

## **PROCESOS DE INGENIERÍA DE SOFTWARE 1** **(BISOFT-32)**

**Créditos: 4**

### **I. DESCRIPCIÓN DEL CURSO**

El curso repasa los procesos que acompañan el desarrollo de software de manera ingenieril, para luego conocer nuevas técnicas que representen un valor agregado para las organizaciones en el proceso de creación y mantenimiento de software.

### **II. OBJETIVO GENERAL**

Aplicar los procesos de Ingeniería del software en proyectos de desarrollo de software.

### **III. OBJETIVOS ESPECÍFICOS**

1. Entender una visión integral del proceso de desarrollo de software.
2. Ampliar el estudio de las principales técnicas y herramientas utilizadas al realizar la ingeniería de un producto de software.
3. Profundizar en el entendimiento del proceso de desarrollo de software en los niveles personal y de equipo (PSP & TSP).
4. Generar documentación pertinente, de acuerdo con los entregables que prescriba una metodología o un estándar.
5. Llevar a cabo un proceso completo de ingeniería del software de calidad:
  - Administración de Riesgos,
  - Administración de la Calidad,
  - Administración de la Configuración,
  - Administración del Tiempo,
  - Administración del Personal (Individuos y Equipos),
  - Verificación, Validación y Pruebas,
  - Métricas y mejoramiento continuo.
6. Realizar análisis y diseño de sistemas de software mediante modelos expresados en UML 2.x.
7. Discutir y compartir experiencias sobre ética y responsabilidad profesional en el ambiente laboral.
8. Conocer aspectos generales sobre integración de software, implantación de software y mantenimiento de software.

## **IV. CONTENIDO PROGRAMÁTICO**

### **1. INTRODUCCIÓN A LOS PROCESOS DE INGENIERÍA DEL SOFTWARE**

- a. Modelos de procesos de Ingeniería del software (organización, proyecto, equipo e individuo).
- b. Estándares de modelos de procesos (CMMI, ISO 15504, IEEE 12207, SPICE).
- c. Ciclos de vida y metodologías existentes de procesos de Ingeniería del software y su relación costo-beneficio.
- d. Conceptos básicos de administración de proyectos de Ingeniería del software.
- e. Organización del personal en proyectos de Ingeniería del software.

### **2. PLANEACIÓN DE PROCESOS DE INGENIERÍA DEL SOFTWARE**

- a. Detalle de actividades (WBS: desglose de la estructura del trabajo).
- a. Secuenciación de actividades.
- b. Distribución de recursos.
- c. Estimación y planeación de actividades.
- d. Plan de Tiempo.

### **3. ADMINISTRACIÓN DE LA CALIDAD DEL SOFTWARE**

- a. Nociones de calidad.
- b. Atributos de calidad e ISO 9126.
- c. Administración de la Calidad.
- d. Control de la Calidad.
- e. Aseguramiento de la Calidad.
- f. Roles del proceso de administración de la calidad.

### **4. ADMINISTRACIÓN DE RIESGOS EN PROYECTOS DE SOFTWARE**

- a. Identificación de Riesgos.
- b. Análisis cualitativo y cuantitativo de Riesgos.
- c. Planeación de la Respuesta al Riesgo (Plan de contingencia).
- d. Monitoreo y Control de Riesgos.

### **5. ADMINISTRACIÓN DE LA CONFIGURACIÓN**

- a. Definición de Administración de la Configuración.
- b. Información necesaria para Administrar la Configuración.
- c. Roles en el proceso de Administración de la Configuración.
- d. Prácticas clave de Administración de la Configuración.

### **6. CONCEPTOS BÁSICOS DE ESTIMACIÓN Y MÉTRICAS DE CALIDAD Y PRODUCTIVIDAD DEL SOFTWARE**

- a. Vistazo general a las métricas de calidad en el software.
- b. Aplicación de las 7 herramientas básicas de calidad en el desarrollo de software.
- c. Métricas dentro del proceso para pruebas del software.
- d. Estimación de tamaño, esfuerzo, tiempo y costo.
- e. Métricas de productividad en el desarrollo de software.
- f. Mejoramiento continuo de procesos de Ingeniería del software.

