

1. Datos Generales de la asignatura

| | |
|---------------------------------|---|
| Nombre de la asignatura: | Estándares de calidad y pruebas de software |
| Clave de la asignatura: | AID-2105 |
| SATCA¹: | 2-3-5 |
| Carrera: | Ingeniería en Tecnologías de la Información y Comunicaciones. |

2. Presentación

| |
|---|
| Caracterización de la asignatura |
| <p>La asignatura de Estándares de calidad y pruebas de software aporta al perfil del egresado las bases necesarias para aplicar estándares de calidad a los procesos de desarrollo y a sus productos de software mediante la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y pruebas del software que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, así como de elevar la productividad, tanto para la labor de desarrollo como para el control de la calidad del software.</p> <p>Elaborar cuadros comparativos que permitan al estudiante discernir sobre la mejor opción de sobre las ventajas y desventajas de estándares tanto de seguridad como de desarrollo de software.</p> <p>Elaborar mapeo de procesos tanto en redes como en sistemas de información con la finalidad de eficientar los procesos y modelos en una organización.</p> <p>Identificar los puntos importantes a considerar para obtener una certificación de estándares de seguridad y/o desarrollo de sistemas de información, así como determinar una tabla guía que permita ordenar prioridades en para acreditar una evaluación de estándares.</p> |
| Intención didáctica |
| <p>Se organiza el temario, en cinco unidades, con un nivel de abstracción creciente agrupando los contenidos conceptuales de la asignatura de manera que, el alumno empleará herramientas digitales para la creación de proyectos de diseños individuales y colaborativos, con animación y tecnología digital en la elaboración de nuevos productos y servicios.</p> |

¹ Sistema de Asignación y Transferencia de Créditos Académicos

3. Participantes en el diseño y seguimiento curricular del programa

| Lugar y fecha de elaboración o revisión | Participantes | Observaciones |
|---|--|--|
| TecNM/Instituto Tecnológico de Roque de 20/07/2020 a 20/08/2020 | Representantes de la academia de sistemas y computación. | Reunión Academia de sistemas y computación del TecNM -Roque. |

4. Competencia(s) a desarrollar

| Competencia(s) específica(s) de la asignatura |
|---|
| <p>Evaluar proyectos de software para asegurar su eficacia y eficiencia al ser liberados, migrados y entregados al usuario analizando y solucionando problemas informáticos y representando la solución mediante procesos y métodos orientados a determinar la calidad y el buen funcionamiento del software.</p> |

5. Competencias previas

| |
|--|
| <p>Competencias Instrumentales</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis • Capacidad de organizar y planificar • Comunicación oral y escrita • Solución de problemas Toma de decisiones <p>Competencias Interpersonales</p> <ul style="list-style-type: none"> • Capacidad de crítica y autocrítica • Trabajo en equipo <p>Competencias sistémicas</p> <ul style="list-style-type: none"> • Capacidad de aplicar conocimientos en la práctica • Habilidades de investigación • Habilidad para trabajar en forma autónoma • Capacidad de generar nuevas ideas • Búsqueda del logro |
|--|

6. Temario

| No. | Temas | Subtemas |
|-----|--|--|
| 1. | Introducción a los procesos de software | 1.1 Procesos. 1.1.1 Modelos de procesos. 1.2 Metodologías 1.2.1 Metodologías tradicionales. 1.2.2. Metodologías ágiles. 1.3 CMM. 1.3.1 Madurez VS Inmadurez. 1.4 CMMI. 1.5 Evolución de madurez de los procesos. 1.5.1 Descripción del modelo. 1.5.2 Representaciones. |
| 2. | Aseguramiento de la calidad del software | 2.1 Calidad del software. 2.1.1 Aseguramiento de la calidad. 2.2 Surgimiento de SQA (Software Quality Assurance). 2.2.1 Definición de SQA (Software Quality Assurance). 2.2.2 SQA no es lo mismo que SQC (Software Quality Control). 2.3 Funciones generales del SQA. 2.4 Consideraciones. 2.5 Aseguramiento de la calidad de procesos y productos en CMMI (PPQA). 2.6 Metas y prácticas. |
| 3. | Certificación CMMI - SW | 3.1 Requerimientos para la certificación de CMMI 3.2 Principios y conceptos fundamentales de ARC. (Appraisal Requirements for CMMI). 3.3 Tipos de indicadores de implementación de prácticas. 3.4 Estructura de los diferentes métodos de ARC (Appraisal Requirements for CMMI). 3.5 SCAMPI - Método Estándar de Evaluación de CMMI para Mejora de Procesos. 3.6 SCAMPI Leader Appraiser. |
| 4. | Estándares para la madurez de procesos. | 4.1 Modelado de procesos |

