

Curso Académico: (2020 / 2021)

Fecha de revisión: 10-07-2020

Departamento asignado a la asignatura: Departamento de Informática

Coordinador/a: AMESCUA SECO, ANTONIO DE

Tipo: Obligatoria Créditos ECTS : 6.0

Curso : 3 Cuatrimestre : 2

REQUISITOS (ASIGNATURAS O MATERIAS CUYO CONOCIMIENTO SE PRESUPONE)

Principios de Desarrollo de Software
Ingeniería del Software

OBJETIVOS

El objetivo de este curso es que el estudiante adquiera los conocimientos básicos para la aplicación de los principios básicos y técnicas ágiles para la gestión eficiente de proyectos software. Para lograr este objetivo, el alumno debe adquirir una serie de conocimientos y capacidades.

Por lo que se refiere a los conocimientos al finalizar el curso, el estudiante será capaz de:

- 1) Conocer los principios básicos e ideario de las metodologías ágiles de desarrollo de software.
- 2) Conocer las principales metodologías ágiles de desarrollo software. (PO: a, c)
- 3) Conocer los procesos de negocio asociados a la construcción de soluciones tecnológicas innovadoras y disruptivas (PO: a, c)
 - a) Visión general de Lean Startup
 - b) Producto Mínimo Viable
 - c) Medición
 - d) Aprendizaje
 - e) Business Model Canvas
- 4) Conocer los aspectos básicos para una gestión eficiente de los requisitos en un proyecto ágil. (PO: a, c)
 - a) Técnicas de identificación y conceptualización de ideas: Product Vision Box, Story Mapping y Story Telling
 - b) Técnicas para la gestión de una lista de producto: Estimación, Priorización, Deuda Técnica
- 5) Conocer los aspectos básicos de la gestión de proyectos de desarrollo de software basados en ciclos de vida iterativos e incrementales con etapas de duración prefijada. (PO: a, c)
 - a) Planificación de portfolio, productos, versiones e iteraciones.
 - b) Seguimiento cuantitativo de proyectos ágiles.
 - c) Seguimiento del desarrollo de iteraciones.
 - d) Revisiones de fin de iteración y retrospectivas
- 6) Entender la diferencia entre metodologías de desarrollo de software ágiles y las tradicionales y siendo capaz de determinar las condiciones idóneas para su aplicación. (CECRI16) (PO: a, c)

En cuanto a las capacidades, estas las podemos clasificar en dos grupos uno de capacidades específicas y otro de capacidades más genéricas o destrezas.

En cuanto a las capacidades específicas, al finalizar el curso el alumno será capaz de (CG1):

- 1) Extraer y gestionar los requisitos de una aplicación de software mediante la aplicación de la técnica de las historias de usuario (CESI2). (PO: a, c, i, k)
- 2) Organizar las distintas etapas de desarrollo de software (iteraciones) mediante la aplicación de la técnica de planificación adaptativa (CESI5). (PO: a, c, i, k)
- 3) Diseñar la arquitectura de un sistema software aplicando los principios de diseño simple y la técnica de refactoring. (CESI3) (PO: a, c, i, k)
- 4) Desarrollar las funcionalidades requeridas por una aplicación software mediante el diseño y la programación en parejas. (CESI3) (PO: a, c, i, k)
- 5) Aplicar los principios del desarrollo dirigido por las pruebas. (PO: a, c, i, k)
- 6) Aplicar formalismos, técnicas y herramientas para la integración, compilación y construcción automatizada de unidades de software. (PO: a, c, i, k)
- 7) Utilizar prácticas y técnicas de gestión de configuración para la puesta en marcha de los principios de propiedad colectiva de código. (PO: a, c, i, k)

8) Conocer las técnicas de seguimiento de proyectos incluyendo: reuniones diarias, gráficos de seguimiento, reuniones de fin de iteración y retrospectivas. (PO: a, c, i, k)

En cuanto a las capacidades generales o destrezas, durante el curso se trabajarán:

- 1) Capacidad de organización y planificación de un trabajo de desarrollo en equipo. (PO: d)
- 2) Capacidad de seguir un plan de trabajo establecido para obtener un producto de calidad. (PO: e)
- 3) Capacidad para aplicar los conocimientos adquiridos con respecto al desarrollo de un proyecto.
- 4) Capacidad de resolución de problemas y formulación de hipótesis de una manera conjunta y participativa (PO: d, e)
- 5) Capacidad para informar del avance de un proyecto (PO: g)

