

PROGRAMA DE ESTUDIOS: ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE

PROTOCOLO

Fechas	Mes/año
Elaboración	05-2010
Aprobación	
Aplicación	09-2010

Clave			Semestre	6°		
Nivel	Licenciatura	X	Maestría		Doctorado	
Ciclo	Integración		Básico		Superior	X
Colegio	H. y C.S.		C. y T.	X	C. y H.	

Plan de estudios del que forma parte: Ingeniería de Software

Propósito(s) general(es):
 El estudiante aprenderá a utilizar los estándares, métodos, técnicas y herramientas para la validación y verificación de sistemas de software desde sus fases iniciales hasta la liberación y entrega de los mismos.

Carácter		Modalidad				Horas de estudio semestral (16 semanas)					
Indispensable	X	Seminario		Taller		Con Docente	Teóricas	36	Autónomas	Teóricas	20
		Curso	X	Curso-taller			Prácticas	36		Prácticas	36
Optativa *		Laboratorio	X	Clínica		Carga horaria semanal: 4.5 + 3.5 = 8			Carga horaria semestral: 128		

Asignaturas Previas:	Asignaturas Posteriores:
Construcción y evolución del software, Análisis de requisitos	Técnicas de pruebas de software

Requerimientos para cursar la asignatura	Conocimientos y habilidades: Saber solucionar problemas mediante la ayuda de una computadora, por medio de programas
---	---

Perfil deseable del profesor: Ingeniero en computación o alguna carrera afín, con maestría o doctorado de preferencia, en Ciencias de la Computación, o bien, amplios conocimientos y experiencia laboral en el área. Además, amplia experiencia docente y de investigación.

Academia responsable del programa: Informática	Diseñador (es): Dr. Alfonso San Miguel Aguirre
--	--

* Aquellas en las que se ofrece la posibilidad de cursar una de las asignaturas, para cubrir un requisito INDISPENSABLE será considerada INDISPENSABLE.

INTRODUCCIÓN

El aseguramiento de la calidad del software (SQA, software quality assurance por sus siglas en inglés) es un medio de monitorear los procesos involucrados en la ingeniería de software así como los métodos usados para asegurar la calidad. Los métodos con los cuales esto puede obtenerse son muchos y variados, y puede ser que se apeguen a estándares, como el ISO9000 o bien a modelos, tales como CMMI.

Un software de calidad se puede decir que (prácticamente) no tiene “bugs”. Se entrega a tiempo y dentro del presupuesto, cumple con los requerimientos o las expectativas y es factible de darle mantenimiento.

En esta materia el estudiante tomará conciencia de la importancia de asegurar la calidad del software en un proyecto durante todo su desarrollo. Esto incluye los procesos de la definición de requerimientos, el diseño del software, la codificación, el control del código fuente, revisiones del código, manejo de versiones, de configuraciones, pruebas y, finalmente, la liberación e integración del producto. El estudiante aprenderá a evaluar y a crear software de calidad por medio del uso de estándares, procedimientos y procesos usados en la producción de software.

PROPÓSITOS GENERALES

El estudiante distinguirá y aplicará los estándares, métodos, técnicas y herramientas para la validación y verificación de sistemas de software desde sus fases iniciales hasta la liberación y entrega de los mismos.

CONTENIDOS

TEMAS Y SUBTEMAS	PROPÓSITOS ESPECIFICOS
1 Calidad del software 1.1 Motivación de la importancia de SQA 1.2 Problemas comunes en el proceso de desarrollo de SW 1.3 Herramientas para confeccionar y evaluar pruebas. 1.4 Confección y evaluación de las pruebas 1.5 Herramientas para confeccionar y evaluar pruebas	El estudiante tomará conciencia de la importancia de poder asegurar la calidad del software que se desarrolle. Tomará en cuenta que el primer punto a realizar para poder asegurar SQA es tener una definición clara de los requerimientos. Requerimientos del cliente, del producto y de los componentes del producto. Sabrá los problemas más comunes que impiden lograr la calidad el software, es decir, requerimientos pobres, calendario irreal, pruebas inadecuadas, falta de comunicación, cambios en los requerimientos después de que el proyecto ha iniciado.
2 Inspecciones y revisiones 2.1 Fases y tipos de inspección 2.2 Iniciación a la inspección. Pre-requisitos 2.3 Inspecciones de requerimientos, de especificación, de documentación 2.4 Métricas de inspección	El estudiante será capaz de entender el proceso de desarrollo de software en su totalidad y como se relaciona en los planes de negocio y los fines de una organización. Entenderá la importancia de contar con una buena comunicación y el ser capaz de tener en cuenta los varios puntos de vista. Tendrá la habilidad de encontrar problemas en las inspecciones y revisiones.

