



UNIVERSIDAD AUTÓNOMA DE YUCATÁN
FACULTAD DE MATEMÁTICAS



MISIÓN

Formar profesionales altamente capacitados, desarrollar investigación y realizar actividades de extensión en matemáticas y computación, así como en sus diversas aplicaciones.

ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE

Sexto semestre

LICENCIATURA EN INGENIERÍA DE SOFTWARE

Enero – Junio, 2007

LICENCIATURA EN INGENIERÍA DE SOFTWARE

1. Explique y aplique un proceso de desarrollo de software sistémico acorde a la magnitud y complejidad de proyectos de aplicación, sean o no éstos nuevos desarrollos, tomando en cuenta la evolución y el cambio de los mismos.
2. Describa y aplique las herramientas necesarias para la especificación, diseño, verificación y validación de sistemas de software.
3. Se desempeñe en al menos un área de concentración, trabaje y se comunique de forma profesional en equipos interdisciplinarios.
4. Aplique el conocimiento y las habilidades para mejorar el proceso de desarrollo de software.
5. Contribuya al avance de la Ingeniería de Software con un acervo de conocimientos tanto teóricos como prácticos.

I Generales.

ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE

Clave del curso: IS-07

Semestre: Sexto

Licenciatura: Ingeniería de Software

Clases por semana: 3 **Número de clases:** 48 **Duración de la clase:** 80 minutos

Horas teóricas: 50 **Horas prácticas:** 22 **Créditos:** 9

Antecedentes académicos: Ninguno

Cuerpo(s) académico responsable(s): Diseño de Aplicaciones-Ingeniería de Software

Elaboradores: Carlos Benito Mojica Ruiz, Antonio Aguilera Güemez y Carlos Aguilar Buenfil.

Fecha de elaboración: Noviembre de 2006

Versión 0.1.0

Plan de estudio aplicable: 2004

II Descripción y justificación del curso.

En este curso se plantea el aseguramiento de la calidad del software como una actividad que da inicio desde el arranque mismo de un proyecto de software, desde la especificación misma de los requerimientos. Se enfatiza en la fase de Pruebas, haciéndole ver al estudiante que a pesar que en los modelos de ciclo de vida del software esta fase aparentemente comienza después de la codificación, siendo que, como se mencionó, da inicio con el proyecto mismo. Por ello, durante la planeación del proyecto de software se plantea la importancia de la construcción del plan de aseguramiento de la calidad y se comienza a detallar el plan de pruebas técnicas, el cual cubrirá todos los productos que se consideren susceptibles de ser revisados o probados.

Se describen las técnicas de aseguramiento de la calidad antes de la compilación del código y las posteriores a ésta, haciendo hincapié sobre las diferencias administrativas y técnicas en ambos casos.

III Objetivos.

General

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.

Específicos

1. Conducir revisiones e inspecciones eficaces y eficientes.
2. Diseñar e implementar planes adecuados de prueba.
3. Aplicar diversas técnicas de prueba de manera eficiente y efectiva.
4. Aplicar técnicas estadísticas para evaluar y predecir la densidad de defectos.

IV Resultados esperados.

Al final del curso el estudiante:

- a) Entenderá la razón y el propósito del aseguramiento de la calidad del software.
- b) Comprenderá los atributos del aseguramiento de la calidad, la verificación, validación y las pruebas del software.
- c) Diseñará y aplicará planes adecuados de prueba.
- d) Entenderá y aplicará diversas técnicas y estrategias de aseguramiento de la calidad tanto estáticas como dinámicas aplicadas al software.

V Contenido.

- Unidad 1 Introducción al Aseguramiento de la Calidad del Software. (8 sesiones)
Al finalizar la unidad, el alumno describirá los conceptos básicos del aseguramiento de la calidad del software pasando por los modelos CMM y SPICE.
- i. ¿Qué es calidad?
 - ii. Los costos y el impacto de una mala calidad.
 - iii. Atributos de la calidad del software (usabilidad, confiabilidad, disponibilidad, etc.).
 - iv. El proceso de Aseguramiento de la Calidad según CMM Y SPICE.
 - v. Planeación de la Calidad.
 - vi. Aseguramiento del proceso de Aseguramiento de la Calidad del Software.
 - vii. Aseguramiento del producto.
- Unidad 2 Verificación y Validación del Software. (5 sesiones)
Al finalizar la unidad, el alumno describirá los conceptos de verificación y validación de software durante el ciclo de vida del software.
- i. Conceptos y fundamentos.
 - ii. Objetivos y restricciones.
 - iii. La V&V en diversos puntos y productos del ciclo de vida del software.
 - iv. Desarrollo conducido por pruebas.
- Unidad 3 Inspecciones. (5 sesiones)
Al finalizar la unidad, el alumno describirá los métodos para conducir inspecciones, pasando por revisiones personales y pruebas de escritorio.
- i. Revisiones personales.
 - ii. Prueba de escritorio.
 - iii. Caminatas.
 - iv. Inspecciones.
- Unidad 4 Plan de Aseguramiento de la Calidad. (10 sesiones)
Al finalizar la unidad, el alumno aplicará las guías para la elaboración de un plan de aseguramiento de la calidad.

- i. Participantes en la elaboración de un plan de aseguramiento de la calidad.
- ii. Partes que componen a un plan de aseguramiento de la calidad.
- iii. Revisión del plan de aseguramiento de la calidad.

Unidad 5 Pruebas del Software. (10 sesiones)

Al finalizar la unidad, el alumno aplicará diversas técnicas de prueba de software.