| | | |
|----|------------------------------------|---|
| | | <p>4.2 MOPRPOSFT</p> <p>4.2.1 Madurez vs Inmadurez</p> <p>4.2.2 Requerimientos para la certificación MOPROSOFT</p> <p>4.3 ISO 27000 y sus diferentes versiones</p> <p>4.3.1 Evolución de madurez en los procesos ISO 27000</p> <p>4.3.2 Buenas prácticas del ISO 27000</p> |
| 5. | Tipos de pruebas y Caso de estudio | <p>5.1 Pruebas de Software</p> <p>5.1.1 Pruebas estáticas.</p> <p>5.1.2 Pruebas dinámicas.</p> <p>5.1.3 Tipos de pruebas por su ejecución.</p> <p>5.2 Enfoques de pruebas.</p> <p>5.3 Niveles de pruebas.</p> <p>5.3.1 Pruebas funcionales.</p> <p>5.3.2 Pruebas no funcionales.</p> <p>5.4 Casos de Estudio</p> <p>5.4.1 Definición del dominio y el alcance.</p> <p>5.5 Aplicación de la herramienta al caso de estudio.</p> <p>5.5.1 Estados.</p> <p>5.5.2 Tipos de Proyectos.</p> <p>5.5.3 Tipos de Auditorías y sus tipos de tareas.</p> <p>5.5.4 Roles y Niveles de escalamiento de no conformidad.</p> |

7. Actividades de aprendizaje de los temas

| 1. Introducción a los procesos de software | |
|--|--|
| Competencias | Actividades de aprendizaje |
| <p>Específica(s): Aplicar un análisis para determinar el grado de madurez e inmadurez de un software.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis • Trabajo en equipo • Habilidades en el manejo de la computadora • Capacidad de crítica y autocrítica • Capacidad de aplicar los conocimientos en la práctica | <p>Desarrollar escenarios en clase para generar intercambio, discusiones, conclusiones y resultados.</p> <p>Emplear software para determinar el nivel de madurez.</p> <p>Representación de modelos</p> |
| 2. Aseguramiento de la calidad del software, | |
| Competencias | Actividades de aprendizaje |
| <p>Específica(s): Aplicar modelos SQA, CMMI Y PPQA para solución de problemas.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> •Capacidad de análisis y síntesis •Capacidad crítica y autocrítica •Habilidades en el manejo de la computadora •Trabajo en equipo •Capacidad de aplicar los conocimientos en la práctica •Capacidad de generar nuevas ideas (creatividad e innovación) | <p>Identificar las necesidades de una organización y evaluar la factibilidad de automatizar procesos.</p> <p>Desarrollar el plan de auditoria para evaluar la factibilidad de la implementación del CMMI.</p> <p>Implementar el SQA a manera de ejercicio en una organización virtual.</p> <p>Generar un modelo de resultado entre los sistemas.</p> |
| 3. Certificación CMMI - SW | |

