



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

Ingeniería de Software

Carrera: Ingeniería en Computación
Profesor Responsable: Pesado, Patricia
Año: 3º
Duración: Semestral
Carga Horaria Semanal: 9hs
Carga Horaria Total: 144hs

Objetivos Generales

Introducir al alumno en los conceptos fundamentales de la Ingeniería de Software. Profundizar las etapas del ciclo de vida (requerimientos, análisis, diseño, codificación, pruebas). Introducir los conceptos de re-ingeniería e ingeniería inversa. Estudiar los temas de gestión, planificación y evaluación de proyectos de software, incluyendo el análisis de riesgo. Presentar los conceptos de calidad de software. El alumno deberá desarrollar sistemas concretos utilizando las metodologías/herramientas estudiadas.

Contenidos Mínimos

El proceso de software. Ciclos de vida de software. Herramientas para el proceso de software. Ingeniería de requerimientos. Introducción a los métodos formales. Metodologías de Análisis. Diseño e Implementación. Verificación y validación. Mantenimiento. Interacción hombre-máquina. Reingeniería e ingeniería inversa. Gestión de proyectos. Planificación. Métricas. Estimaciones. Análisis y gestión del riesgo. Conceptos de calidad de software.

PROGRAMA

1. Conceptos de software e ingeniería de software.

- Evolución del software. Características. Componentes.
- Definición de Ingeniería de Software. Evolución.
- Software de alta calidad.
- Ingeniería de Sistemas.

2. Procesos del Software.

- El significado de proceso.
- Modelos de proceso. Modelo de cascada.
- Iteración de procesos. Modelos incrementales. Modelos Evolutivos.
- Prototipación. Metodologías ágiles. Desarrollo basado en componentes.
- Métodos formales. Desarrollo orientado a aspectos. Proceso unificado.
- Actividades del Proceso. Especificación. Diseño. Implementación. Validación. Evolución.
- Herramientas y técnicas para modelado de procesos.



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

- Ingeniería de Software Asistida por Computadora.

3. Ingeniería de Requerimientos.

- El proceso de requerimientos.
- Tipos de requerimientos. Requerimientos funcionales, no funcionales, del usuario, del sistema.
- Características de los requerimientos. Obtención y análisis de los requerimientos.
- Técnicas de comunicación. Los problemas de la comunicación. Elicitación de requisitos. Entrevistas, cuestionarios, JAD, brainstorming.
- Validación de requerimientos. Gestión de requerimientos. Medición de requerimientos.
- Documentos de Especificación de requerimientos. STD 830.

4. Modelos del Sistema.

- Modelo de contexto, de comportamiento, de datos, de objetos.
- Técnicas de especificación de requerimientos: estáticas, dinámicas, relacionales, orientadas a estados, formales.
- Tablas de Decisión, Diagramas de Transición de Estados, Redes de Petri...
- Casos de Uso, Diagramas de Flujos de Datos, Diagramas de Flujos de Control, UML.
- Prototipado de los requerimientos. Técnicas de construcción rápida.

5. Gestión de Proyectos

- Conceptos. El problema de las 4 "P" (personal, producto, proceso, proyecto).
- Actividades de gestión, planificación del proyecto, hitos y entregas. El plan de proyecto.
- Métricas y Estimaciones.
- Clasificación de las métricas. Métricas del proceso y del proyecto. Métricas orientadas al tamaño, a la función, a casos de uso. Recopilación, cálculo y evaluación de métricas.
- Estimación de proyectos. Técnicas de descomposición. Modelos empíricos (COCOMO). Decisión de desarrollar-comprar.
- Planificación Temporal: calendarización del proyecto, distribución del esfuerzo, redes de tareas, seguimiento de la planificación. Métodos PERT, Gantt.
- Planificación Organizativa: del equipo y del proyecto.
- Gestión del Riesgo: identificación de riesgos, proyección, impacto, reducción, supervisión y gestión. Planes de contingencia. El plan de RSGR.
- Gestión de la configuración del software: Línea base, gestión del cambio, control de versiones, auditoría.

6. Diseño e Implementación

- Conceptos. Abstracción, arquitectura, patrones, modularidad, ocultamiento de la información, independencia funcional, cohesión, acoplamiento, refinamiento.



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

- El modelo de diseño: diseño de datos, diseño arquitectónico, diseño de interfaz, diseño al nivel de componentes.
- Diseño Arquitectónico.
- Organización del sistema: modelo de repositorio, modelo cliente-servidor, modelo de capas. Arquitecturas de Sistemas Distribuidos: multiprocesador, c-s, objetos distribuidos, interorganizacional (peer-to-peer, sistemas orientados a servicios).
- Descomposición modular: orientada a objetos, orientada a flujos de funciones.
- Control: centralizado, dirigido por eventos.
- Diseño de interfaces de usuario: interacción del usuario, presentación de la información, análisis del usuario, prototipo de la interfaz, evaluación de la interfaz.
- Diseño a nivel de componentes: notaciones gráficas, notaciones tabulares, lenguajes de diseño.
- Características de un bien diseño. Técnicas para la mejora del diseño. Evaluación y validación del diseño. Documentando el diseño.
- Aspectos de Implementación

