

# Plan de Estudios

## Maestría en

### Ingeniería de Software

#### Antecedentes y Fundamentación

La **Ingeniería de Software (IS)** es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, mantenimiento y operación del software. Los principios y prácticas de esta disciplina son esenciales para el desarrollo de sistemas grandes, complejos y confiables.<sup>1</sup>

La IS es una disciplina con un rico cuerpo de conocimiento, práctica y teoría. Esta disciplina ha madurado muy rápidamente. Desde 1968, cuando el término ingeniería de software fue usado por primera vez, la disciplina de IS ha generado numerosas revistas, conferencias, sociedades profesionales, programas de grado y posgrado, certificaciones profesionales y estándares.<sup>2</sup>

Debido a que las funcionalidades de los productos y sistemas crecen, se necesita implementar eficientemente y correctamente el complejo software que posibilita este crecimiento. Todas las medianas y grandes compañías así como las agencias de gobierno tienen sistemas sofisticados que son críticos para gestionar sus proyectos y la propia organización. Estos sistemas dan soporte a procesos del negocio y técnicos. Debido a que hoy en día gran parte de su funcionalidad está implementada en software, una gran parte de la responsabilidad de esta implementación recae en los especialistas en Ingeniería de Software.

Entonces, es necesario contar con una Maestría Profesional en el área. Esta maestría busca cubrir un espacio aún no cubierto por la Universidad de la República, siendo una continuación con un alto valor agregado en su formación profesional para aquellos estudiantes que hayan egresado de la Especialización en Ingeniería de Software. Contar con esta maestría permitirá que las empresas desarrolladoras de software logren acortar cronogramas, reducir costos y mejorar la calidad de sus productos; factores todos esenciales para que sean competitivas en el mercado.

#### 1. Objetivos

El objetivo fundamental que persigue el presente Plan de Estudio es la formación de profesionales altamente capacitados en IS al grado de maestría, para analizar y resolver problemas complejos en dicha área. Esta Maestría en IS se dirige a profesionales en informática, que deseen especializarse en el área IS. Particularmente el público objetivo son los:

- Profesionales del área de desarrollo de software
- Profesionales que actúan en grupos de mejora de la calidad de procesos de software
- Líderes (o Gerentes) de proyectos de Software, de desarrollo de software, de pruebas de software u otras áreas de la disciplina de IS o afines.

---

<sup>1</sup> IEEE Standard Glossary of Software Engineering Terminology, *IEEE Std 610.12-1990*, vol., no., pp.1, 1990

<sup>2</sup> A. Pyster (ed.), Graduate Software Engineering 2009 (GSWE2009) Curriculum Guidelines for Graduate Degree Programs in Software Engineering, Integrated Software & Systems Engineering Curriculum Project, Stevens Institute, September 30, 2009.

Esta maestría brinda una formación mucho más profunda y específica en temas de actualidad dentro de las distintas disciplinas de la Ingeniería de Software que la que se brinda en el grado.

La Maestría apunta a lograr profesionales capaces de abordar con solvencia la resolución de problemas de importancia en el área. Se busca promover la actualización de los conocimientos en las distintas disciplinas de la IS, así como generar recursos humanos capaces de afrontar y resolver problemas de la industria nacional de software, aumentando así la calidad de servicios y de vida de la sociedad.

El trabajo de tesis se orienta al manejo activo del conocimiento en el marco de un tema concreto o aplicación específica, incluyendo el empleo de bibliografía actualizada, preferentemente aquella publicada en conferencias y/o revistas arbitradas y reconocidas en el área del tema de tesis.

## **2. Perfil del egresado**

El egresado adquirirá la capacidad de aplicar con profundidad y solvencia en su actividad profesional los temas de estudio incluidos en la maestría; asimismo, adquirirá los elementos metodológicos que, junto con la capacidad de abordar bibliografía especializada, le permitan comprender y emplear las nuevas tecnologías para la resolución de problemas relativos a la IS en su actividad profesional.