| Competencias | Actividades de aprendizaje |
|--|--|
| <p>Específica(s):</p> <p>Identificar y utilizar de los diferentes modelos para la certificación mediante procesos CMMI-SW.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Habilidades en el manejo de la computadora • Habilidad para buscar y analizar información proveniente de fuentes diversas • Toma de decisiones. • Capacidad de aplicar los conocimientos en la práctica • Habilidades de investigación | <p>Elaborar un cuadro comparativo de los diferentes niveles del CMMI.</p> <p>Hacer un levantamiento de los requerimientos básicos para la implementación del CMMI-SW.</p> |
| <p>4. Estándares para la madurez de procesos.</p> | |
| Competencias | Actividades de aprendizaje |
| <p>Específica(s):</p> <p>Identificar y aplicar los lineamientos vigentes en la norma ISO 27000 y el estándar MORPOSOFIT.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis • Capacidad crítica y autocrítica • Habilidades en el manejo de la computadora • Trabajo en equipo • Capacidad de aplicar los conocimientos en la práctica • Capacidad de generar nuevas ideas (creatividad e innovación) | <p>Realizar investigación documental para conocer los conceptos de la norma ISO 27000 y sus diferentes versiones</p> <p>Investigar los conceptos clave para el uso de la norma MOPROSOFIT</p> <p>Crear una mesa de debate para dar a conocer los puntos de vista de los estudiantes acerca de puntos estratégicos de las normas.</p> <p>Empatar los conceptos teóricos con su aplicación para un proyecto del área de redes.</p> |

| 5. Tipos de pruebas y casos de estudio | |
|--|---|
| Competencias | Actividades de aprendizaje |
| <p>Específica(s):</p> <p>Aplicar casos de uso y diferentes tipos de pruebas dependiendo del escenario que se presente.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de abstracción, análisis y síntesis. • Capacidad de aplicar los conocimientos en la práctica. • Conocimiento sobre el área de estudio y la profesión. • Habilidades en el uso de las Tic's • Capacidad creativa. • Habilidad para buscar y analizar información proveniente de fuentes diversas • Toma de decisiones. | <p>Identificar los puntos a los que aplica la norma CMMI-SW para la elaboración de casos de uso.</p> <p>Identificar los puntos de la norma CMMI-SW en los que aplican las diferentes pruebas para garantizar el funcionamiento de los sistemas.</p> |

8. Práctica(s)

Elaboración de un programa de diagnóstico para el área de TI considerando:

1. Glosario.
2. Elaborar mapa conceptual para motivar al equipo de evaluación de la calidad del software.
3. Determinar un problema para desarrollar proyecto final a realizar.
4. Investigación de los procesos CMMI.
5. Representación de los proyectos con CMMI.
6. Presentación de la evaluación de las pruebas de calidad por equipo.

7. Determinar la madures de cada proyecto con CMMI.
8. Hacer pruebas de mejora.
9. Determinar y corregir las no conformidades.
10. Delimitar el grado de escalonamiento.

9. Proyecto de asignatura

La asignatura tendrá por objetivo el comprender y aplicar - en lo posible - los conceptos y técnicas adecuados para asegurar que el software cumple con sus especificaciones y satisface las expectativas de los usuarios.

Para esto se cubrirán las técnicas de testing y análisis clasificadas como Caja Blanca y Caja Negra, tanto para software tradicional como para Orientación a Objetos, y para los nuevos paradigmas basado en Componentes Software y orientado a Servicios o basado en la web (Servicios Web).

10. Evaluación por competencias

La evaluación debe ser continua y cotidiana por lo que se debe considerar el desempeño en cada una de las actividades de aprendizaje, haciendo especial énfasis en:

- Determinar un proyecto inicial por equipo.
- Elegir una metodología para el desarrollo, evaluación y pruebas del software.
- Generar reportes de diferentes fuentes de información.
- Desarrollar planes de evaluación y aplicación del modelo CMMI.
- Estructurar planes de acción en caso de que el modelo tenga áreas de oportunidad
- Reportar pruebas en el proceso de CMMI.
- Implementar soluciones a problemas planteados.
- Integrar los productos obtenidos de cada unidad al proyecto final.

11. Fuentes de información

1. Castejón, Miriam. Software agentes seguros.
2. Rodríguez, Damián. Software para encuestas Ed. Mc Graw Hill.
3. Ureña López y fundamentos de la calidad de software. Herramientas de Autor. RA-MA EDITORIAL.
4. Pressmasn, Ingeniería de software Mc Graw Hill.
5. Vara Mesa, medición y estimación de software. Mc Graw Hill.
6. Díaz, E., J. Tuya, et al. (2003). Pruebas automáticas de cobertura de software mediante una herramienta basada en Búsqueda Tabú.
7. Myers, Glenford J. (1979). The Art of Software Testing, (USA): Wiley-Interscience Publications.