<p>3 Pruebas</p>	<p>El estudiante sabrá que la parte de prueba de un sistema es esencial en el desarrollo de software de calidad. Conocerá las distintas fases de prueba y sabrá aplicarlas. Asimismo, conocerá distintas herramientas computacionales para la realización de pruebas.</p>
<p>3.1 Caja blanca y caja negra 3.2 Pruebas unitarias y pruebas de integración 3.3 Pruebas de aceptación 3.4 Pruebas de sistema, alfa, beta y aceptación 3.5 Herramientas para el diseño y el desarrollo de pruebas</p>	
<p>4 Aseguramiento de la calidad (modelos, estándares y procesos)</p>	<p>El estudiante identificará la diferencia entre modelos, estándares y procesos. Los estándares son los criterios contra los cuales se comparan los productos de software. Los modelos son los criterios contra los que se comparan los procesos de control y de desarrollo. Es decir, sabrá que los estándares definen lo que debe hacerse, mientras que los modelos definen como se hace el trabajo realmente, por quién, cuándo y qué se hace con los resultados.</p>
<p>4.1 Ciclo de vida de un proyecto 4.2 Estándares de documentación 4.3 Estándares de diseño 4.4 Estándares de código</p>	
<p>5 Métricas</p>	<p>El estudiante podrá identificar diferentes tipos de métrica usados para medir la calidad de un proyecto. Una métrica básica es tener en cuenta los defectos en el código. Se puede sea tratar de disminuir la densidad de los defectos, sea mantener esta densidad pero disminuyendo el tiempo de entrega de un producto.</p>
<p>5.1 Validación de métricas 5.2 Implantación de las métricas 5.2 Análisis de resultados 5.4 Validación 5.5 Ejemplos: IEEE, PSP, Microsoft, etc.</p>	

METODOLOGÍA PARA EL CURSO

Este curso es teórico práctico. La materia se impartirá por medio de clases teóricas, prácticas y asesorías. Se tendrá, al menos, un caso de estudio para que en él el estudiante pueda poner en práctica los conocimientos adquiridos en el curso durante el transcurso del semestre. El fin último, no solamente de este curso, sino de de ingeniería de software que se está proponiendo es que el futuro egresado de ella sepa responder de una manera rápida a las necesidades cambiantes de los así llamados clientes. Puesto que la aplicación en la práctica de los conocimientos adquiridos en el aula es indispensable, se tendrán sesiones en aula de computadoras una vez por semana, en ellas el estudiante pondrá en práctica los conceptos tratados en el curso realizando un software de calidad.

La Universidad Autónoma de la Ciudad de México dispone de equipos de cómputo, PCs, para que el estudiante pueda programar los algoritmos vistos en la materia. A medida que la tecnología va avanzando es necesario ir actualizando este equipo, tal como se ha hecho hasta ahora.

EVALUACIÓN DIAGNÓSTICA

Se aplicará un examen escrito sobre las estructuras de datos y programación orientada a objetos en algún lenguaje de programación, C, C++ C# o Java entre otros. Esto con el fin de detectar los conocimientos con los que cuentan los estudiantes y sugerir acciones para que adquieran los que no tengan, por medio de un repaso general dentro de la clase o en asesorías individuales.

EVALUACIÓN FORMATIVA

Con el propósito de dar seguimiento al proceso de enseñanza aprendizaje, se propone aplicar dos evaluaciones formativas, la primera se aplicará al término del tema (2). La segunda evaluación formativa se realizará al terminar el curso, donde el estudiante mostrará sus conocimientos acerca de cómo diseñar un sistema con calidad. Los instrumentos para las evaluaciones formativas constarán tanto de preguntas sobre la teoría de cómo lograr escribir código de calidad. Además de las evaluaciones formativas se dejarán tareas a los estudiantes al terminar cada tema. Se hará un análisis junto con los estudiantes para evaluar los temas que presenten problemas y realizar una revisión de los mismos. En el caso de que sean pocos los estudiantes que tengan problemas con alguno de los temas, se tratarán en asesoría.

Se pondrá especial atención en el cumplimiento de las prácticas en el laboratorio, así como en la realización de las tareas, la participación y asistencia a clases, ya que estas actividades ayudan a la formación del estudiante.

EVALUACIÓN DE CERTIFICACIÓN

El examen de certificación evaluará los conocimientos del estudiante sobre todos los temas tratados durante el curso. Este instrumento consta de dos partes: una evaluación teórica mediante examen escrito y la segunda mediante el desarrollo de un proyecto. En esta, al igual que en otras materias de final de la ingeniería de software, lo más importante es que el estudiante lleve a la práctica el material tratado en clase, de aquí que la certificación escrita tendrá al 30% de la calificación final, mientras que la parte práctica se evaluará por medio de un proyecto. En este proyecto debe mostrar que realmente ha alcanzado los propósitos del curso y se tratará de que el mismo proyecto se estudie y se realice en distintas materias relacionadas con ésta, tales como “Análisis y modelamiento de software”, “Diseño de software”, “Técnicas de prueba de software”, “Métricas de software”, entre otras.

BIBLIOGRAFÍA

La bibliografía que se menciona a continuación es la básica para el estudiante y el profesor:

Núm.	Bibliografía	Temas para los que se recomienda
1	Bruegge, Bernd, Dutoit, Allen H. Ingeniería de Software Orientada a objetos . Prentice Hall, 2002.	1 -5
2	Lewis, William E. Software Testing and Continuous Quality Improvement . 2000.	1-5
3	Myers, J. Glenford; El arte de probar ; El Ateneo editorial; 1984.	1
4	Dennis de Champeaux; Object oriented Development Process and Metrics ; Prentice Hall. 1997.	4, 5
5	Kan, Stephen H. Metrics and Models in Software Quality Engineering (2nd Edition). 2002.	2, 3
6	Buerstein, Llene. Practical Software Testing . Springer Verlag. 2003.	1-4

OTROS RECURSOS

Pintarrón, computadora laptop para las presentaciones y proyector de video (cañón).