



Universidad
Tecnológica
de Pereira



UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA FÍSICA Y CIENCIAS DE LA
COMPUTACIÓN
PROGRAMA INGENIERIA DE SISTEMAS Y COMPUTACION
AÑO 2010

ASIGNATURA: Ingeniería del Software I

CÓDIGO: IS714

PRERREQUISITO: IS663 Bases de Datos I y IS463 Teoría de Sistemas

INTENSIDAD: 6 horas semanales

· **JUSTIFICACIÓN**

El software es uno de los tres fundamentos de todo programa del área de la informática y la computación, y los profesionales de dicha área, deben incluir en su perfil habilidades y destrezas que le permitan, liderar proyectos de desarrollo de software, conocer los fundamentos en los cuales se basan las técnicas y metodologías para desarrollar software a nivel industrial, administrar la configuración del software, definir el proceso para construir software, definir las herramientas y métodos para crear software, asegurar la calidad del proceso y del producto, conocer las áreas de conocimiento relacionadas.

Para proveer tales competencias se incluyen en el programa la propuesta de la IEEE Computer Society la cual establece, de manera amplia en el SWEBOK, los temas esenciales que debe conocer el profesional en ingeniería de software.

2. OBJETIVO

2.1 Objetivo General

Presentar los temas necesarios para que al terminar el curso el estudiante desarrolle las competencias propias de la disciplina del desarrollo de software, independientemente de las herramientas tecnológicas disponibles en el mercado.

2.2 Objetivos Específicos

Comunicar los conceptos relacionados con la ingeniería de software de tal forma que al final del curso el estudiante esté en capacidad de:

- Utilizarlos adecuadamente en sus diversas formas de expresión oral y escrita.
- Realizar el análisis de un problema, modelar los requisitos del software, modelar una visión arquitectónica de la solución y luego realizar el diseño detallado y los módulos que componen dicha solución.
- Crear los documentos de análisis de requerimientos, diseño del sistema.
- Administrar la configuración del software como una estrategia para manejar el cambio.
- Definir el proceso para la producción de software.
- Conocer herramientas y métodos para la producción de software.
- Conocer qué conocimientos de campos relacionados se necesitan.
- Tener la capacidad de escribir los documentos que requiere la ingeniería del software.

3. CONTENIDO

- **INTRODUCCIÓN A LA INGENIERÍA DE SOFTWARE**
 - Presentación del curso

- Marco Histórico
- Historia del lenguaje de modelado unificado
- REQUERIMIENTOS DE SOFTWARE
 - Fundamentos sobre requerimientos de software
 - Requerimientos de procesos
 - Levantamiento de requerimientos
 - Norma IEEE830
 - Análisis de requerimientos (INGENIERIA DE REQUERIMIENTOS)
 - Modelo de casos de uso
 - Diagramas de secuencia
 - Patrones grasp (UML Y PATRONES)
 - Colaboración de clases
 - Diagramas de actividades
 - Transición de estados
 - Modelo de clases
 - Especificación de requerimientos no funcionales
 - Validación de requerimientos
- DISEÑO DE SOFTWARE (Capítulo 3 del SWEBOK)
 - ARQUITECTURAS.
 - PRINCIPIOS DE ARQUITECTURA DE SOFTWARE.
 - Modelo de arquitectura de software.
 - ADL (Lenguajes de definición de arquitectura).
 - Enfoques arquitectónicos de software: arquitectura por capas, las 4+1 vistas de Kruchten.
 - Modelado Arquitectónico en UML 2.0.
 - SOA (Services Oriented Architecture).
 - Arquitecturas para computación móvil.
 - Arquitectura web
 - Fundamentos del diseño de software
 - Temas clave en el diseño de software
 - Estructura y arquitectura del software (DEFINICION)
 - Análisis y evaluación de la calidad del diseño de software
 - Notaciones para diseño de software
 - Estrategias y métodos para diseño de software
 - Modelo interfaz H-M
 - Modelo de componentes
 - Modelo de distribución
 - Patrones Arquitectónicos
- CONSTRUCCIÓN DE SOFTWARE (Capítulo 4 del SWEBOK)
 - Fundamentos de la construcción de software
 - Administrando la construcción
 - Consideraciones prácticas
 - Estándares de documentación
 - Generación automática de código
 - Refactoring REFACTORING Y REINGENIERÍA
 - Principios de refactoring
 - Malas sensaciones en el código
 - Componer métodos
 - Mover características entre objetos
 - Organizar datos
 - Simplificar expresiones condicionales

- PRUEBAS DE SOFTWARE (Capítulo 5 del SWEBOK)
 - Fundamentos de la prueba del software
 - Niveles de prueba
 - Técnicas de prueba
 - Medidas relacionadas con la prueba
 - Proceso de la prueba
- MANTENIMIENTO DE SOFTWARE (Capítulo 6 del SWEBOK)
 - Fundamentos del mantenimiento de software
 - Temas clave en mantenimiento de software
 - Proceso del mantenimiento
 - Técnicas para mantenimiento

4. METODOLOGÍA

Los temas se exponen en forma magistral y se complementan con ejemplos presentados por el profesor y ejercicios propuestos por el profesor y desarrollados por los estudiantes.

5. EVALUACION

Se realizarán 3 parciales y un final.

Nota:

- El tercer parcial será la nota acumulativa de talleres, quices y tareas durante el semestre hasta el último día de clases.
- El examen final si bien puede tener los mismos temas, no debe ser el mismo para todos los grupos pues los enfoques y profundidad en los diversos temas son diversos para cada docente. Por sus diversos perfiles, y experiencias.
- Todas las evaluaciones tienen igual valor (25%).

