

**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO****Licenciatura en Ciencias de la Computación****Facultad de Ciencias**

Programa de la asignatura

**Denominación de la asignatura:*****Pruebas de Software y Administración de la Configuración***

Clave:	Semestre: 7-8	Eje temático: Ingeniería de Software			No. Créditos: 10
Carácter: Optativa		Horas		Horas por semana	Total de Horas
Tipo: Teórico-Práctica		Teoría:	Práctica:	7	112
		3	4		
Modalidad: Curso		Duración del programa: Semestral			

Asignatura con seriación indicativa antecedente: Ingeniería de Software**Asignatura con seriación indicativa subsecuente:** Ninguna**Objetivo general:**

Comprender, explicar y transmitir la importancia de las pruebas de software bajo una administración de la configuración adecuada, proporcionando los elementos necesarios apoyados en técnicas y herramientas que permitan asegurar el control de versionado y niveles de calidad del software requeridos.

Índice temático

Unidad	Temas	Horas	
		Teóricas	Prácticas
I	Introducción	1.5	2
II	Administración de la configuración	12	16
III	Pruebas de software	21	28
IV	Administración de la configuración y las pruebas de software	3	4
V	Herramientas para pruebas de software	10.5	14
Total de horas:		48	64
Suma total de horas:		112	

Contenido temático

Unidad	Tema
I Introducción	
I.1	Importancia de la administración de la configuración y realización de pruebas como aspectos de la calidad en los productos Ubicación de la administración de la

	configuración y realización de pruebas dentro del ciclo de desarrollo.
I.2	Ubicación de la administración de la configuración y realización de pruebas dentro del ciclo de desarrollo.
I.3	Beneficios.
II Administración de la configuración	
II.1	Conceptos básicos. <ul style="list-style-type: none"> Definición y objetivos de la administración de la configuración. Psicología de la administración de la configuración. Importancia y aportación de la administración de la configuración. Línea base.
II.2	Proceso de la administración de la configuración.
II.3	Administración de la configuración en el ciclo de vida de desarrollo de software.
II.4	Proceso de control de cambios.
II.5	Administración de la configuración en el modelo de procesos CMMI.
II.6	Auditoría de la administración de la configuración.
III Pruebas de software	
III.1	Conceptos básicos. <ul style="list-style-type: none"> Psicología de las pruebas. Error, defecto y falla. Cómo identificar si es o no un defecto. Importancia y costo de las pruebas. Principios de las pruebas.
III.2	Calidad de los productos de software (ISO 25010).
III.3	Pruebas en el ciclo de vida de desarrollo del software.
III.4	Proceso general de las pruebas (IEEE 829).
III.5	Pruebas de software en los modelos de procesos. <ul style="list-style-type: none"> CMMI Verificación y validación. Pruebas en MoProSoft.
III.6	Evaluación del proceso de pruebas (TMM y TPI).
III.7	Inicio y término de las pruebas.
III.8	Revisiones técnicas.
III.9	Tipos de pruebas (características, selección de las pruebas acorde a las necesidades). <ul style="list-style-type: none"> Pruebas de caja blanca. Pruebas de caja negra. Pruebas de regresión. Pruebas de usabilidad. Pruebas no funcionales. Pruebas de aceptación.
III.10	Técnicas de pruebas. <ul style="list-style-type: none"> Caja blanca.

	<ul style="list-style-type: none"> • Caja negra.
III.11	Pruebas de usabilidad. <ul style="list-style-type: none"> • Usuario y cliente. • Identificación de elementos de usabilidad. • Estandarización en los elementos de la pantalla. • Accesibilidad.
III.12	Pruebas no funcionales. <ul style="list-style-type: none"> • Diseño de las pruebas. • Monitoreo e interpretación de los resultados.
IV Administración de la configuración y las pruebas de software	
IV.1	En el ciclo de vida de desarrollo de software.
IV.2	Proceso de la administración de la configuración y pruebas de software.
V Herramientas para pruebas de software	
V.1	Administración de la configuración.
V.2	Pruebas de software. <ul style="list-style-type: none"> • Generación de datos. • Pruebas de regresión. • Pruebas no funcionales.

Bibliografía básica:

1. Glenford J. Myers. *The Art of Software Testing*. John Wiley y Sons, Inc., Segunda edición 2004.
2. SWEBOK *Guide to the Software Engineering Body of Knowledge*, A project of the IEEE Computer Society Professional Practices Committee, capítulo 5 y 7.
3. ISO/IEJ JTC1/SC7 N4522, 2010-01-04, *Software and Systems Engineering*. IEEE Std 829-1998.
4. C/S2ESC – Software & Systems Engineering Standards Committee, *829 – IEEE Standard for Software and System and Test Documentation*, IEEE Computer Society, 2008.
5. Specialist Group in Software Testing (BCS SIGIST), *BS7925, Standard for Software Component Testing*, British Computer Society, 2003.
6. Koomen, T., Pol., M; *Test Process Improvement: A step by step guide to structured testing*, Pearson Education Limited, 1999.
7. TMM Testing Maturity Model, U.S., Air force magazine Crosstalk, 1996.
8. CMMI, Capability Maturity Model Integration for Development, V.1.2, 2006.

Bibliografía complementaria:

1. Banford, Robert y William J. Deibler; *Software Configuration Management*. Technology Report, STSC, 1994.
2. Estublier Jacky (Ed.) *System Configuration Management*. Springer 1999.
3. Kenareh, Mansour. *Creating an Adaptable Software Configuration Management (SCM) Infrastructure*. Alltel, SEPG 2002.

4. Leon Alexis. *A Guide to Software Configuration Management*. Artech House, 2000.
5. Joseph Defeo and J.M. Juran. *Juran's Quality Handbook*. McGraw-Hill. 2010.
6. Roger S. Pressman. *Software engeneering: A practitioner's approach*. 7Th edition. McGraw-Hill. 2009
7. Marnie L. Hutcheson. *Software Testing Fundamentals, Methods and Metrics*. Wiley Publishing, Inc. 2003.

Sugerencias didácticas:		Métodos de evaluación:	
Exposición oral	(X)	Exámenes parciales	(X)
Exposición audiovisual	(X)	Examen final escrito	(X)
Ejercicios dentro de clase	(X)	Trabajos y tareas fuera del aula	(X)
Ejercicios fuera del aula	(X)	Prácticas de laboratorio	()
Seminarios	()	Exposición de seminarios por los alumnos	()
Lecturas obligatorias	(X)	Participación en clase	()
Trabajo de investigación	()	Asistencia	()
Prácticas de taller o laboratorio	(X)	Proyectos de programación	()
Prácticas de campo	()	Proyecto final	()
		Seminario	()
Otras: _____		Otras: _____	
Perfil profesiográfico: Egresado preferentemente de la Licenciatura en Ciencias de la Computación o Matemático con especialidad en computación con amplia experiencia de programación. Es conveniente que posea un posgrado en la disciplina. Con experiencia docente.			