

## **1. OBJETIVOS:**

Incorporar los conceptos de indicador, métrica, medida, escala de medición, y proceso de medición. Entender la importancia de los indicadores de desempeño de procesos, su medición y seguimiento.

Conocer el proceso de Aseguramiento de la calidad del software.

Aprender los conceptos teórico-prácticos que se emplean en la etapa de testing de un proyecto de software.

Conocer las buenas prácticas inherentes a las áreas de proceso de Revisión, Inspección, Verificación y Validación de software.

Aprender metodologías y modelos del proceso de evaluación de productos *software*.

Interiorizarse en modelos, técnicas, métodos, herramientas y métricas que se usan en el proceso de evaluación de productos *software*. Proporcionar mecanismos que faciliten las estimaciones en proyectos de evaluación de software.

Conocer las técnicas más recientes como ser el Desarrollo Dirigido por las Pruebas (TDD por su sigla en Inglés *Test Driven Development*) y su relación con los métodos *Ágiles*.

## **2. CONTENIDOS:**

A continuación se describe con mayor detalle cada una de las unidades:

### **UNIDAD 1:**

#### **La Calidad y su importancia en las TICs**

- a. Conceptos de calidad aplicados a las tecnologías de la información
- b. Aproximación de la calidad en el Software: Calidad en la Gestión, Calidad de los procesos de Tecnologías de la información y Calidad de Productos de TI.
- c. Normas y modelos de calidad de TI.

### **UNIDAD 2:**

#### **Calidad en el desarrollo, adquisición, operación y mantenimiento del software**

- a. Modelos de referencia de procesos, modelos de evaluación de procesos, modelos de mejora de procesos.
- b. Mejora de procesos de software (SPI por su siglas en inglés).
- c. Modelos y normas de referencia de procesos (Por ejemplo: ISO 9000, COMPETISOFT, CMMI-DEV, ISO/IEC 12207):
  - Sistemas de Gestión de la calidad según la norma IRAM-ISO 9001:2008 y su guía de aplicación la IRAM-ISO/IEC 90003:2006. Foco en los procesos

- de Revisión, Inspección, Verificación y Validación del software.
- Modelo COMPETISOFT: áreas de aplicación y beneficios, procesos asociados al modelo. Foco en los procesos de Revisión, Inspección, Verificación y Validación del software.
  - Capability Maturity Model Integrated for Development (CMMI-DEV), representación continua y por etapas, comparación entre las representaciones continua y por etapas, estructura del modelo, componentes: niveles, áreas de proceso, metas específicas y genéricas, prácticas específicas y genéricas, notas. Foco en las áreas de proceso de Validación y Verificación.
  - La Norma ISO/IEC 12207:2008 Ingeniería de software y sistemas – Procesos de ciclo de vida del software: campo de aplicación, beneficios, descripción de los procesos citados en la norma. Foco en las áreas de proceso de Revisión, Verificación y Validación.
- d. Métodos de evaluación basados en modelos y/o normas. Los procesos de certificación y evaluación. Por ejemplo: ISO 19011, SCAMPI, ISO/IEC 15504
- e. El proceso de mejora, normas y modelos de mejora de procesos.

### **UNIDAD 3:**

#### **Métricas de Software: de Producto y de Procesos**

- a. El proceso de medición y su importancia
- b. La necesidad de utilizar métricas en los procesos de desarrollo y evaluación de software. Principales conceptos: métrica, medida, medición, escala de medición. Características de una buena métrica. Métricas de proceso, y de producto.
- c. Las métricas utilizadas como indicadores de gestión de Sistemas de Información. Medición y estimaciones de atributos de procesos. Métricas para calidad y satisfacción del usuario.
- d. Las métricas a lo largo del ciclo de vida de desarrollo de software.
- e. Mediciones y predicciones.
- f. Utilización de las métricas para estimaciones en proyectos de evaluación de software.

### **UNIDAD 4:**

#### **El proceso de Testing (pruebas) de software**

- a. El proceso de testing y su importancia en el ciclo de vida de desarrollo de sistemas software. El testing como herramienta de verificación y validación.
- b. Planificación de las pruebas, diseño de los casos de prueba, ejecución, registro y seguimiento de las pruebas
- c. Desarrollo dirigido por casos de prueba, trazabilidad durante las pruebas
- d. Técnicas de testing

- e. Tipos de testing
- f. Estrategias de prueba
- g. Investigación de herramientas de testing open source y propietarias.
- h. Diseño de un procedimiento de testing y su aplicación en un caso práctico.

#### **UNIDAD 5:**

##### **Calidad del producto software y su método de evaluación**

- a. Modelos de calidad del producto *software*: Mc Call, Boehm, FURPs, ISO/IEC 9126-1.
- b. Vistas de la calidad: calidad interna, externa, y en el uso.
- c. Proceso de evaluación de productos de software definido en la norma ISO/IEC 14598-1.
- d. Ejemplos de métricas externas definidas en la ISO/IEC 9126-2 para evaluar la calidad externa del producto de software. Su aplicación en un caso práctico.

