



Universidad Nacional Experimental
de los Llanos Occidentales
"Ezequiel Zamora"

CONTENIDO PROGRAMÁTICO DEL SUBPROYECTO:

METODOLOGIAS DE DESARROLLO DE SOFTWARE

VICERRECTORADO: Planificación y Desarrollo Social
PROGRAMA: Ingeniería, Arquitectura y Tecnología
SUBPROGRAMA: Ingeniería en Informática
CARRERA: Ingeniería en Informática
ÁREA DE
CONOCIMIENTO: Formación Profesional Específica
PROYECTO: Desarrollo de Software
CÓDIGO: II51002070703
PRELACIÓN: Principios de Ingeniería del Software y Algoritmos y Programación III
HORAS SEMANALES: 3 Horas: 3 Horas Teóricas y 2 Horas Teóricas -Prácticas
UNIDADES CRÉDITO: 04
SEMESTRE: VII
CONDICIÓN: Obligatoria (De Naturaleza Especial)
PERFIL DEL
DOCENTE: Ingeniero en Informática, Ingeniero de Sistemas, Licenciado en Computación, Ingeniero en Información, Licenciado en Informática, Ingeniero de Sistemas, Ingeniero en Computación, o afines
PROFESOR(ES)
DISEÑADOR(ES): Prof. José Rafael Quintero A.

Barinas, Julio 2008

JUSTIFICACIÓN

Puede entenderse una metodología de desarrollo de software como un conjunto de pasos y procedimientos que deben seguirse para desarrollar software; en general, las metodologías llevan a cabo una serie de procesos comunes que son buenas prácticas para lograr los objetivos de desarrollo de software. Estos procesos comunes pueden entenderse como un conjunto de pasos y procedimientos que deben seguirse para desarrollar software, los cuales de manera general permiten conocer y dividir un proyecto de desarrollo, determinar las tareas de desarrollo, aplicar las restricciones requeridas y utilizar herramientas diseñadas para el desarrollo de software. El conocimiento, definición y manejo de las teorías sobre las metodologías de desarrollo de software son fundamentales para el ejercicio profesional de los ingenieros en informática.

OBJETIVO GENERAL

Al finalizar el sub-proyecto, el estudiante estará en capacidad de manejar los conceptos básicos sobre las metodologías de desarrollo de software; podrá definir y clasificar las metodologías, conocerá los diversos métodos ágiles para el desarrollo de software, estudiará de forma específica las metodologías: métrica, XP , MSF. Conocerá los elementos y las principales características que conforman los métodos formales en el desarrollo de software.

MÓDULO I: INTRODUCCIÓN A LAS METODOLOGÍAS

Valor 25%

Objetivo Específico

1. Conocer los conceptos básicos relacionados con las metodologías de desarrollo de software.
2. Conocer las metodologías ágiles para el desarrollo de software.

Contenido

Definición de metodología de desarrollo de software, definición de procedimiento, técnica y herramienta.

Evolución histórica de las metodologías de desarrollo, descripción y características para: el desarrollo convencional, el desarrollo estructurado y el desarrollo OO.

Clasificación de las metodologías de desarrollo de software. Breve recuento sobre la evolución y las características de los procesos de desarrollo: cascada, espiral, iterativo, incremental y prototipos.

Surgimiento de las metodologías ágiles para el desarrollo de software.

Los métodos ágiles descripción y características: EXtreme Programming (XP), Scrum, Evolutionary Project Management, Crystal Methods, Feature Driven Development (FDD), Rational Unified Process (RUP), Dynamic Systems Development Method (DSDM), Adaptive Software Development, Agile Modeling, Lean Development (LD) y Lean Software Development (LSD).

La crítica a los Métodos Ágiles y los esfuerzos para la estandarización de los mismos.

MÓDULO II: CASO DE ESTUDIO: METRICA 3

Valor 20%

Objetivos Específicos

1. Conocer y manejar los conceptos básicos relacionados a METRICA 3 .
2. Entender y manejar la estructura principal de METRICA 3.

Contenido

Objetivos generales del método.

Ámbito de aplicación.

Alcance del método.

Versiones.

Objetivos en el desarrollo de la versión 3.

Modelo de procesos de Métrica 3.

Estructura Principal: Planificación de sistemas de información (PSI), Estudio de viabilidad del sistema (EVS), Análisis del sistema de información (ASI), Diseño del sistema de información (DSI), Construcción del sistema de información (CSI), Implantación y aceptación del sistema (IAS) y Mantenimiento de sistemas de información (MSI).

Interfaces, Técnicas y Participantes.

