

NOMBRE DE LA INSTITUCIÓN	Universidad de Sonora
DIVISIÓN ACADÉMICA	División Ciencias Exactas y Naturales
DEPARTAMENTO QUE IMPARTE LA MATERIA	Departamento de Matemáticas
LICENCIATURAS USUARIAS	Ciencias de la Computación
NOMBRE DE LA MATERIA	Verificación y Validación de Software
CLAVE	9485
EJE FORMATIVO	Especializante
REQUISITOS	Análisis y Diseño Orientado a Objetos
CARÁCTER	Optativa
VALOR EN CRÉDITOS	8 (3 hrs. teoría/2 hrs. laboratorio)

Introducción

La sociedad moderna cada día depende más y más de sistemas informáticos eficientes y confiables. Las demandas por servicios de procesamiento de información que se adecuen a las necesidades de los usuarios de manera fiable, y que además se libere el software en corto tiempo y su desarrollo sea de bajo costo tienen un impacto profundo en las estrategias y procesos de desarrollo de software.

El desarrollo de sistemas o productos de software de alta calidad requiere de recursos humanos que tengan conocimientos profundos de los distintos paradigmas de construcción de software así como de las diversas técnicas que se pueden aplicar para garantizar la calidad de los distintos productos generados y de los procesos utilizados. Además, requiere conocer y desarrollar las habilidades de comunicación y coordinación que le permitan integrarse y colaborar eficazmente en un equipo de desarrolladores de software.

El proyecto de desarrollo de software, en términos generales, es un reto para cualquier grupo de profesionales de la tecnología de la información. En este curso, los estudiantes formarán equipos de desarrollo de software, en donde ellos trabajarán con un cliente real, del sector privado, gubernamental o educativo. Ellos tendrán que organizar las tareas que se deben abordar por el proyecto, tanto técnicas como administrativas para entregar un producto de software de alta calidad al final del semestre, incluyendo todos los artefactos generados según el proceso y ciclo vital de desarrollo de software seleccionado.

Objetivo General del Curso

El alumno será capaz de aplicar los conocimientos adquiridos a lo largo de su formación académica, en las ciencias computacionales y en particular el área de ingeniería de software, en un proyecto de desarrollo de software para un cliente real, en donde el alumno seleccione, y adapte, junto con sus compañeros, las técnicas y métodos del aseguramiento de calidad que serán aplicados en cada una de las fases del proyecto para entregar un producto que satisfaga los requisitos funcionales y de desempeño especificados por el cliente.

Objetivos Específicos del Curso

- Comprender la necesidad de verificar y validar el software.
- Analizar los criterios de calidad que se evalúan en el software y el proceso.
- Analizar y evaluar distintas técnicas de verificación y validación del software.
- Establecer un enfoque sistemático para la planeación de pruebas de software.
- Discutir aspectos de lógica; álgebra y complejidad computacional y sus aplicaciones a la verificación de programas computacionales.
- Establecer y aplicar los lineamientos de aseguramiento de la calidad a cada uno de los productos generados durante el proceso de desarrollo del software.
- Aplicar las técnicas más adecuadas al proyecto para la construcción técnica de cada uno de los artefactos generados de acuerdo al proceso de desarrollo de software seleccionado.
- Verificar y validar la calidad del producto entregado al cliente.
- Organizar y distribuir las tareas entre los integrantes del equipo de desarrolladores.
- Establecer criterios específicos para evaluar el avance del proyecto y favorecer la comunicación y coordinación entre todos los miembros del equipo.

Contenido

1. Conceptos básicos.

- 1.1. Objetivos de las pruebas.
- 1.2. Modelos generales de prueba.
- 1.3. Relación de pruebas con el aseguramiento de calidad.
- 1.4. Validación y verificación.

2. Técnicas de prueba estáticas.

- 2.1. Tipos de prueba estática.
- 2.2. Revisiones técnicas formales
- 2.3. Aspectos de calidad.

3. Técnicas de prueba dinámica

- 3.1. Casos de prueba.
- 3.2. Pruebas de caja negra.
- 3.3. Pruebas de caja blanca.
- 3.4. Planeación de la prueba.

4. Robustez y prueba

- 4.1. Corrección y robustez
- 4.2. Diseño de la prueba.
- 4.3. Verificación de programas.

5. Métodos formales.

- 5.1. Redes de Petri
- 5.2. Máquinas de estado finito.
- 5.3. EOI cálculo lambda.
- 5.4. El cálculo de predicados.
- 5.5. Lattices.

Estrategias Didácticas

- Propiciar la vinculación del alumno con el sector productivo al solicitarle que construya un sistema de software que sea utilizado por un cliente real.
- Promover la colaboración de los alumnos al permitirle la elección de distintas responsabilidades de trabajo, de acuerdo a las diversas tareas involucradas en el proyecto.
- Promover la participación oral y escrita de todos los alumnos al discutir el ámbito de los proyectos que aborden y especificar los requisitos de usuario en un documento.
- Implementar diversas técnicas de revisión, formales e informales, para evaluar el avance del proyecto y la calidad de los productos generados.
- Promover la investigación bibliográfica y en línea de información necesaria para comprender el dominio del problema abordado en el proyecto por el alumno así como de aspectos técnicos relacionado con la construcción del sistema de software a entregar.

Estrategias de Evaluación

Para la evaluación de los estudiantes, el instructor tomará en cuenta:

- Resultados de los exámenes parciales aplicados (de acuerdo a la teoría abordada, el maestro decidirá la cantidad de éstos).
- La calidad de los productos generados en el proceso de desarrollo de software y que se entreguen de acuerdo al plan de desarrollo establecido.
- participación individual y colectiva en las revisiones.
- Proyecto completo terminado y aprobado por el cliente.
- Los porcentajes de cada uno de los indicadores serán acordados al inicio del semestre.

Bibliografía

La bibliografía básica del área de ingeniería del software es la siguiente:

- Sommerville, Ian. *Ingeniería de software*. 6ta. ed. Pearson Educación, México, 2002.
- Pressman, Roger. *Ingeniería del software: Un enfoque práctico*. 5ta. ed. McGraw Hill, Madrid, 2002.
- Jacobson, I., Booch, G. y Rumbaugh, J. *El proceso unificado de desarrollo de software*. Pearson Educación, Madrid, 2000.

Otros libros del área que pueden utilizarse son los siguientes:

- Booch, Grady, Rumbaugh, J. y Jacobson, I. *The Unified Modeling Language User Guide*. Addison Wesley, 1998.

Perfil académico deseable del maestro

Se recomienda que el profesor tenga las siguientes características:

- Formación sólida en ciencias de la computación o área afín y se haya especializado en ingeniería de software.