## **7. PRUEBAS DEL SOFTWARE**

- a. Principios generales.
- b. Técnicas utilizadas para probar.
- c. Organización para realizar pruebas.
- d. Infraestructuras disponibles para probar software.

## **8. SEGUIMIENTO Y CONTROL**

- a. Definiciones.
- b. Técnica de Valor ganado ('Earned Value').

## **9. TEMAS AVANZADOS DE INGENIERÍA DEL SOFTWARE**

- a. Ética y responsabilidad profesional.
- b. Desarrollo de carrera profesional y certificaciones (p.e. CSDP, CSQE).
- c. Proceso de desarrollo de software individual (PSP) y en equipo (TSP).
- d. Administración de las comunicaciones.
- e. La integración y los proyectos de Ingeniería del software.
- f. La implantación y los proyectos de Ingeniería del software.
- g. El mantenimiento y los proyectos de Ingeniería del software.

## **V. METODOLOGÍA Y RECURSOS PARA EL APRENDIZAJE**

El curso consta de exposiciones magistrales, prácticas dirigidas en clase y sesiones de discusión. El trabajo en clase se distribuye en exposiciones magistrales, prácticas y casos prácticos, así como en sesiones de discusión. El curso completa la base teórica para emprender futuros cursos de proyectos de Ingeniería del software o cursos avanzados de Administración de proyectos. El curso aprovecha las experiencias de los estudiantes en sus proyectos de Ingeniería del software.

El estudiante aprende a diseñar procesos de desarrollo de software adecuados a la organización en la que se encuentre y además a ser capaz de planearlo y darle seguimiento de la manera apropiada. Para esto, se estimula la presentación de borradores de diseño y su discusión.

Como lectura obligatoria del curso se utilizará el libro *"The Mythical Man-Month: essays on software engineering"*. Se harán comprobaciones de lectura oral y se pedirán resúmenes semanalmente para comprobar su lectura.

Además permitirá al estudiante fundamentar sus conocimientos en Ingeniería del software adquiridos en cursos anteriores y poder discutir temáticas de actualidad en el campo.

## **VI. REQUISITOS**

Ingeniería de Requerimientos.

## VII. EVALUACIÓN

| Concepto                        | Ponderación |
|---------------------------------|-------------|
| Trabajo en Equipo Práctico      | 20%         |
| Trabajo en Equipo Investigativo | 15%         |
| Examen Parcial Individual # 1   | 15%         |
| Examen Parcial Individual # 2   | 15%         |
| Examen Parcial Individual # 3   | 15%         |
| Exámenes y Tareas Cortas        | 20%         |
| <b>Total</b>                    | <b>100%</b> |

La evaluación de los trabajos individuales o de equipo considera no solo el desempeño técnico sino también las actitudes y disciplina de trabajo aplicados en su realización. En los trabajos que se realizan en equipo la nota será distribuida entre los miembros del equipo según su participación y su involucramiento.

El curso se aprueba obteniendo una nota final igual o superior a 70. Aquellos estudiantes que en sus dos primeros exámenes parciales obtengan una nota igual o superior a 90 (no el promedio), estarán eximidos del tercer examen parcial, no así de los exámenes y tareas cortas de esa materia.

La asistencia al curso es obligatoria.

## VIII. CRONOGRAMA

| Semana    | Contenido de Estudio  | Estrategias de Evaluación |
|-----------|---|---------------------------|
| Semana 1  | Introducción a los procesos de Ingeniería del software                |                           |
| Semana 2  | Introducción a los procesos de Ingeniería del software                |                           |
| Semana 3  | Planeación de procesos de Ingeniería del software                     |                           |
| Semana 4  | Planeación de procesos de Ingeniería del software                     |                           |
| Semana 5  | Administración de la Calidad del software                             |                           |
| Semana 6  | Administración de la Calidad del software                             | Examen Parcial 1          |
| Semana 7  | Administración de Riesgos en proyectos de software                    |                           |
| Semana 8  | Administración de la Configuración                                    |                           |
| Semana 9  | Administración de la Configuración                                    |                           |
| Semana 10 | Conceptos básicos de estimación y métricas de calidad y productividad |                           |

