

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	Ingeniería de Software II
CLAVE	9453
EJE FORMATIVO	Profesional
REQUISITOS	Ingeniería de software I
CARÁCTER	Obligatoria
VALOR EN CRÉDITOS	8 (3 hr. teoría/2 hr. 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 comprenderán la importancia del desarrollo sistemático del software y el impacto que tiene entregar un producto a tiempo, con la funcionalidad y calidad requerida por el cliente, y con los recursos que se establezcan al principio del semestre.

Objetivo General del Curso

El alumno será capaz de comprender los principios de administración de proyectos de desarrollo de software y de aseguramiento de calidad para aplicarlos en un proyecto, en donde, el alumno junto con sus compañeros establezcan esquemas de organización y división del trabajo, seguimiento de actividades terminadas, elección de técnicas para construir cada uno de los artefactos del ciclo vital elegido por el alumno así como mantener bajo control el proyecto y entregar un producto a tiempo, con las capacidades solicitadas por el cliente y con los recursos establecidos al inicio del proyecto.

Objetivos Específicos del Curso

- Establecer con claridad el ámbito del software en el proyecto.
- Realizar la planificación del proyecto fundamentada en las necesidades de procesamiento de información de un cliente real.
- Adaptar el proceso de desarrollo de software de acuerdo a la cantidad de desarrolladores de software, recursos disponibles y complejidad del producto solicitado por el cliente.
- 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 el proyecto y favorecer la comunicación y coordinación entre todos los miembros del equipo.

Contenido

- 1. Panorama general**
 - 1.1. Introducción a la ingeniería de software
 - 1.2. Proceso de desarrollo de software
- 2. Conceptos sobre gestión de proyectos**
 - 2.1. Introducción
 - 2.2. Personal
 - 2.3. Producto
 - 2.4. Proceso
 - 2.5. Proyecto
- 3. Planificación de proyectos de software**
 - 3.1. Objetivos
 - 3.2. Ámbito del software
 - 3.3. Recursos
 - 3.4. Estimaciones
 - 3.5. Análisis y gestión del riesgo
 - 3.6. Plan del proyecto y seguimiento de actividades
- 4. Aseguramiento de la calidad del software**
 - 4.1. Conceptos de calidad
 - 4.2. Revisiones de software
 - 4.3. Gestión de la configuración del software
- 5. Análisis y diseño de software**
 - 5.1. Obtención y análisis de requerimientos
 - 5.2. Modelos de objetos
 - 5.3. El documento de requerimientos
 - 5.4. Diseño orientado a objetos
 - 5.5. Diseño de la interfaz de usuario
- 6. Implementación**
 - 6.1. Estrategias de implementación
 - 6.2. Documentación de código
 - 6.3. Pruebas dinámicas de código
- 7. Legado del proyecto**

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:

- Larman, Craig. UML y Patrones. Introducción al análisis y diseño orientado a objetos. Prentice Hall, México, 1999.
- Fowler, Martin. UML gota a gota. Addison Wesley Longman de México, México, 1999.
- Sintés, Anthony. Aprendiendo programación orientada a objetos en 21 lecciones avanzadas. Pearson Educación, México, 2002.
- Jacobson, I., Booch, G. y Rumbaugh, J. El proceso unificado de desarrollo de software. Pearson Educación, Madrid, 2000.
- Booch, G. Análisis y diseño orientado a objetos con aplicaciones. 2da ed. Addison-Wesley Iberoamericana, México, 1996.
- Rosenberg, Doug y Scott, Kendall. Use Case Driven Object Modeling with UML: A practical approach. Addison Wesley, Boston, 1999.
- Rosenberg, Doug y Scott, Kendall. Applying Use Case Driven Object Modeling with UML: An annotated e-commerce Example. Addison Wesley, Boston, 2001.
- 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.
- Conozca los principios de aseguramiento de calidad del software y sea capaz de aplicarlos en la revisión de los distintos artefactos generados en cada una de las etapas del proceso de desarrollo de software.