MÓDULO III: CASO DE ESTUDIO: PROGRAMACIÓN EXTREMA (XP)

Valor 15%

Objetivos Específicos

1. Presentar al estudiante los elementos que conforma la metodología XP.
2. Conocer y comprender las principales características de la metodología XP.

Contenido

Definición de XP como una metodología ágil, historia y principios básicos.

Elementos favorables de la metodología y sus características, Elementos en el proceso de desarrollo de XP, fases: cliente e interacción con los mismos, definición y planificación del proyecto, diseño y desarrollo de pruebas.

Ciclo de vida para un proyecto XP.

Definición y características de los elementos del ciclo de vida.

MÓDULO IV: CASO DE ESTUDIO: MICROSOFT SOLUTION FRAMEWORK.

Valor 15%

Objetivos Específicos

1. Definir los elementos que conforman el Microsoft Solution Framework (MSF).
2. Presentar al estudiante los elementos básicos relacionados a los modelos que conforman el Microsoft Solution Framework (MSF).

Contenido

Origen e historia del MSF.

Definición y principios fundamentales del MSF.

Gerencia de proyectos en MSF.

Definición y características de los modelos que conforman el MSF.

Ejemplos de aplicación de MSF.

MÓDULO V: MÉTODOS FORMALES

Valor 25%

Objetivos Específicos

1. Presentar al estudiante los elementos que conforman los métodos formales en el proceso de desarrollo de software.
2. Conocer y comprender las principales características de los métodos formales.

Contenido

Definición de método formal y sus implicaciones en la industria del software.

La importancia de los métodos formales en la Ingeniería del Software.

Limitaciones de los métodos formales en el desarrollo de software.

Estudio de los métodos formales: Especificación de comportamiento: autómatas, álgebra de procesos y lógica temporal; Especificaciones basadas en teoría de conjuntos: notación Z, VDM, y el método B; Especificaciones algebraicas: tipos de datos abstractos y semántica; Verificación de programas mediante aserciones: lógica de Hoare y lógica dinámica.

Reglas de la lógica de Hoare.

Aplicación de la lógica de Hoare a la prueba de programas: precondiciones y poscondiciones.

ESTRATEGIAS INSTRUCCIONALES

- Exposiciones orales por parte del profesor.
- Talleres de ejercicios.
- Exposición de casos para el análisis.

EVALUACION

Módulos	Tipo de Evaluación
I	Prueba Escrita
II	Prueba Escrita
III	Prueba Escrita
IV	Prueba Escrita
V	Prueba Escrita

BIBLIOGRAFÍA

ARTÍCULOS:

- “A Study of Software Methodology Analysis: “Great Taste or Less Filling””. Jeffrey L. Brewer. Proc ISECON 2005, v22 (Columbus OH): §2153 2005 EDSIG.
- “Métodos de desarrollo de software: el Desafío Pendiente de la Estandarización” Ricardo A. Gacitúa Bustos Theoria, Vol. 12: 23-42, 2003
- “Metodologías Ágiles en el Desarrollo de Software”. Canós José et al. Actas de las VIII Jornadas de Ingeniería de Software y Bases de Datos. Alicante 2003.
- “Microsoft Solution Framework Overview”. Microsoft White Paper. 2006.
- “The Fiction of Methodological Development: a field study of information system development”. Nandhakumar, J. and Avison, J. Information Technology & People 12(2): 176-191, 1999.

LIBROS:

- BECK Kent, “Extreme Programming Explained”, Ed. Addison - Wesley, USA 2006.
- LARMAN Craig, “Agile and Iterative Development”. Ed. Pearson Education USA 2004.
- Ministerio de Administraciones Públicas. Secretaría de Estado para la Administración Pública. Consejo Superior de Informática. «MÉTRICA versión 3». Ed. MAP, 1ª edición, 2001. España.
- PRESSMAN Roger, “Ingeniería del Software, un Enfoque Práctico”, Ed. McGraw Hill, 5ta edición, España 2002.
- SCHACH Sue, “Ingeniería de Software Clásica y Orientada a Objetos”, Ed. McGraw Hill Interamericana, 3ra edición, México 2006.
- SOMMERVILLE Ian, “Ingeniería del Software”, Ed. Addison Wesley, 7ma edición, España 2005.
- WEITZENFELD Alfredo, “Ingeniería de Software Orientada a Objetos con UML, Java e Internet”, Ed. Thomson, USA 2005. WEITZENFELD Alfredo,

“Ingeniería de Software Orientada a Objetos con UML, Java e Internet”, Ed.
Thomson, USA 2005.