|           |  |   |
|-----------|--|---|
|           | del software   |   |
| Semana 11 | Conceptos básicos de estimación y métricas de calidad del software | Examen Parcial 2                                  |
| Semana 12 | Pruebas del software   |   |
| Semana 13 | Seguimiento y control  |   |
| Semana 14 | Temas avanzados de Ingeniería del software                         | Presentación Proyecto Investigación               |
| Semana 15 | Recapitulación y perspectivas                                      | Examen Parcial 3<br>Defender Proyecto<br>Práctico |

## IX. BIBLIOGRAFÍA

1. Arlow, Jim; Neustadt, Ila. UML and the Unified Process. Addison-Wesley 2002.
2. Braude Eric. Ingeniería del software. Una perspectiva orientada a objetos. Grupo Alfaomega 2003.
3. Brooks, Frederick P. The Mythical Man-Month: essays on software engineering, anniversary edition. Addison-Wesley, 1995.
4. Dolado Cosín, José Javier; Fernández Sanz, Luis. Medición para la gestión en la Ingeniería del software. Editorial RA-MA, 2000.
5. Garbanzo, Priscilla. Marco de análisis y comparación de metodologías de desarrollo de software: aplicación a Métrica 3 y el Rational Unified Process. Tesis, Universidad Politécnica de Valencia, 2003.
6. Hamlet, Dick; Maybee, Joe. The Engineering of Software, Technical Foundations for the Individual. Addison-Wesley, 2001.
7. Humphery, Watts. Introduction to the Team Software Process. Addison-Wesley, 2000.
8. Humphrey, Watts. A discipline for Software Engineering. Addison-Wesley, 2000.
9. Jacobson, Ivar; Booch, Grady; Rumbaugh, James. El Proceso Unificado de Desarrollo de software. Addison-Wesley 2000.
10. Jalote, P. CMM in practice: processes for executing software projects at Infosys. Addison-Wesley, 2000.
11. Juristo, Natalia. Proceso Software. Facultad de Informática, Universidad Politécnica de Madrid, 2000.
12. Kan, Stephen. Metrics and Models in Software Quality Engineering. Second Edition. Addison-Wesley 2003.
13. Larman, Craig. Applying UML and patterns; an introduction to object-oriented analysis and design and the Unified process. Prentice-Hall, 2002.
14. Larman, Craig. UML y Patrones. Prentice-Hall, 1999.
15. Pol, Martin; Teunissen, Ruud; van Veenendaal, Eric. Software Testing. A guide to the TMap Approach. Addison-Wesley, 2002.
16. Pressman, Roger S. Software Engineering, 5th Edition. Mc-Graw Hill, 2001.
17. Sommerville, Ian. Software Engineering, 6th Edition. Addison-Wesley, 2000.

18. Varios. The business of software (El negocio del software). IEEE Software Vol. 19, No. 6, Noviembre 2002.
19. Wieggers, Karl E. Peer Reviews in Software. A Practical Guide. Addison-Wesley, 2002.

## **X. PRÁCTICAS**

Sugeridas por el docente.

## **XI. ACERCA DE LA FACULTAD DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIONES: ADMINISTRACIÓN, COORDINADORES Y PROFESORES DE TIEMPO COMPLETO:**

La Facultad de Tecnologías de Información y Comunicaciones cuenta con profesores que brindan consulta de la mayoría de los cursos que se imparten en la carrera. Mediante el Centro de Apoyo Estudiantil, se pueden obtener citas para evacuar dudas o consultas. Sus horarios pueden ser consultados en la Facultad.

Los profesores de cada curso son el canal de comunicación inmediato para cualquier trámite que tenga que ver directamente con el mismo, seguido está la Dirección de la Carrera. Esto indica que ante cualquier eventualidad se debe respetar el canal de comunicación según lo establece el Reglamento de Régimen Estudiantil vigente.

Fecha de revisión y actualización: **23 de Noviembre del 2007.**

---