Entre otros el egresado tendrá la capacidad para:

- Dominar las áreas fundamentales de la IS.
- Ser capaz de tomar decisiones éticas y practicar un comportamiento ético profesional.
- Entender la relación entre IS e Ingeniería de Sistemas y ser capaz de aplicar principios y prácticas de la Ingeniería de Sistemas en la IS.
- Ser un integrante efectivo de un equipo, incluyendo equipos que están geográficamente distribuidos, pudiendo liderar un área del desarrollo o mantenimiento de software. Para esto, también tendrá la capacidad de comunicarse correctamente tanto de forma oral como escrita.
- Ser capaz de conciliar objetivos conflictivos de un proyecto, encontrando compromisos aceptables dentro de las limitaciones de costo, tiempo, conocimiento, sistemas existentes y organizaciones.
- Entender y valorar el análisis de factibilidad, la negociación y las buenas comunicaciones con los *stakeholders* de un ambiente de desarrollo de software típico. Ser capaz de realizar estas tareas de forma correcta, tener hábitos efectivos de trabajo y ser un líder.
- Aprender nuevos modelos, técnicas y tecnologías cuando éstas emergen, y apreciar la necesidad de ese desarrollo profesional continuo.
- Analizar tecnologías de software actuales, articular sus fuerzas y debilidades, compararlas con tecnologías alternativas y especificar y promover mejoras o extensiones a esas tecnologías.
- Ser capaz de analizar y profundizar en un tema concreto mediante una revisión bibliográfica en profundidad.

## **3. Ordenamiento**

La Sub Comisión Académica de Posgrado de Informática (SCAPA-Informática) supervisará las actividades ligadas al desarrollo de la Maestría en Ingeniería de Software; sin perjuicio de las competencias que correspondan a la Comisión Académica de Posgrado (CAP) y al Consejo de la Facultad de Ingeniería.

A propuesta de la SCAPA-Informática, la CAP nombrará para cada estudiante un Director Académico, responsable de la organización de las actividades y de la orientación del mismo.

Los aspectos reglamentarios no mencionados explícitamente se ajustan a lo establecido por los documentos: Ordenanza de las Carreras de Posgrado de la Universidad de la República y Reglamento General de las Actividades de Posgrado y Educación Permanente de la Facultad de Ingeniería (RGP-FING).

#### **4. Requisitos de ingreso**

Podrán ingresar a la Maestría en Ingeniería de Software quienes cumplan con al menos una de las siguientes condiciones:

**Condición 1:** Contar con un título de grado, en informática, otorgado por la Universidad de la República de al menos 360 créditos. Ejemplo: título de Ingeniero en Computación,

**Condición 2:** Contar con formación equivalente que, a juicio de la Comisión de Posgrado, permita la realización y aprovechamiento del Plan de Estudios de la Maestría en Ingeniería de Software. En este caso, la SCAPA-Informática podrá proponer la realización de cursos de nivelación en caso de ser necesario.

#### **5. Admisión y selección de los candidatos**

Las candidaturas deberán ser presentadas a la SCAPA-Informática, quien deberá elevar un informe a la CAP sugiriendo la aprobación o no de la candidatura. La admisión tendrá en cuenta los antecedentes del candidato, pudiéndose realizar una entrevista a los aspirantes para complementar la información presentada. La CAP resolverá la admisión de los candidatos en base a los antecedentes del candidato y al informe de la SCAPA-Informática.

#### **6. Formación**

Para cada estudiante, la SCAPA-Informática formulará una propuesta de plan de formación, que será aprobada por la CAP. Los planes de formación se integrarán con actividades programadas (cursos de posgrado, seminarios, etc.) y con la realización de una Tesis, de manera de cumplir:

- un mínimo de 70 créditos en actividades programadas (el crédito es la unidad de medida de la carga de trabajo en la Universidad de la República, y un crédito equivale a quince horas de dedicación por parte del estudiante) y
- contar con créditos en 6 Materias o áreas temáticas diferentes. Para este mínimo no cuenta la materia "Asignaturas sin materia específica".
- La actividad programada deberá cubrir el mínimo de horas presenciales exigido por la Ordenanza de las Carreras de Posgrado de la Universidad de la República.
- La realización de una Tesis la cual deberá ser defendida en una exposición oral y pública ante un tribunal designado por la CAP a propuesta de SCAPA-Informática.

Todas las actividades programadas deberán contar con alguna forma de evaluación de los conocimientos adquiridos.

Este número mínimo de créditos y la exigencia de contar con créditos en al menos 6 materias es imprescindible para poder transmitir el conocimiento necesario, tanto en amplitud como en profundidad, de forma de desarrollar las habilidades deseadas en el estudiante.

La duración prevista para la realización de la Maestría en Ingeniería de Software es de 24 meses, con una dedicación estimada de 20 horas semanales.

## 7. Estructura del plan de estudios

El plan de estudios de Maestría en IS está estructurado en Materias y Asignaturas (y otras actividades) que otorgan créditos dentro de las Materias. Una asignatura u otro tipo de actividad puede otorgar créditos en varias materias a la vez.

