

30259 - Metodologías ágiles y calidad

Información del Plan Docente

Año académico: 2020/21

Asignatura: 30259 - Metodologías ágiles y calidad

Centro académico: 110 - Escuela de Ingeniería y Arquitectura

Titulación: 439 - Graduado en Ingeniería Informática

Créditos: 6.0

Curso: 4

Periodo de impartición: Segundo semestre

Clase de asignatura: ---

Materia: ---

1. Información Básica

1.1. Objetivos de la asignatura

La asignatura y sus resultados previstos responden a los siguientes planteamientos y objetivos:

En esta asignatura el alumno aprenderá a enfrentarse al desarrollo de proyectos software aplicando metodologías ágiles y técnicas dirigidas por modelos para la evaluación de la calidad del software, siendo la calidad un principio básico a garantizar.

1.2. Contexto y sentido de la asignatura en la titulación

Esta asignatura se oferta como optativa en la Mención de Ingeniería del Software del Grado de Ingeniería en Informática.

1.3. Recomendaciones para cursar la asignatura

Es recomendable que el alumno haya previamente adquirido formación básica en principios y técnicas de ingeniería del software y gestión de proyectos.

2. Competencias y resultados de aprendizaje

2.1. Competencias

Al superar la asignatura, el estudiante será más competente para...

Planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social.

Conocer y aplicar los principios, metodologías y ciclos de vida de la ingeniería de software.

2.2. Resultados de aprendizaje

El estudiante, para superar esta asignatura, deberá demostrar los siguientes resultados...

Conoce los conceptos vinculados a la calidad de los procesos del software y métricas sobre los mismos.

Conoce un mapa de los estándares y metodologías más actuales en el campo de la calidad del software.

Conoce el paradigma de ingeniería del software dirigida por el modelado (*Model-driven software engineering*) y sabe utilizarlo para la evaluación del rendimiento y fiabilidad del software.

Conoce técnicas de revisión de código para la eliminación de vulnerabilidades, en el contexto del desarrollo del software orientado a la seguridad.

Conoce las bases de las metodologías ágiles y un mapa de los estándares y aproximaciones más actuales en este campo. Sabe cómo desarrollar un proyecto software sobre alguna de las mismas.

2.3. Importancia de los resultados de aprendizaje

En la época en que nos encontramos, todo ingeniero informático debe tener los conocimientos básicos y habilidades necesarias para enfrentarse a la gestión de un proyecto software. También debe tener formación en un aspecto clave del

resultado obtenido como es la calidad del software desarrollado, así como conocimiento de los métodos y técnicas para alcanzarlo.

3.Evaluación

3.1.Tipo de pruebas y su valor sobre la nota final y criterios de evaluación para cada prueba

El estudiante deberá demostrar que ha alcanzado los resultados de aprendizaje previstos mediante las siguientes actividades de evaluación continua:

Pruebas prácticas (40%). El objetivo de las pruebas es evaluar los conocimientos y destrezas que han adquirido los alumnos en las sesiones prácticas.

Realización y/o defensa de proyectos en grupo (40%). Durante esta actividad se les planteará a los alumnos la gestión y desarrollo de un proyecto software en equipo, donde ponga en práctica los conocimientos que vaya adquiriendo durante la asignatura.

Realización de entregables periódicos (20%). Esta actividad incluye la realización de entregables relacionados con la temática de la asignatura, donde los alumnos deberán aplicar los conocimientos adquiridos.

Para los estudiantes que no opten por evaluación continua, está prevista una prueba global de evaluación en la que se plantearán cuestiones y/o problemas relacionados con el programa impartido en la asignatura. Su tipología y complejidad será similar a la presentada en las sesiones magistrales y prácticas. En general, se valorará la calidad y claridad de las respuestas, así como las estrategias de solución planteadas por los alumnos.

