

Formulario de aprobación de curso de posgrado/educación permanente

Asignatura: Técnicas de calidad de software para proyectos ágiles y tradicionales

(Si el nombre contiene siglas deberán ser aclaradas)

Modalidad: (posgrado, educación permanente o ambas)	Posgrado	<input checked="" type="checkbox"/>
	Educación permanente	<input checked="" type="checkbox"/>

Profesor de la asignatura: Dr. Ing. Diego Vallespir, Profesor Agregado, Instituto de Computación

Otros docentes de la Facultad: Msc. Carolina Valverde, Asistente, Instituto de Computación
Msc. Silvana Moreno, Asistente, Instituto de Computación
Msc. Cecilia Apa, Asistente, Instituto de Computación

Docentes fuera de Facultad: Msc. Darío Macchi, Scrum Master, VIX

Programa(s) de posgrado: Especialización en Ingeniería de Software – Maestría en Ingeniería de Software

Instituto o unidad: Instituto de computación

Departamento o área: Ingeniería de software

Horas Presenciales: 45

Nº de Créditos: 6

(de acuerdo a la definición de la UdelaR, un crédito equivale a 15 horas de dedicación del estudiante según se detalla en el ítem Metodología de enseñanza)

Público objetivo: Líderes de equipos de desarrollo y/o mantenimiento de software. Líderes de equipos de aseguramiento de la calidad. Integrantes de equipos de aseguramiento de la calidad, de desarrollo y/o mantenimiento de software. Gerentes de proyectos de software. Estudiantes de posgrado interesados en las técnicas de calidad de software.

Cupos: No tiene cupo.

Objetivos: En este curso se presentan temas de calidad de software y técnicas asociadas que se utilizan tanto en proyectos ágiles como tradicionales. Se tratan los siguientes temas: conceptos de calidad de software, técnicas de calidad de software, Inspección de software, mejora de procesos de software como una forma de abordar la mejora continua de la calidad, implantación (y adaptación) de procesos de desarrollo de software para lograr cambios organizacionales, pull request, TDD, programación de a pares y revisiones individuales. Durante el curso se realizan ejercicios prácticos y se desarrolla un laboratorio para que el profesional que asista al mismo pueda aplicar o proponer la aplicación de estas técnicas en su ambiente laboral.

Conocimientos previos exigidos: Conocimientos básicos de ingeniería de software

Conocimientos previos recomendados:

Metodología de enseñanza:

Descripción de la metodología: La metodología es teórico práctico con talleres. El curso tendrá clases expositivas y ejercicios que se realizarán en clase y fuera de la misma. Los estudiantes, de forma grupal, deberán realizar un trabajo final vinculado a algún tema del curso.

Detalle de horas:

- Horas de clase (teórico): 25
- Horas de clase (práctico): 5
- Horas de clase (laboratorio): 6
- Horas de consulta: 6
- Horas de evaluación: 3
 - Subtotal de horas presenciales: 45
- Horas de estudio: 20
- Horas de resolución de ejercicios/prácticos: 10
- Horas proyecto final/monografía: 15
 - Total de horas de dedicación del estudiante: 90

Forma de evaluación:

Pruebas individuales y elaboración de un trabajo final grupal.

Temario:

1. Calidad de software
 - 1.1. Fundamentos de la calidad de software
 - 1.2. Procesos de gestión de la calidad de software
 - 1.3. Consideraciones prácticas
2. Técnicas de calidad de software
3. Técnicas de lectura de artefactos de software
4. Implantación de procesos de desarrollo de software y entrenamiento
 - 4.1. Implantación de procesos de desarrollo de software
 - 4.2. Entrenamiento
 - 4.3. Dificultades comunes y cómo evitarlas
5. Calidad de software en proyectos ágiles
6. El proceso de Inspección de software
 - 6.1 Inicio
 - 6.2 Documentación
 - 6.3 Revisión

6.4 Conclusión

7. Mejora de procesos (basado en medidas de calidad de la Inspección)

Bibliografía:

(título del libro-nombre del autor-editorial-ISBN-fecha de edición)

Software Inspection – Tom Gilb, Dorothy Graham– Addison-Wesley - 978-0201631814 – Enero 1994

Gilb, T. Agile Specification Quality Control. Cutter It Journal 18(1), 35 -39, 2005.

Fagan M. A History of Software Inspections. In Broy M., Denert E. (eds) Software Pioneers. Springer, Berlin, Heidelberg, 2002

Capers Jones and Olivier Bonsignour, Do You Inspect?. Publicado en línea <http://www.drdoobs.com/architecture-and-design/do-you-inspect/231903203>, November 30, 2011

Biffi St. "Using Inspection Data for Defect Estimation. " IEEE Software 17(6); special issue on recent project estimation methods pp. 36-43 2000.

A. Aurum, H. Petersson and C. Wohlin, "State-of-the-Art: Software Inspections after 25 Years", Software Testing Verification and Reliability, Vol. 12, No. 3, pp. 133- 154, 2002.

Basili, V., Green, S., Laitenberger, O., Lanubile, F., Shull, F., Sorumgard, S., and Zelkowitz, M. The Empirical Investigation of Perspective-based Reading. Journal of Empirical Software Engineering, 2(1):133–164, 1996

Brykczynski, B; Meeson, R.; Wheeler, D; Software Inspection: Eliminating Software Defects. Proceedings of the Sixth Annual Software Technology Conference, 1994.

Otros artículos actuales

Datos del curso

Fecha de inicio y finalización: 11 de octubre de 2022 a 13 de noviembre de 2022

Horario y Salón: Lunes, martes y jueves de 18 a 21 hs. Salón a confirmar.

Arancel: \$ 25.200

[Si la modalidad no corresponde indique "no corresponde". Si el curso contempla otorgar becas, indíquelo]

Arancel para estudiantes inscriptos en la modalidad posgrado: \$ 25.200

Arancel para estudiantes inscriptos en la modalidad educación permanente: \$ 25.200