Las materias se presentan en el siguiente cuadro:

<b>Materia</b>
Ética y conducta profesional
Ingeniería de sistemas
Ingeniería de requisitos
Diseño de software
Construcción de software
Pruebas ( <i>testing</i> )
Mantenimiento de software
Gestión de la configuración
Gestión de la ingeniería de software
Procesos de ingeniería de software
Calidad de software
Asignaturas sin materia específica

## 8. Tesis

El estudiante deberá contar con un Director de Tesis y, eventualmente, un co-director, que deberá ser nombrado por la CAP a propuesta de la SCAPA-Informática a lo sumo un año después del ingreso a la Maestría.

El tema de la tesis se definirá en mutuo acuerdo entre el estudiante y el Director de Tesis, debiendo ser comunicado a la SCAPA-Informática para su validación. El trabajo de tesis tendrá un valor de 40 créditos.

En la elaboración de la Tesis, el estudiante deberá alcanzar el estado del arte y desarrollar un análisis en profundidad del tema elegido. El manuscrito de la Tesis deberá contener una correcta exposición del tema encarado, una discusión de la bibliografía actualizada (preferentemente aquella publicada en conferencias y/o revistas arbitradas y reconocidas en el área del tema de tesis) y deberá incluir los detalles necesarios para la comprensión de los objetivos perseguidos y de los resultados y conclusiones alcanzadas durante el trabajo.

Cuando el Director de Tesis considere que el trabajo ha sido culminado, el mismo se presentará por escrito y se defenderá en una exposición oral y pública ante un tribunal designado por la CAP a propuesta de la SCAPA-Informática, según lo establecido en el Reglamento General de

las Actividades de Posgrado y Educación Permanente de la Facultad de Ingeniería.

## **9. Título**

Cuando el aspirante haya completado los requisitos del programa, la SCAPA- Informática notificará a la CAP, quien propondrá al Consejo de la Facultad el otorgamiento del Título "Magister en Ingeniería de Software". Este título será firmado por el Decano de la Facultad de Ingeniería y el Rector de la Universidad de la República.

Aprobado por el Consejo de Facultad de Ingeniería  
en sesión de fecha 24/04/14.

Aprobado por el Consejo Directivo Central  
en sesión de fecha 18/11/14.

Publicado en el Diario Oficial el 15/12/14.

## Apéndice A. Conceptos Generales

La **Ingeniería de Software (IS)** es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, mantenimiento y operación del software. Los principios y prácticas de esta disciplina son esenciales para el desarrollo de sistemas grandes, complejos y confiables.<sup>3</sup>

La IS es una disciplina con un rico cuerpo de conocimiento, práctica y teoría. Esta disciplina ha madurado muy rápidamente. Desde 1968, cuando el término ingeniería de software fue usado por primera vez, la disciplina de IS ha generado numerosas revistas, conferencias, sociedades profesionales, programas de grado y posgrado, certificaciones profesionales y estándares.<sup>4</sup>

En el **Curriculum Guidelines for Graduate Degree Programs in Software Engineering (GSWE2009)** se presenta un currículo de referencia para programas de Maestría Profesional en IS. Este currículo es una guía para aquellas Facultades que están diseñando o mejorando sus programas de Maestrías Profesionales en Ingeniería de Software.

El GSWE2009 se ocupa principalmente de la educación de graduados que quieren obtener un título de Maestría Profesional en IS. Este título está destinado a un individuo cuyo interés primario es ejercer una carrera en la **práctica de la IS** y que no está necesariamente interesado en obtener un doctorado en IS o en campos relacionados.<sup>5</sup>

La evolución y mantenimiento de GSWE2009 es gestionado por la Association for Computing Machinery (ACM) y por el Institute of Electrical and Electronics Engineers (IEEE); esto garantiza la internacionalización de estas guías y posiblemente se transforme en base para la acreditación de programas de Maestrías Profesionales en IS en un futuro.

Por otro lado, distintas asociaciones, institutos, etc. han desarrollado certificaciones internacionales en la disciplina IS y en varias de sus subdisciplinas. Muchas de estas certificaciones comienzan a tener un valor profesional muy importante debido al cada vez más fuerte y marcado fenómeno de globalización. Para ejemplificar, en el Apéndice E se presentan distintas certificaciones internacionales y las instituciones que las expiden. Estas muestran cuanto ha madurado la Ingeniería de Software como profesión.

---

<sup>3</sup> IEEE Standard Glossary of Software Engineering Terminology, *IEEE Std 610.12-1990*, vol., no., pp.1, 1990

<sup>4</sup> A. Pyster (ed.), Graduate Software Engineering 2009 (GSWE2009) Curriculum Guidelines for Graduate Degree Programs in Software Engineering, Integrated Software & Systems Engineering Curriculum Project, Stevens Institute, September 30, 2009.