4.Metodología, actividades de aprendizaje, programa y recursos

4.1.Presentación metodológica general

El proceso de aprendizaje que se ha diseñado para esta asignatura se basa en lo siguiente:

El aprendizaje se obtendrá a partir de tres entradas: la sesiones explicativas del profesorado, los trabajos desarrollados en las sesiones prácticas y el trabajo del alumno (individual o en grupo).

4.2.Actividades de aprendizaje

El programa que se ofrece al estudiante para ayudarle a lograr los resultados previstos comprende las siguientes actividades...

Las actividades se organizarán en base a las exposiciones teóricas, la resolución de problemas (con y sin tutela del profesor), las prácticas, el trabajo en equipo y las actividades de evaluación.

4.3.Programa

El programa de la asignatura consta de dos partes:

Metodologías ágiles

- Fundamentos, valores y principios del *Agile*
- Diseño de proyectos ágiles
- Dirección de proyectos ágiles: *Scrum*, *XP* y *Kanban*
- Transiciones al *Agile*

Calidad del software

- Fundamentos de la calidad del software
- Métricas de la calidad del software
- Introducción al paradigma de ingeniería del software dirigido por modelos
- Evaluación del rendimiento y fiabilidad del software
- Revisión de código en el desarrollo del software orientado a la seguridad

4.4.Planificación de las actividades de aprendizaje y calendario de fechas clave

Calendario de sesiones presenciales y presentación de trabajos

El calendario de la asignatura estará definido por el centro en el calendario académico del curso correspondiente.

Trabajo del estudiante

La dedicación del estudiante para alcanzar los resultados de aprendizaje en esta asignatura se estima en 150 horas distribuidas del siguiente modo:

- 60 horas, aproximadamente, de actividades interactivas (sesiones de teoría y de problemas y sesiones prácticas)
- 60 horas de trabajo en grupo
- 22 horas de trabajo y estudio individual efectivo (estudio de apuntes y textos, resolución de problemas, preparación de clases y prácticas, etc.)
- 8 horas dedicadas a distintas pruebas de evaluación

El calendario detallado de las diversas actividades a desarrollar se establecerá una vez que la Universidad haya aprobado el calendario académico del curso correspondiente. En cualquier caso, las fechas importantes serán anunciadas con la suficiente antelación.

4.5. Bibliografía y recursos recomendados

[BB: Bibliografía básica / BC: Bibliografía complementaria]

Web biblioteca: <http://psfunizar7.unizar.es/br13/egAsignaturas.php?codigo=30259&Identificador=15423>

Metodologías ágiles

- [BB] The Agile Samurai. How Agile Masters Deliver Great Software, Rasmusson J., The Pragmatic Programmers. 2010.
- [BB] Scrum and XP from the trenches, H. Kniberg, InfoQ.com, 2015.
- [BB] Kanban and Scrum, making the most of both, H. Kniberg and M. Skarin, InfoQ.com, 2010.
- [BB] Agile project management with Kanban, E. Brechner, Microsoft Press, 2015.
- [BC] Impact Mapping: Making a big impact with software products and projects, Adzic G., Provoking Thoughts. 2012.
- [BC] User Stories Applied: for Agile Software Development, Cohn M., Addison Wesley Signature Series. 2004.
- [BC] Project Retrospectives: a handbook for team reviews, Kerth N.L., Dorset House. 2001.
- [BC] Agile Software Development: The Cooperative Game, Cockburn, A., 2nd ed. Addison-Wesley. 2006

Calidad del software

- [BB] Metrics and Models in Software Quality Engineering. Kan, Stephen H., 2nd edition Addison-Wesley, 2002
- [BB] Model-driven software engineering in practice. Brambilla M., Cabot J., Wimmer M., Morgan & Claypool Publishers. 2012
- [BC] Model-based Software Performance Analysis. Cortellessa V., Di Marco A., Inverardi P., Springer Verlag. 2011.
- [BC] Model-driven Dependability Assessment of Software Systems. Bernardi S., Merseguer J., Petriu D.C., Springer Verlag. 2013.
- [BC] Secure programming with static analysis. Chess B., West J., Software Security Series. Addison-Wesley. 2007.