- i. Desarrollo de casos de pruebas basados en los requerimientos (casos de uso).
- ii. Pruebas unitarias. JUnit.
- iii. Manejo de excepciones (escritura de casos de prueba para disparar excepciones).
- iv. Pruebas basadas en análisis de cobertura y en estructura.
- v. Pruebas funcionales (caja negra).
- vi. Pruebas de integración.
- vii. Pruebas basadas en el perfil operacional.
- viii. Pruebas del Sistema, Alfa, Beta y de Aceptación.
- ix. Pruebas de Regresión.
- x. Herramientas de prueba.

Unidad 6 Administración del proceso de prueba y Liberación. (6 sesiones)

Al finalizar la unidad, el alumno describirá el proceso de liberación de un producto de software.

- i. Administración del proceso de pruebas.
- ii. Seguimiento y análisis del reporte de problemas.
- iii. El proceso de liberación.
- iv. Plan de capacitación.

VI Criterios de evaluación.

Método	Porcentaje	Descripción
Exámenes parciales	90%	Se aplicarán 3 exámenes escritos (30% c/u).
Tareas	10%	Diversos reportes de investigación para su presentación oral y escrita personales o en equipo. Algunas tareas dentro de la parte práctica.
La suma de estos porcentajes representan el 70% de la calificación final		
Examen Ordinario	30%	El examen ordinario puede ser exentado habiendo obtenido una calificación del 85% entre los exámenes parciales y tareas.

VII Bibliografía.

No hay un libro de texto, el material que se va a cubrir se encuentra disperso en la siguiente bibliografía señalada, en muchos casos, los temas están repetidos.

Libros.

1. Software Testing and Continuous Quality Improvement 2nd Edition; William E. Lewis - Auerbach Publications. ISBN 0-8493-2524-2.
2. Software Quality Assurance: Principles And Practice; Nina Godbole; August 1, 2004; Alpha Science International; ISBN 1842651765.
3. Software Testing and Continuous Quality Improvement; Lewis, William E; 2nd edition (June 1, 2004); AUERBACH; ISBN 0849325242.
4. High Quality Low Cost Software Inspections; Ronald A. Radice; April 28, 2004; Paradoxicon Publishing; ISBN 0964591316.
5. Software Quality Assurance: From Theory to Implementation; Galin, Daniel; September 11, 2003; Addison Wesley; ISBN 0201709457.
6. Practical Software Testing; Buerstein, Llene; 1 edition (June 24, 2003); Springer Verlag; ISBN 0387951318.
7. Metrics and Models in Software Quality; Kan, Stephen; (2nd Edition) 2002; Addison Wesley; ISBN 0201633396.
8. Testing Computer Software; Cem Kaner, Jack Falk, Hung Q. Nguyen; 2 edition (April 12, 1999); Wiley; ISBN 0471358460.
9. Systematic Software Testing; Rick D. Craig, Stefan P. Jaskiel; May 2002; Artech House Publishers; ISBN 1580535089.
10. Unit Testing in Java: How Tests Drive the Code; Johannes Link; April 2003; Morgan Kaufmann; ISBN 1558608680.

X Recursos computacionales necesarios.

Indispensable:

- Editor de texto.
- Conexión a Internet.
- Computadora conectada a Internet con sistema operativo y un navegador de Internet instalado y configurado.
- Impresora.
- Herramienta para realizar presentaciones.

Recomendable:

- Herramientas CASE para la realización de pruebas.

VIII Perfil académico del docente.

Ingeniero de Software, Licenciado en Ciencias Computacionales, Ingeniero en Sistemas Computacionales o afín, preferentemente con postgrado y experiencia docente, de investigación o de trabajo en el área.

IX Reglamento

1. Para aprobar el curso se deberá contar con al menos el 80% de asistencia.
2. Después de 15 minutos de haber iniciado la clase no tendrá asistencia la persona que entre, es decir, no se prohíbe la entrada tardía.
3. Si entran a clase es para poner atención y participar activamente.
4. Los trabajos deben cumplir con todos los requisitos especificados, de lo contrario no serán calificados. Es responsabilidad del alumno obtener los criterios de evaluación de cada una de las tareas o proyectos.
5. Los trabajos deberán entregarse durante los primeros 20 minutos de haber comenzado la clase en la cual se recibirán.
6. Cada día de atraso en la entrega de los trabajos significa 10 puntos menos.
7. No se prohíbe la cooperación en la resolución de las tareas o proyectos, sin embargo, los de carácter individual deben ser realizados en la misma forma, individual, de lo contrario no serán calificados.
8. Se podrán obtener puntos extras con la presentación de soluciones creativas.
9. En los trabajos por equipo, la calificación entre sus miembros puede variar dependiendo de la contribución que tengan en los trabajos.
10. Poner los celulares en modo silencio o apagarlos y utilizarlos fuera del salón de clase.
11. En cualquier día de clase, se evaluará el grado de comprensión de los temas vistos, a manera grupal o de una parte de la clase. Los puntos malos obtenidos serán restados de las evaluaciones parciales.
12. Este reglamento complementa los otros dictados por la Facultad y la Universidad.

Perfil Profesiográfico:

Ingeniero de Software, Licenciado en Ciencias de la Computación o afín, preferentemente con postgrado y experiencia docente, de investigación o de trabajo en el área.

Elaboración: M.S.C. Carlos Javier Aguilar Buenfil, M. en C. Antonio Aguilera Güemez, M. en C. Carlos Benito Mojica Ruiz.

Fecha de elaboración: Noviembre, 2006.