<sup>5</sup> *Ibíd.*

## **Apéndice B. La Importancia de la Ingeniería de Software**

El software es un componente crítico en casi cualquier producto nuevo; normalmente es el componente que distingue a un producto en el mercado. El software ha permitido avances tecnológicos que llevaron a nuevos productos y sistemas en todos los sectores comerciales, incluyendo: componentes médicos, automóviles, teléfonos celulares, agroindustria y sistemas de entretenimientos.

Así como las funcionalidades de los productos y sistemas crecen, también se necesita implementar eficientemente y correctamente el complejo software que posibilita este crecimiento. Todas las medianas y grandes compañías así como las agencias de gobierno tienen sistemas sofisticados que son críticos para gestionar sus proyectos y la propia organización. Estos sistemas dan soporte a procesos del negocio y técnicos. Debido a que hoy en día gran parte de su funcionalidad está implementada en software, una gran parte de la responsabilidad de esta implementación recae en los especialistas en Ingeniería de Software.

Debido a la complejidad del software y a las dificultades inherentes de su desarrollo, la mayoría de las “sorpresas” que ocurren durante la integración de sistemas o durante el funcionamiento en producción del software provienen de prácticas incorrectas de la IS.

## **Apéndice C. Consideraciones Generales**

*La industria de software es una de las industrias fundamentales del Uruguay productivo. [...] la preocupación en las empresas es que en los próximos años los recursos humanos capacitados no sean suficientes en calidad y cantidad.<sup>6</sup>*

Para que esta industria siga creciendo es necesario contar con un posgrado profesional en el área. Este posgrado busca cubrir un espacio aún no cubierto por la Universidad de la República, especializando en IS a los profesionales en informática. Contar con esta maestría permitirá que las empresas desarrolladoras de software logren acortar cronogramas, reducir costos y mejorar la calidad de sus productos; factores todos esenciales para que sean competitivas en el mercado.

GSwE2009 provee guías acerca del contenido de un programa de posgrado en IS y recomendaciones de cómo implementar esas guías, permitiendo de esta manera la creación de nuevos programas de posgrado profesionales en IS. Además, GSwE2009 forma parte de los currículos de la ACM y de la IEEE, garantizando de esta manera la calidad de la propuesta y la internacionalización de la misma. Este programa de Maestría en IS está basado en la guía GSwE2009. Esto dejará a esta Maestría en una posición excelente si en un futuro se realizan acreditaciones internacionales basándose en la guía.

Como ya se mencionó, existen diversas certificaciones internacionales en la disciplina de IS. Es importante que en el transcurso de esta Maestría el estudiante esté en condiciones de rendir algunos de estos certificados internacionales. Se busca de esta manera darle un mayor valor agregado a la Maestría así como también estandarizar los cursos del mismo según tendencias internacionales en la disciplina.

---

<sup>6</sup> Noticia de Presidencia de la República: [http://www.presidencia.gub.uy/\\_Web/noticias/2007/01/2007013104.htm](http://www.presidencia.gub.uy/_Web/noticias/2007/01/2007013104.htm)

## Apéndice D. Descripción de Materias

Las materias se corresponden a las *Key Areas* (Áreas Clave) presentadas en el GSwE2009 más una materia agregada “Asignaturas sin materia específica”. Esta materia incluye asignaturas que pueden integrar esta Maestría pero que no se corresponden con ninguna de las materias descritas.

Las descripciones de las materias, temas y subtemas son traducciones al español de las propuestas del GSwE2009 y del SWEBOK<sup>7</sup>.

<b>ÉTICA Y CONDUCTA PROFESIONAL</b>
Los ingenieros de software desarrollan y mantienen productos que se encuentran en casi todas las áreas del quehacer humano: medicina y salud, transporte y comunicaciones, negocios y finanzas, educación, gobierno y leyes, y arte y entretenimiento. Para lograr entregar productos de manera eficiente y efectiva, los ingenieros de software deben comportarse de forma ética y profesional. Esta materia esboza las cuestiones y elementos de dicha conducta.
<b>Temas</b>
<b>1. Cuestiones sociales, legales e históricas</b>
Confidencialidad de datos y seguridad, vigilancia y privacidad
Cuestiones de desarrollo histórico, sexo, minorías y culturas
Contratos y responsabilidad, propiedad intelectual y libertad de la información
Crímenes informáticos y aplicación de la ley
<b>2. Códigos de ética y conducta profesional</b>
Responsabilidad frente a la sociedad
Sociedades profesionales
Código de ética y práctica
<b>3. La naturaleza y el rol de los estándares de la ingeniería de software</b>
Naturaleza y rol de los estándares
Estándares internacionales
Cuerpos de conocimiento, prácticas aceptadas y mejores prácticas

---

<sup>7</sup> Guide to the Software Engineering Body of Knowledge; IEEE Computer Society, 2004.