6. BIBLIOGRAFÍA

BELLAGIO, David E. MILLIGAN, Tom J. **Software Configuration Management Strategies and IBM Rational ClearCase Second Edition A Practical Introduction.** Addison Wesley Professional, IBM Press, 2005, EEUU. ISBN 0-321-20019-5.

BOURQUE, Pierre. DUPUIS, Robert. **SWEBOK Versión 2004: Guide to Software Engineering Body of Knowledge.** IEEE Computer Society, EEUU, 2004. ISBN 0-7695-2330-7.

BRITO E ABREU, Fernando. POELS, Geert. SAHRAOUI, Houari A. ZUSE, Horst. **Quantitative Approaches in Object-Oriented Software Engineering.** Hermes Penton Science, 2001, París. ISBN 1-90399-627-9.

DICKELMAN, Gary J. **EPSS Revisited—A Lifecycle for Developing Performance-Centered Systems.** International Society for Performance Improvement, 2003, United States of America. ISBN: 1-890289-15-9.

EVANS, Isabel. **Achieving Software Quality through Teamwork.** Artech House Ing, London, 2004. ISBN 1-58053-662-X.

GALORATH, Daniel D. EVANS, Michael W. **Software Sizing, Estimation and Risk Management: When performance is measured, performance improves.** Auerbach Publications, EEUU, 2006. ISBN 0-8493- 3593-0.

GOODMAN, Allan F. **Defining and Deploying Software Process.** Auerbach Publications, 1a Ed. 2006, EEUU. ISBN 0-8493-9845-2.

HALLOWS, Jolyon. **Information Systems Project Management: How to Deliver Function and Value in Information Technology Projects.** 2a Ed. AMACOM: American Management Association, EEUU, 2005. ISBN: 0814472737.

HILBURN, Thomas B. HIRMANPOUR, Hiraj. KHAJENOORI, Soheil. TURNER, Richard. QASEM, Abir. **A Software Engineering Body of Knowledge Versión 1.0.** Software Engineering Institute, Cargenie Mellon University, EEUU, 1999.

KENDALL, Keneth E. KENDALL, Julie E. **Análisis y Diseño de Sistemas.** 3ª Edición, México, Prentice Hall Hispanoamericana, 1997, ISBN: 968-880-694-4.

LAIRD, Linda M. BRENNAN, M. Carol. **Software Measurement and Estimation.** IEEE Computer Society, EEUU, 2006. ISBN 0-471-67622-5.

LAWRENCE, Shari. Ingeniería de software, teoría y práctica. Prentice Hall. Primera edición, 2002. ISBN: 987-9460-71-5

MARTIN, James. **Information Engineering. Book I: Introduction. Book II: Planning & Analysis. Book III: Design & Construction.** 1a Edición, Prentice Hall International, USA, 1989. ISBN BOOK I:0-13- 464462-X. ISBN BOOK II: 0-13-464885-4. ISBN BOOK III: 0-13-465501-X.

McCONNELL, Steve. Desarrollo y gestión de proyectos informáticos. McGraw-Hill, Madrid, 1997, ISBN: 1-55615-900-5.

MYERS, Glenford J. **The Art of Software Testing.** 2a Ed. John Wiley & Sons Inc. EEUU, 2004. ISBN 0- 471-46912-2.

PFLEEGER, Shari Lawrence. **Ingeniería de Software: Teoría y Práctica.** 2ª Edición, Argentina, Pearson Education S. A., 2002. ISBN: 987-9460-71-5.

POULIN, Louis. **Reducing Risk with Software Process Improvement.** Auerbach Publications, EEUU, 2005. ISBN 0-8493-3828-X.

PRESSMAN, Roger S. **Ingeniería del Software: Un enfoque práctico.** 4ª Edición, España, Mc Graw Hill, 1998, ISBN: 84-481-1186-9.

SOMMERVILLE, Ian. **Ingeniería de Software.** 6ª Edición, México, Pearson Educación de México S. A. De C. V., 2002. ISBN: 970-26-0206-8.

STAHL, Bernd Carsten. **Responsible Management of Information Systems.** Idea Group Publishing, EEUU, 2004. ISBN 1-59140-173-9.

WASSON, Charles S. **System Analysis, Design and Development: Concepts, principles and practices**. John Wiley & Sons, EEUU, 2006. ISBN 0-471-39333-7.

WHITTEN, Jeffrey L. BENTLEY, Lonnie D. BARLOW, Victor M. **Análisis y Diseño de Sistemas de Información**. 3ª Edición, España, Irwin Inc. 1996. ISBN: 84-8086-252-1.

WONG, Yuk Kuen. **Modern Software Review: Techniques and Technologies**. IRM Press and Idea Group Inc, EEUU, 2006. ISBN 1-59904-015-8.

6. PROGRAMACIÓN

Tema	Semanas
Presentación del curso e introducción a la ingeniería de software	1
Requerimientos de software (capítulo 2 del swebok)	4
Diseño de software (capítulo 3 del swebok)	4
Construcción de software (capítulo 4 del swebok)	1
Pruebas de software (capítulo 5 del swebok)	2
Mantenimiento de software (capítulo 6 del swebok)	1
Gestión de la configuración de software (capítulo 7 del swebok)	1
Personalización del software	1
Exámenes	1
Holgura para imprevisto	1
Total semanas	16