DESCRIPCIÓN DE CONTENIDOS: PROGRAMA

- 1) Principios generales del desarrollo ágil
 - 1.1.- El manifiesto Ágil
 - 1.2.- Principios de desarrollo ágil
 - 1.3.- Visión general del ciclo de vida de un proyecto de software ágil
- 2) Técnicas ágiles de Gestión de requisitos
 - 2.1.- Historias de Usuario: Caja de Visión de Producto, Story Telling, Story Mapping
 - 2.2.- La Lista de Producto
- 3) Lean Startup
 - 3.1.- Visión general de Lean Startup
 - 3.2.- Producto Mínimo Viable
 - 3.3.- Medición
 - 3.4.- Aprendizaje
 - 3.5.- Business Model Canvas
- 4) Técnicas ágiles de Planificación de Proyectos
 - 4.1.- Niveles de planificación ágil
 - 4.2.- Planificación de un producto
 - 4.3.- Planificación de una release
 - 4.4.- Planificación de una iteración
 - 4.5.- Definición de pruebas de sistema como mecanismo de gestión de proyectos
- 5) Técnicas y herramientas para la gestión ágil de proyectos software
 - 5.1.- Gestión del trabajo basado en tableros Kanban
 - 5.2.- Seguimiento cuantitativo de proyectos ágiles.
- 6) Retrospectivas y Showcases
 - 6.1.- Seguimiento del desarrollo de iteraciones.
 - 6.2.- Revisiones de fin de iteración y retrospectivas

ACTIVIDADES FORMATIVAS, METODOLOGÍA A UTILIZAR Y RÉGIMEN DE TUTORÍAS

Las actividades que se llevan a cabo en la impartición de la asignatura son:

- 1) Clases Teóricas (1,5 ECTS): Estas clases tienen por objetivo alcanzar las competencias específicas asociadas al conocimiento que se han presentado para esta asignatura, dentro de un marco teórico. (PO: a, c, i, k) En el ámbito de las clases teóricas, los conceptos de cada semana se presentarán mediante minivideos que los alumnos tendrán que visualizar con anterioridad a las clases. Una vez visualizados deberán responder un cuestionario tipo test que verificará el conocimiento de los conceptos presentados en los minivideos. Durante las clases presenciales de teoría se discutirán los resultados del test. Posteriormente se presentarán vídeos, casos de estudio o lecturas que permitirán afianzar el correcto entendimiento de los conceptos presentados en los minivideos proporcionando un segundo nivel de aprendizaje. Por último, se planteará el reto que se debe conseguir la semana proporcionando una pauta inicial para su elaboración y dejando que los estudiantes puedan comenzar su realización en equipos.
- 2) Clases Prácticas (1,5 ECTS): En estas clases los alumnos aprenderán las nociones básicas de las herramientas, técnicas y guías de aplicación de los principios y técnicas del desarrollo ágil de software. (PO: a, c, i, k) En el ámbito de las clases de prácticas, cada uno de los equipos presentará el grado de avance de su resolución del reto y los pasos planteados para su finalización de tal manera que se permita el intercambio de ideas entre los distintos equipos de prácticas. Por último, el profesor proporcionará feedback necesario para que los alumnos completen correctamente el reto de la semana.
- 3) Realización de Actividades Académicas Dirigidas. Resolución de los retos (3 ECTS):
 - 3.a) Los alumnos realizarán trabajos semanales (retos) de manera individual o en parejas que permitirán la puesta en práctica de las técnicas y herramientas aprendidas mediante la aplicación de las instrucciones que el profesor establecerá en un guión de prácticas que se comentará durante las clases de prácticas. (PO: c, e)
 - 3.b) Desarrollo de un proyecto final: El objetivo de este trabajo pongan en práctica los conocimientos impartidos en las clases teóricas y prácticas. Para ello se procederá a la mejora y ampliación de una aplicación software mediante la realización de las actividades técnicas en equipo mediante la aplicación

de las técnicas de desarrollo ágil. (PO: d, e, g)

SISTEMA DE EVALUACIÓN

La evaluación de la asignatura estará completamente basada en el modelo de evaluación continua. El total de la nota del alumnos se derivará de su empeño en las diferentes actividades propuestas en cada sesión. El total de la nota se calculará de la siguiente manera:

- 1) Evaluación de teoría - Tests y actividades en clase (PO: a, c, e, k)
 - Supondrá el 30% de la nota
- 2) Ejercicios Guiados (PO: a, c, e, k)
 - Supondrá el 30% de la nota
- 3) Proyecto Final (PO: a, c, d, e, g, i, k)
 - Supondrá el 40% de la nota

Peso porcentual del Examen Final: 40

Peso porcentual del resto de la evaluación: 60

BIBLIOGRAFÍA BÁSICA

- Jeff Patton User Story Mapping, O'Reilly Media, 2014
- Kniberg, Henrik. Lean from the Trenches. Managing Large-Scale Projects with Kanban. , The Pragmatic Bookself, 2011
- Richard Levy Agile Foundations: Principles, practices and frameworks, BCS, 2015
- Rubin, Kenneth S., Essential Scrum: a practical guide to the most popular agile process, Pearson Education, Inc., 2013
- Venkat Subramaniam and Andy Hunt Practices of an Agile Developer, The Pragmatic Bookshelf. Pragmatic Programmers, 2013

BIBLIOGRAFÍA COMPLEMENTARIA

- Cockburn, Alistair Agile Software Development, Addison-Wesley, 2001.
- Highsmith, Jim Agile Software Development Ecosystems, Addison-Wesley, 2002.
- Schwaber, Ken et al. Agile Software Development With SCRUM, Prentice Hall, 2001

RECURSOS ELECTRÓNICOS BÁSICOS

- Agile Alliance . Agile Alliance: <http://www.agilealliance.org>