## INGENIERÍA DE SISTEMAS

Un sistema es un conjunto de componentes interconectados que existen dentro de un entorno e interactúan con el mismo. Los ingenieros de sistemas analizan las necesidades, desarrollan los conceptos de la solución y trabajan junto con especialistas en componentes y especialistas en calidad de atributos del sistema (seguridad, costo, performance, etc.) para sintetizar la definición de sistemas complejos constituidos por diversos tipos de componentes. Los ingenieros en sistemas también juegan un rol importante en la instalación y en el soporte de estos sistemas en su entorno operativo, así como en su eventual remoción del servicio y eliminación segura. Prácticamente todos los sistemas modernos que van desde sistemas de control de tránsito aéreo, reactores nucleares y hasta los sistemas de transacciones financieras dependen del software para coordinar las interconexiones entre componentes del sistema y para proporcionar la funcionalidad de los mismos. Por lo tanto, los ingenieros de software son miembros clave en los equipos de ingeniería de sistemas modernos.

### Temas

#### 1. Conceptos de Ingeniería de Sistemas

Contexto del sistema

Personas y sistemas

Relaciones jerárquicas en los sistemas

El rol de los ingenieros de sistemas

#### 2. Gestión del Ciclo de Vida en Ingeniería de Sistemas

Gestión del ciclo de vida

Ingeniería de sistemas y procesos de ingeniería de software

#### 3. Requerimientos

Requerimientos de los *stakeholders*

Análisis de requerimientos

#### 4. Diseño de Sistemas

Diseño arquitectónico

Implementación

Estudios del negocio

#### 5. Integración y Verificación

#### 6. Transición y Validación

#### 7. Operación, Mantenimiento y Soporte

# INGENIERÍA DE REQUERIMIENTOS

La ingeniería de requerimientos es el área que se encarga de la obtención, análisis, especificación y validación de los requerimientos del sistema y del software. Los proyectos de ingeniería de software son sumamente vulnerables cuando las actividades de ingeniería de requerimientos se realizan mal. Los requerimientos de software expresan las necesidades y limitaciones de un producto de software que contribuyen a la solución de algunos problemas del mundo real.

## Temas

### 1. Fundamentos de Ingeniería de Requerimientos

Relación entre ingeniería de sistemas e ingeniería de software

Definición de requerimientos

Limitaciones del diseño de sistemas

Diseño de sistemas y asignación de requerimientos

Requerimientos del producto y del proceso

Requerimientos funcionales y no funcionales

Propiedades emergentes

Requerimientos cuantificables

### 2. Proceso de Ingeniería de Requerimientos

Modelos del proceso

Actores del proceso

Soporte y gestión del proceso

Calidad y mejora del proceso

### 3. Iniciación y Definición de Alcance

Determinación y negociación de los requerimientos

Análisis de factibilidad

Proceso para revisión de requerimientos

### 4. Extracción de Requerimientos

Fuentes de requerimientos

Técnicas de extracción

### 5. Análisis de Requerimientos

Clasificación de requerimientos

Modelado conceptual

Métodos heurísticos
Métodos formales
Negociación de requerimientos
<b>6. Especificación de Requerimientos</b>
Técnicas de especificación de requerimientos
<b>7. Validación de Requerimientos</b>
Revisión de requerimientos
Creación de Prototipos
Validación del modelo
Pruebas de aceptación
<b>8. Consideraciones Prácticas</b>
Naturaleza iterativa del proceso de requerimientos
Gestión de cambios
Atributos de requerimientos
Seguimiento de requerimientos
Medición de requerimientos

## DISEÑO DE SOFTWARE

Diseño es definido como “el proceso de definir la arquitectura, componentes, interfaces y otras características de un sistema o componente” o como el “el resultado de ese proceso”. Visto como un proceso, el diseño de software es la actividad del ciclo de vida de ingeniería de software en la cual los requerimientos son analizados con el fin de producir la descripción de la estructura interna del sistema que servirá como base para su construcción. Más precisamente, el diseño de software debe describir la arquitectura de software (es decir, cómo el software se descompone y se organiza en componentes) y las interfaces entre sus componentes. También debe describir los componentes con un nivel de detalle tal que permita su construcción.