#### **UNIDAD 6:**

##### **El aseguramiento de la calidad en software**

- a. Aseguramiento de la calidad (QA por sus siglas en Inglés) y Aseguramiento de la calidad del software (SQA).
- b. Visión estratégica del Control de calidad (**QC**), aseguramiento de la calidad (**QA**) y gestión de la calidad (**QM**).
- c. Aseguramiento de la calidad según las normas y modelos de calidad citadas (ISO 9000, CMMI-DEV, ISO/IEC 12207:2008). Procesos de certificación.
- d. Relación entre el proceso de inspección del software y el proceso de aseguramiento de la calidad

#### **UNIDAD 7:**

##### **Los procesos de gestión de la configuración y gestión de los cambios**

- a. El proceso de gestión de la configuración y su aplicación en estándares (ISO 9000, CMMI-DEV, COMPETISOFT, ISO/IEC 12207, ISO/IEC 20000)
- b. El proceso de gestión de cambios y su aplicación en estándares (ISO 9000, CMMI-DEV, COMPETISOFT, ISO/IEC 12207, ISO/IEC 20000)
- c. La importancia de los procesos de gestión de la configuración y de gestión de cambios durante el desarrollo de software
- d. La importancia de los procesos de gestión de la configuración y de gestión de cambios durante el proceso de evaluación de software

#### **UNIDAD 8:**

##### **La calidad y los Métodos ágiles**

- a. Desarrollo Dirigido por las Pruebas (TDD por su sigla en Inglés Test Driven Development) y su relación con los métodos Ágiles.

### **3. BIBLIOGRAFIA**

#### **3.1 Básica**

Pressmann Roger, Ingeniería de software: un enfoque práctico, *7a edition*, Pearson Education, S.A., Madrid 2007.

Sommerville Ian, *Ingeniería del Software, 7a edition*, Pearson Education, S.A., Madrid 2005. ISBN: 84-7829-074-5.

#### **3.2 Adicional**

Beizer Boris, Software Testing Techniques, Van Nostrand Reinhold Company New York, 1983.

IEEE Guide to the Software Engineering Book of Knowledge – 2004 Version.

ISO/IEC 12207:2008 Software and system engineering - software life cycle processes.

ISO/IEC 15504 Information technology -- Process assessment -- Part 2: Performing an assessment

ISO/IEC 9126-1:2001 Software engineering – Product quality – Part 1: Quality model.

ISO/IEC 9126-2:2003 Software engineering – Product quality – Part 2: External metrics

ISO/IEC 14598-1: 1999 Information Technology – Software product evaluation – Part 1: General overview.

ISO 9001:2008 Quality management systems – Requisites.

ISO/IEC 90003: 2004 Information technology – Guidelines for the application of ISO 9001 to software

PMI Standards Committee, Project Management Body of Knowledge, Project Management Institute, 1995

Direcciones de Internet:

1. Carnegie Mellon University, Software Engineering Institute (SEI): [www.sei.cmu.edu](http://www.sei.cmu.edu)
2. Association for Computing Machinery (ACM): [www.acm.org](http://www.acm.org)
3. The Institute for Electrical and Electronic Engineers (IEEE): [www.ieee.org](http://www.ieee.org)

#### **3.3 Software necesario para desarrollar las clases**

1. Microsoft Office
2. Herramientas CASE: por ejemplo Enterprise Architect, Trial edition Enterprise Architect (SPARX Systems):  
<http://www.sparxsystems.com/products/ea/trial.html>

User's Guide: <http://www.sparxsystems.com/bin/EAUUserGuide.pdf>

3. Herramientas de testing Open source o propietaras, por ejemplo de IBM

#### **4. METODOLOGIA DE LA ENSEÑANZA**

Se realizarán presentaciones conceptuales teóricas sobre cada tema y presentación de casos prácticos de aplicación.

Los alumnos, organizados en grupos de 3 a 4 personas, deberán elaborar trabajos prácticos de seguimiento de cada tema, y entregables de un proyecto de diseño de un producto de software en base a lo visto en clase (metas de proyecto). Los proyectos serán presentados a sus compañeros, analizados, comentados y defendidos, analizados y las decisiones de diseño defendidas en las clases de trabajos prácticos.

Además deberán confeccionar una monografía (paper) sobre algún tema planteado en el curso y efectuar una presentación en clase del tema seleccionado.

#### **5. CRITERIOS DE EVALUACION**

- 1 Examen parcial (teórico y práctico)
- 2 Examen parcialito homologable optativo (si el grupo lo decide, se puede tomar un parcialito antes del primer parcial y las respuestas correctas se dan por homologadas en el parcial, manteniendo su nota relativa).
- 3 Trabajos prácticos, metas de proyecto, paper y presentaciones requeridos durante el cuatrimestre. Los mismos serán evaluados por grupo y con conceptos individuales según el grado de compromiso de cada alumno.
- 4 Examen final (teórico y práctico)
- 5 Presentismo (conforme las disposiciones vigentes de la UB)