



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA



PROGRAMA DE ESTUDIO

ADMINISTRACIÓN DE PROYECTOS DE SOFTWARE		6	8	
Asignatura	Clave	Semestre	Créditos	
INGENIERÍA ELÉCTRICA		INGENIERÍA EN COMPUTACIÓN		
División		Departamento		
Asignatura: Obligatoria	<input checked="" type="checkbox"/> X	Horas/semana: Teóricas	<input type="text"/> 4.0	
Optativa	<input type="checkbox"/>	Prácticas	<input type="text"/> 0.0	
		Total	<input type="text"/> 4.0	
			Horas/semane: Teóricas	<input type="text"/> 64.0
			Prácticas	<input type="text"/> 0.0
			Total	<input type="text"/> 64.0

Modalidad: Curso teórico

Seriación obligatoria antecedente: Ingeniería de Software

Seriación obligatoria consecuente: Ninguna

Objetivo(s) del curso:

El alumno organizará con un enfoque sistemático y disciplinado la administración de procesos y costeo de la ingeniería de software.

Temario

NÚM.	NOMBRE	HORAS
1.	Administración de la configuración del software	12.0
2.	Administración básica de proyectos de software	16.0
3.	Procesos de ingeniería de software	20.0
4.	Evolución y tendencias	16.0
		64.0
	Actividades prácticas	0.0
	Total	64.0

1 Administración de la configuración del software

Objetivo: El alumno identificará la función de la administración y configuración del software de los diferentes puntos de vista para los sistemas implantados.

Contenido:

1.1 Introducción.

- 1.1.1 Concepto de administración de la configuración (AC).
- 1.1.2 Guías y limitación en el proceso de AC.
- 1.1.3 Recursos y perfiles de la AC.
- 1.1.4 Definición de estructuras para la AC.

1.2 Planeación de la AC.

- 1.2.1 Calendarización y plan de la AC.
- 1.2.2 Administración de liberaciones.
- 1.2.3 Auditoria de la AC.
- 1.2.4 Documentación de la AC.

1.3 Herramientas para la AC.

- 1.3.1 Versionamiento.

1.4 Líneas base.

- 1.4.1 Creación detallada de líneas base.
- 1.4.2 Contenido mínimo de una línea base.
- 1.4.3 Periodicidad de la línea base.
- 1.4.4 Validación de respaldos de líneas base.

1.5 Procesos de equipos de trabajo.

- 1.5.1 Interacción de personal en los procesos.
- 1.5.2 AC de software y su interacción con los stakeholders.
- 1.5.3 PCMM.

2 Administración básica de proyectos de software

Objetivo: El alumno aprenderá a administrar las actividades de planeación, coordinación, medidas, monitoreo, control y reportes para asegurar que el desarrollo y mantenimiento del software sea sistemático, disciplinado y cuantificable.

Contenido:

2.1 Medición de la administración del software.

- 2.1.1 Definición de métricas de estimación del desarrollo, procesos.
- 2.1.2 Definición y análisis de la matriz de riesgo.
- 2.1.3 Definición y análisis de desvíos y ajustes.
- 2.1.4 Evaluación de mediciones.
- 2.1.5 Creación de reportes de métricas de administración de software.
- 2.1.6 Inicio y definición del alcance.

2.2 Ejecución y control de los requerimientos.

- 2.2.1 Planificación del proyecto del software.
- 2.2.2 Planeación de los procesos y entregas.
- 2.2.3 Técnicas para determinar la calendarización, costo y esfuerzo de la estimación.

2.2.4 Administración de recursos, riesgo, calidad y configuración.

2.3 Seguimiento del acuerdo de trabajo.

- 2.3.1** Administración de los recursos.
- 2.3.2** Implementación de los planes.
- 2.3.3** Administración de proveedores.
- 2.3.4** Monitoreo y control de procesos.
- 2.3.5** Reportes de proceso.

2.4 Cierre de proyectos.

- 2.4.1** Actividades durante el cierre.
- 2.4.2** Evaluación y reporte de la métrica global del proyecto.

3 Procesos de ingeniería de software

Objetivo: El alumno aplicará las actividades de administración del ciclo de vida del software como un proceso de Ingeniería y lo adaptará a cualquier organización para producir software de calidad, adecuadamente documentado.

Contenido:

3.1 Definición de procesos.

- 3.1.1** Modelos para el ciclo de vida del software.
- 3.1.2** Procesos para el ciclo de vida del software.
- 3.1.3** Notaciones para las definiciones de los procesos.
- 3.1.4** Adaptación y automatización de procesos.
- 3.1.5** Técnicas de medición de procesos.
- 3.1.6** Herramientas para procesos de ingeniería de software.

3.2 Métodos para la ingeniería de software.

- 3.2.1** Métodos formales.
- 3.2.2** Métodos heurísticos.

3.3 Proceso de documentación de arquitecturas de software.

- 3.3.1** Documentación de arquitecturas de software: orientadas a servicio, cliente-servidor, multi-capas.
- 3.3.2** Documentación de vistas, estilos y tipos de vista.
- 3.3.3** Refinamiento, diagramas de contexto, documentación de interfaces.

3.4 Roles y equipos de desarrollo de software.

- 3.4.1** Procesos y desarrollo de software en general.
- 3.4.2** Procesos y desarrollo de software para fábricas de software.

