical work, exercises, problem-solving etc.	1,5
	Assessment activities	3
<b>Individual study</b>	Tutorials	0,5
	Individual study	25
	Individual coursework preparation	18
	Group cousework preparation	9
	Research work	7
	Recommended reading	3
	Other individual study activities	1
<b>Total hours:</b>		<b>75</b>

### ASSESSMENT SCHEME:

#### Calculation of final mark:

Written tests:	40 %
Individual coursework:	30 %
Group coursework:	10 %
Final exam:	15 %
Participación y Trabajos Voluntarios:	5 %

**TOTAL** 100 %

\*Las observaciones específicas sobre el sistema de evaluación serán comunicadas por escrito a los alumnos al inicio de la materia.

## **BIBLIOGRAPHY AND DOCUMENTATION:**

### **Basic bibliography:**

Patton, Ron "Software Testing", 2nd ed., Sams Publishing, 2005
Horch, John W. "Practical Guide to Software Quality Management", Artech House, 2003

### **Recommended bibliography:**

Kan, Stephen. "Metrics and Models in Software Quality Engineering", Addison Wesley, 2002
Evans, Isabel "Achieving Software Quality through Teamwork", Artech House, 2004
Ravindranath, Pandian. "Software Metrics. A Guide to Planning, Analysis and Application", Auerbach Publications, 2004
Stamelos, Ioannis G. "Agile Software Development Quality Assurance", IdeaGroup Inc, 2007
Burnstein, Ilene. "Practical Software Testing", Springer-Verlag, 2003
Lewis, William. "Software Testing and Continuous Quality Improvement" 2nd ed., Auerbach Publications, 2005

### **Recommended websites:**

The most popular Software Testing and Quality Assurance discussions site. With over 50 forums that cover almost every area in software testing, quality assurance and quality engineering.	<a href="http://www.qaforums.com/">http://www.qaforums.com/</a>
Grupo de calidad del software, ingeniería y gestión en español.	<a href="http://www.calidaddelsoftware.com/">http://www.calidaddelsoftware.com/</a>
El sistema de calidad del centro de vuelos espacial de la NASA.	<a href="http://sw-assurance.gsfc.nasa.gov/disciplines/quality/">http://sw-assurance.gsfc.nasa.gov/disciplines/quality/</a>
Enlaces a gran número de instituciones y organizaciones relacionados con la Calidad del Software.	<a href="http://www.soft.com/Institute/HotList/">http://www.soft.com/Institute/HotList/</a>
Project Management Institute (PMI). PMI is the world's leading not-for-profit association for the project management profession.	<a href="http://www.pmi.org">http://www.pmi.org</a>
Algunos blogs relacionados con la calidad del software	<a href="http://nobugleftbehind.com/">http://nobugleftbehind.com/</a> <a href="http://jmbeas.blogspot.com/">http://jmbeas.blogspot.com/</a> <a href="http://adam.goucher.ca">http://adam.goucher.ca</a>
Comunidad CMS:	<a href="http://www.cmcrossroads.com/">http://www.cmcrossroads.com/</a>

\* Guía Docente sujeta a modificaciones