El diseño de software juega un papel importante en el desarrollo de software: permite a los ingenieros de software producir diversos modelos que forman una especie de anteproyecto de la solución que se va a implementar. Podemos analizar y evaluar dichos modelos para determinar si nos permitirán o no cumplir con todos los requerimientos. También podemos examinar y evaluar varias soluciones alternativas. Por último, podemos usar los modelos resultantes para planificar el desarrollo posterior de actividades, además de utilizarlos como entrada y punto de partida para la construcción y el *testing*.

### Temas

#### 1. Fundamentos de Diseño de Software

Conceptos generales de diseño

Contexto del diseño de software

Proceso de diseño de software

Principios de diseño de software

#### 2. Cuestiones Clave en Diseño de Software

Concurrencia

Control y manejo de eventos

Distribución de componentes

Error, manejo de excepciones y tolerancia a fallas

Interacción y presentación

Persistencia de datos

#### 3. Estructura y Arquitectura de Software

Estructuras arquitectónicas y puntos de vista

Estilos arquitectónicos (patrones macro arquitectónicos)

Patrones de diseño (patrones micro arquitectónicos)

Diseño de la interfaz persona-computadora

Familias de programas y *frameworks*

<b>4. Análisis de Calidad y Evaluación de Diseño de Software</b>
Atributos de calidad
Análisis de calidad y técnicas de evaluación
Mediciones
<b>5. Notaciones en Diseño de Software</b>
Descripciones estructurales (estáticas)
Descripciones de comportamiento (dinámicas)
<b>6. Estrategias y Métodos de Diseño de Software</b>
Estrategias generales
Diseño orientado a funciones (estructurado)
Diseño orientado a objetos
Métodos heurísticos
Métodos formales
Diseño basado en componentes (CBD)

## CONSTRUCCIÓN DE SOFTWARE

La construcción de software se refiere a la creación de software significativo y que funcione a través de una combinación de codificación, verificación, pruebas unitarias, pruebas de integración y *debugging*.

### Temas

#### 1. Fundamentos de Construcción de Software

Minimización de complejidad

Anticipando el cambio

Construyendo para la verificación

Estándares de construcción

#### 2. Gestionando la Construcción

Métodos de construcción

Planificación de la construcción

Medición de la construcción

#### 3. Consideraciones prácticas

Diseño de la construcción

Codificación

*Testing* de la construcción

Calidad de la construcción

Integración

## **TESTING**

El *testing* (o pruebas de software) es una actividad realizada para evaluar la calidad del producto y mejorarla, mediante la identificación de defectos y problemas. El *testing* consiste en la verificación dinámica del comportamiento de un programa contra el comportamiento esperado, mediante un conjunto finito de casos de prueba debidamente seleccionados del dominio infinito de ejecuciones.

### **Temas**

#### **1. Fundamentos de *Testing***

*Testing* de sistema y *testing* de software

Terminología relacionada al *testing*

Cuestiones claves

Relaciones entre el *testing* y otras actividades

#### **2. Niveles de Pruebas**

El blanco del *testing*

Los objetivos del *testing*

Pruebas de componentes

Pruebas de integración

Pruebas de sistema

Pruebas de aceptación

#### **3. Técnicas de *Testing***

Basadas en la intuición y experiencia del verificador

Basadas en la especificación

Basadas en el código

Basadas en las fallas

Basadas en el uso

Basadas en la naturaleza de la aplicación

Selección y combinación de técnicas

#### **4. Mediciones Relacionadas a las Pruebas**

Evaluación del programa o sistema bajo prueba

Evaluación de las pruebas realizadas

#### **5. Proceso de Prueba**

Referente a la gestión
Actividades de pruebas

## MANTENIMIENTO DE SOFTWARE

Los esfuerzos realizados durante el desarrollo de software resultan en la entrega de un producto de software que satisface los requerimientos del usuario. A medida que pasa el tiempo, el producto de software debe cambiar o evolucionar. Una vez en funcionamiento, los defectos “salen a la luz”, hay cambios de entorno y surgen nuevos requerimientos del usuario. La fase de mantenimiento del ciclo de vida comienza luego de un período de garantía, pero las actividades de mantenimiento se producen mucho antes.

El mantenimiento de software se define como la modificación de un producto de software después de la entrega para corregir defectos, para mejorar el rendimiento u otros atributos, o para adaptar el producto a un entorno modificado.