3.5 Administración de la calidad del software.

- 3.5.1** Aseguramiento de la calidad en el software.
- 3.5.2** Administración de procesos para la calidad del software.
- 3.5.3** Caracterización de los defectos.
- 3.5.4** Revisiones y auditorías.
- 3.5.5** Técnicas para la administración de software de calidad.
- 3.5.6** Mediciones de la calidad en el software.
- 3.5.7** Consideraciones prácticas.

3.6 Documentación de software.

- 3.6.1 Tipos de básicos documentación.
- 3.6.2 Documentación durante el ciclo de desarrollo de software.
- 3.6.3 Estándares de documentación.
- 3.6.4 Usabilidad de la documentación.
- 3.6.5 Distribución de la documentación.

4 Evolución y tendencias

Objetivo: El alumno explicará los conceptos asociados con la evolución de los sistemas de software y elaborará una prospectiva del software.

Contenido:

4.1 Sistemas heredados.

- 4.1.1 Estructura y componentes del sistema.
- 4.1.2 Categorías.

4.2 Evolución del software.

4.3 Tendencias del software.

- 4.3.1 En la industria.
- 4.3.2 En la investigación y academia.
- 4.3.3 Resumen y aplicación del CMMI-DEV.
- 4.3.4 Resumen y aplicación del MoProSoft.

Bibliografía básica

Temas para los que se recomienda:

ABRAN, Alain

Software Metrics and Software Metrology

Todos

New Jersey

Wiley, 2010

ALEXANDER, Ian, STEVENS, Richard

Writing Better Requirements

Todos

Addison-Wesley, 2002

BASS, Len, CLEMENTS, Paul, et al.

Software Architecture in Practice

Todos

2nd edition

Boston Massachusetts, USA

Addison-Wesley Professional, 2003

BECK, Kent

Extreme Programming Explained: Embrace Change

Todos

2nd edition

Massachusetts

Addison-Wesley, 2004

BERLACK, Ronald		
<i>Software Configuration Management</i>	Todos	
John Wiley & Sons, 1992		
BUCKLEY, Fletcher		
<i>Implementing Configuration Management: Hardware, Software, and Firmware</i> 2nd edition	Todos	
Los Alamitos, CA		
IEEE Computer Society Press, 1996		
DART, Susan		
<i>Spectrum of Functionality in Configuration Management</i>	Todos	
Systems Carnegie Mellon University		
Software Engineering Institute, 1990		
DAVIS, Alan		
<i>Software Requirements: Objects, Functions and States</i>	Todos	
2nd edition		
Prentice-Hall, 1994		
MIDHA, Anil		
<i>Software Configuration Management for the 21st Century</i>	Todos	
Bell Labs Technical Journal Winter, 1997		
Vol. 2, ISS. 1, pp. 154-165		
PERRY, William		
<i>Effective Methods for Software Testing</i>	Todos	
3th edition		
Editorial Wiley, 2006		
PFLEGER, Shari		
<i>Software Engineering: Theory and Practice</i>	Todos	
4th edition		
Prentice Hall		
ROBERTSON, Suzanne, ROBERTSON, James		
<i>Mastering the Requirements Process: Getting Requirements Right</i> 3rd edition	Todos	
New Jersey		
Addison-Wesley Professional, 2012		
SANCHEZ, Salvador, SICILIA, Miguel Ángel, et al.		
<i>Ingeniería de software</i>	Todos	
Alfaomega, 2012		
SOMMERRVILLE, Ian		
<i>Software Engineering</i>	Todos	
9th edition		

Boston Massachusetts, USA

Addison-Wesley, 2010

TAKANG, Armstrong, GRUBB, Penny

Software Maintenance: Concepts and Practice

Todos

2nd Edition

MA, USA

World scientific publishing company, 2003

Bibliografía complementaria

Temas para los que se recomienda:

BECK, Kent

Extreme Programming Explained: Embrace Change

Todos

2nd edition

Massachusetts

Addison-Wesley, 2004

Sugerencias didácticas

Exposición oral	<input type="checkbox"/>	Lecturas obligatorias	<input type="checkbox"/>
Exposición audiovisual	<input type="checkbox"/>	Trabajos de investigación	<input type="checkbox"/>
Ejercicios dentro de clase	<input type="checkbox"/>	Prácticas de taller o laboratorio	<input type="checkbox"/>
Ejercicios fuera del aula	<input type="checkbox"/>	Prácticas de campo	<input type="checkbox"/>
Seminarios	<input type="checkbox"/>	Búsqueda especializada en internet	<input type="checkbox"/>
Uso de software especializado	<input type="checkbox"/>	Uso de redes sociales con fines académicos	<input type="checkbox"/>
Uso de plataformas educativas	<input type="checkbox"/>		

Forma de evaluar

Exámenes parciales	<input type="checkbox"/>	Participación en clase	<input type="checkbox"/>
Exámenes finales	<input type="checkbox"/>	Asistencia a prácticas	<input type="checkbox"/>
Trabajos y tareas fuera del aula	<input type="checkbox"/>		

Perfil profesiográfico de quienes pueden impartir la asignatura

Licenciatura en Ingeniería en Computación, Ciencias de Computación, Matemáticas Aplicadas o una carrera similar. Deseable haber realizado estudios de posgrado, contar con conocimientos y experiencia en el área de Ingeniería de Software, contar con experiencia docente o haber participado en cursos o seminario de iniciación en la práctica docente.