7. Implementación

- Estándares de programación y procedimientos
- Pautas para la programación
- Documentación

8. Verificación y Validación

- Técnicas de Prueba
- Pruebas de Caja blanca: camino básico, bucles.
- Pruebas de Caja negra: partición equivalente, análisis de valores límites.
- Estrategias de Prueba
- Defectos y fallas. Planificación. Diseño de casos de prueba. Resultados. Documentación de las pruebas. Automatización.
- Pruebas de unidad (arquitecturas convencionales y arquitecturas orientadas a objetos)
- Pruebas de integración (arquitecturas convencionales y arquitecturas orientadas a objetos)
- Pruebas de validación: alfa y beta.
- Pruebas del sistema: de recuperación, de seguridad, de resistencia, de desempeño.
- Pruebas de regresión.
- La depuración: proceso, estrategia, corrección del error.

9. Entrega

- Entrenamiento
- Documentación



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

10. Mantenimiento

- Evolución del software. Tipos de mantenimiento: correctivo, adaptativo, perfectivo, preventivo.
- Sistemas heredados.
- Métricas, técnicas y herramientas para el mantenimiento.
- Rejuvenecimiento del software: redocumentación, reestructuración, ingeniería inversa, reingeniería.

11. Calidad.

- Conceptos de Calidad y Calidad Total.
- Calidad del proceso y del producto.
- Aseguramiento y estándares de calidad.
- Planeamiento de la calidad.
- Control de la calidad.
- Revisiones Técnicas formales. Inspecciones.
- Modelos de madurez para las organizaciones de desarrollo de software (CMM-CMMI).
- Normas ISO 9000.

BIBLIOGRAFIA

Título: Ingeniería de Software.

Autores: Ian Sommerville.

Editorial: Pearson - Addison Wesley.

Año de edición: 2011

Título: Ingeniería de Software. Teoría y Práctica.

Autores: Shari Pfleeger.

Editorial: Pearson Education.

Año de edición: 2002

Título: Ingeniería de Software. Un enfoque práctico.

Autores: Roger Pressman.

Editorial: McGraw Hill.

Año de edición: 2010.

Título: Análisis y Diseño de Sistemas.

Autores: Kendall y Kendall.

Editorial: Pearson Education.

Año de edición: 2011.

Título: Análisis de sistemas diseño y métodos.

Autores: Whitten – Bentley.



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

Editorial: McGraw Hill.
Año de edición: 2008.

Título: Calidad de Sistemas Informáticos.
Autores: Piattini, García, Caballero.
Editorial: Alfaomega.
Año de edición: 2007. 2011

Descripción de las Actividades Teóricas y Prácticas

El curso tiene clases teóricas, explicaciones de prácticas y el desarrollo de trabajos prácticos. Las clases teóricas son dictadas por los Profesores de la asignatura. Las clases teóricas se dictarán una vez por semana, la asistencia es obligatoria sólo para los alumnos que aspiren al régimen de promoción. Las explicaciones de práctica son dictadas por los Jefes de Trabajo Prácticos, quienes también coordinan la realización del trabajo práctico y las actividades de los auxiliares de la asignatura. Las actividades prácticas comprenden el desarrollo de un sistema utilizando metodologías ágiles que contemplan todas las etapas de desarrollo de software.

Metodología de Enseñanza y Evaluación

Durante la cursada, los alumnos realizarán un trabajo práctico, con múltiples entregas, en grupo de 3/4 alumnos, cada grupo tendrá un docente asignado y un horario fijo por semana con asistencia obligatoria. La composición del grupo se mantendrá a lo largo de toda la cursada. Adicionalmente los alumnos contarán con un horario semanal general para consultas. La asignatura interactuará con los alumnos a través de la Plataforma WEBUNLP ("curso de Ingeniería de Software") y/o el blog de la cátedra.

Aprobación de Cursada

Los alumnos deberán aprobar el trabajo práctico grupal evolutivo, tener un 80 % de asistencia a los horarios fijos asignados y aprobar el coloquio individual (con 2 recuperatorios)

Régimen de promoción

Los alumnos podrán optar por el régimen de promoción, para lo cual deberán (adicionalmente a las condiciones de aprobación de cursada): cumplir con el 80% de asistencia a las clases teóricas y aprobar los exámenes de promoción (que se tomarán en las clases teóricas) con promedio de 7 (siete). En caso de no alcanzar un promedio de 7 (siete) podrán acceder a un recuperatorio integrador. La nota final de promoción será la nota promedio obtenida de los exámenes de promoción y/o del recuperatorio integrador.



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

De no aprobar el régimen de promoción el alumno podrá rendir el final de la asignatura.

Examen Final de la Asignatura

Habiendo aprobado la cursada, los alumnos podrán rendir el final (teórico-práctico) de la asignatura durante el período de vigencia de la cursada (en los plazos fijados en la Resol. 4/2010 y las fechas de examen final definidos en el calendario académico)