### Temas

#### 1. Fundamentos de Mantenimiento de Software

Definiciones y terminología

Naturaleza del mantenimiento

Necesidad del mantenimiento

Costos de mantenimiento

Evolución del software

Categorías de mantenimiento

#### 2. Aspectos Clave en el Mantenimiento de Software

Técnicos

- Comprensión limitada

- *Testing*

- Análisis de Impacto

- Mantenibilidad

Aspectos de gestión

- Alineación con las cuestiones de la organización

Estimación de costos de mantenimiento

- Estimación de costos

- Modelos paramétricos

- Experiencia

Mediciones del mantenimiento de software

#### 3. Proceso de Mantenimiento

Modelos de proceso de mantenimiento
Actividades de mantenimiento
- Actividades únicas
- Actividades de soporte
<b>4. Técnicas para Mantenimiento</b>
Programa de comprensión
Reingeniería
Ingeniería inversa

## GESTIÓN DE LA CONFIGURACIÓN

La gestión de la configuración (GC) es la disciplina que identifica la configuración de un sistema en distintos instantes en el tiempo, con el propósito de controlar sistemáticamente los cambios en la configuración y mantener la integridad y trazabilidad de la configuración a través de todo el ciclo de vida del sistema.

### Temas

#### 1. Gestión del Proceso de la GC

Contexto organizacional para la GC

Limitaciones y orientación para la GC

Planificación de la GC

- Organización y responsabilidades de la GC
- Recursos y cronograma de la GC
- Control del proveedor / subcontratista
- Control de la interfaz

Plan de gestión de la configuración

Supervisión de la gestión de configuración

- Medidas y mediciones de la GC
- Auditorias *in-process* de la GC

#### 2. Identificación de la Configuración

Identificación de ítems a ser controlados

- Ítems de configuración
- Relación entre ítems de configuración
- Versiones
- Líneas base
- Adquisición de ítems de configuración

Librería de software

#### 3. Control de Configuración

Petición, evaluación y aprobación de cambios

- Tabla de control de configuración
- Proceso de petición de cambio

Implementación de cambios
Desviaciones y exenciones
<b>4. Estado de la Configuración</b>
Reporte de estado de configuración
<b>5. Gestión de la Liberación de Software y Entrega</b>
Construcción de software
Gestión de la liberación de software

## GESTIÓN DE LA INGENIERÍA DE SOFTWARE

La gestión de la ingeniería de software puede definirse como la aplicación de las actividades de gestión, como planificación, coordinación, medición, monitoreo, control y presentación de informes, para garantizar que el desarrollo y mantenimiento de software sea sistemático, disciplinado y cuantificado.

### Temas

#### 1. Planificación de un Proyecto de Software

Objetivos y metas del proyecto

Políticas y estándares del proyecto

Planificación del proceso

Suposiciones del proyecto y previsión

Entregables del proyecto

Personal del proyecto

Esfuerzo, cronograma y estimación de costo

Asignación de recursos

Gestión de la calidad

Plan / presupuesto del desarrollo y gestión del proyecto

#### 2. Gestión de Riesgos

Conceptos de gestión de riesgos

- Probabilidad, impacto

- Marco de tiempo

Proceso de gestión de riesgos

- Marcos de trabajo, estándares y guías

- Identificación de riesgos, análisis y técnicas de priorización de riesgos

- Estrategias de mitigación de riesgos

Herramientas de gestión de riesgos

- Registro y seguimiento del valor ganado

- Medición del performance técnico

- Registro, seguimiento y reporte de defectos

- Paneles de control del proyecto

Gestión del riesgo organizacional
Gestión de riesgos del proveedor / cliente
<b>3. Organización y Promulgación del Proyecto de Software</b>
Organización del proyecto
- Identificar y agrupar funciones, actividades y tareas del proyecto
- Determinar la estructura y posición organizacional
- Definir responsabilidades, relaciones de autoridad y calificaciones de posición
Dirección de proyecto
- Liderazgo, supervisión, delegación de autoridad, coordinación y comunicación
- Motivación, resolución de conflictos, construcción de equipos
Control del proyecto
- Implementación de planes y proceso de medición
- Monitoreo del proceso
- Gestión de cambios
Reporte
Gestión de contratos con el proveedor
<b>4. Revisión y Evaluación</b>
Determinación de la satisfacción de requerimientos
Revisión y evaluación de la performance
<b>5. Cierre</b>
Determinación de cierre
Actividades de cierre
<b>6. Mediciones en Ingeniería de Software</b>
Establecer y mantener el compromiso de medición
Planificar el proceso de medición
Ejecutar el proceso de medición
Evaluar las mediciones
<b>7. Economía en Ingeniería</b>

Fundamentos de la economía en ingeniería
Toma de decisiones con fines de lucro
Toma de decisiones sin fines de lucro
Economía actual
Estimación, riesgos e incertidumbre
Decisiones sobre múltiples atributos

## PROCESO DE INGENIERÍA DE SOFTWARE

La materia Proceso de ingeniería de software puede ser examinada en dos niveles diferentes. El primer nivel abarca las actividades técnicas y de gestión dentro de los procesos del ciclo de vida del software que son realizadas durante la adquisición, desarrollo, mantenimiento y retiro del software. El segundo nivel es un meta-nivel, que se ocupa de la definición, ejecución, evaluación, medición, gestión, cambio y mejora del propio ciclo de vida del proceso de software. El primer nivel es cubierto en las otras Materias. Esta materia se relaciona con el segundo nivel, el meta-nivel.

### Temas

#### 1. Implementación y cambio del Proceso

Infraestructura del proceso

- Grupo del proceso de ingeniería de software (SEPG)
- Experiencia

Modelos de implementación y cambio del proceso

Consideraciones prácticas

#### 2. Definición del Proceso

Modelos de ciclo de vida

Procesos de ciclo de vida de software

Notaciones para definiciones de procesos

Adaptación del proceso

Automatización

#### 3. Evaluación del Proceso

Modelos de evaluación del proceso

Métodos de evaluación del proceso

#### 4. Mediciones del Producto y del Proceso

Mediciones del proceso de software

Mediciones del producto de software

- Mediciones de tamaño
- Mediciones de estructura
- Mediciones de calidad

Calidad de los resultados de la medición

Técnicas de medición
- Técnicas analíticas
- Técnicas de evaluación comparativa

## CALIDAD DE SOFTWARE

Con los años, autores y organizaciones han definido el término calidad de varias maneras. De forma reciente, la calidad se define como el grado con el que un conjunto de características inherentes cumplen con los requerimientos.

### Temas

#### 1. Fundamentos de Calidad de Software

Cultura y ética en ingeniería de software

Valor y costos de calidad

Modelos y características de calidad

- Calidad del proceso de software
- Calidad del producto de software

Mejora de calidad

Requerimientos de calidad de las aplicaciones

- Criticidad de los sistemas
- Dependencia
- Niveles de integridad de software

Caracterización de defectos

#### 2. Procesos de Gestión de Calidad de Software

Aseguramiento de la calidad de software

Técnicas de gestión de calidad de software

- Técnicas estáticas
- Técnicas de uso intensivo de personas
- Técnicas analíticas
- Técnicas dinámicas

Mediciones de calidad de software

#### 3. Verificación y Validación (V&V)

Definiciones de V&V

- V&V de sistema y V&V de software
- V&V independiente

Técnicas de V&V
-----------------

- Testing
-----------

- Demostraciones
------------------

- Trazabilidad
----------------

- Análisis
------------

- Inspecciones
----------------

- Revisiones de a pares
-------------------------

- Recorridas
--------------

- Auditorias
--------------

## **ASIGNATURAS SIN MATERIA ESPECÍFICA**

Esta materia sirve para incluir asignaturas que no se encuentran en ninguna de las materias descritas anteriormente.

## Apéndice E. Ejemplos de certificaciones internacionales en Ingeniería de Software

<b>Certificación</b>	<b>Institución</b>
PSP Certified Developer	Software Engineering Institute – CMU <a href="http://www.sei.cmu.edu/">http://www.sei.cmu.edu/</a>
TSP Certified Coach	Software Engineering Institute – CMU
Certified Instructor for Designing Products and Process Using Six-Sigma	Software Engineering Institute – CMU
Software Architect Professional	Software Engineering Institute – CMU
Certification on Testing Foundation Level	International Software Testing Qualifications Board <a href="http://istqb.org/">istqb.org/</a>
Test Manager	International Software Testing Qualifications Board
Test Analyst	International Software Testing Qualifications Board
Technical Test Analyst	International Software Testing Qualifications Board
Black Box Software Testing Foundations	Association for Software Testing <a href="http://www.associationforsoftwaretesting.org/">http://www.associationforsoftwaretesting.org/</a>
Black Box Software Testing – Bug Advocacy	Association for Software Testing
Software Quality Engineer	American Society for Quality <a href="http://asq.org/">asq.org/</a>
Six-Sigma Black Belt	American Society for Quality
Project Management Professional	Project Management Institute <a href="http://www.pmi.org">www.pmi.org</a>
Risk Management Professional	Project Management Institute
Certification as Professional in Requirements Engineering (foundations, advanced y expert)	International Requirements Engineering Board <a href="http://www.certified-re.de/">http://www.certified-re.de/</a>
Certified Software Development Professional	IEEE Computer Society <a href="http://www.computer.org/">www.computer.org/</a>
Certified Software Development Associate	IEEE Computer Society