

1. DATOS DE LA ASIGNATURA

Nombre de la asignatura:	Verificación y Validación del Software.
Carrera:	Ingeniería en Sistemas Computacionales
Clave de la asignatura:	ISF-1810
Créditos:	3-2-5

2. PRESENTACIÓN.

Caracterización de la asignatura.
<p>Esta asignatura aporta al perfil del Ingeniero en Sistemas Computacionales la formación de personal con un dominio sobre el proceso de verificación y validación con un enfoque que promueve la calidad del software.</p> <p>La importancia de esta asignatura se centra en la realización de pruebas completas al software antes de ser liberado hacia el usuario final, esto debido a que se ha comprendido la importancia de las pruebas de calidad y su impacto en la industria del software.</p> <p>Los aspectos principales a considerar son las revisiones y las pruebas como parte del ciclo de vida del software que se utilizan para detectar fallas en las diferentes etapas del mismo. Se centra en la detección de defectos en el producto, dándole mucha importancia a las revisiones.</p> <p>La verificación y la validación del software incluyen un conjunto de procedimientos, actividades, técnicas y herramientas, que se utilizan paralelamente al desarrollo del mismo para asegurar que el producto cubre los diferentes aspectos de calidad. El objetivo es prevenir las fallas desde los requerimientos hasta su implementación.</p>
Intención didáctica.
<p>La asignatura es teórica – práctica y permite desarrollar en el estudiante la habilidad para la aplicación de las diferentes técnicas de verificación y validación de software, considerando los principios de la ingeniería de software, para lo cual el temario se organiza en cinco bloques.</p> <p>En el bloque uno, introducción a la verificación y validación del software, se retoma de manera importante la calidad del software como fundamento para conocer y comprender el proceso de verificación y validación del software, así como la identificación de los estándares de la IEEE que le son aplicados.</p> <p>El bloque dos apoyará en la definición de las revisiones al software, así como el proceso de pruebas de software.</p>

En el bloque tres se detalla el proceso de verificación del software.

En el bloque cuatro se hace una relación entre el modelado y el proceso de verificación.

En el último bloque, se plantea el proceso a desarrollar en el proceso de validación del software por parte del cliente.

3. COMPETENCIAS A DESARROLLAR

Competencias específicas:	Competencias genéricas:
<p>Manejar técnicas para verificaciones y validaciones al software, considerando los aspectos de revisión y pruebas como parte del ciclo de vida para detectar fallas en el mismo.</p>	<p>Competencias instrumentales</p> <ul style="list-style-type: none">• Capacidad de análisis y síntesis.• Capacidad de organizar y planificar.• Comunicación oral y escrita.• Habilidad para buscar y analizar información proveniente de fuentes diversas.• Solución de problemas.• Toma de decisiones.• Conocimiento de una segunda lengua. <p>Competencias interpersonales</p> <ul style="list-style-type: none">• Capacidad crítica y autocrítica.• Trabajo en equipo.• Habilidades interpersonales.• Capacidad de comunicarse con profesionales de otras áreas.• Compromiso ético. <p>Competencias sistémicas</p> <ul style="list-style-type: none">• Capacidad de aplicar conocimientos en la práctica.• Habilidades de investigación.• Capacidad de aprender.• Capacidad de generar nuevas ideas.• Capacidad de diseñar y gestionar proyectos.• Preocupación por la calidad.• Búsqueda del logro.

4. HISTORIA DEL PROGRAMA

Lugar y fecha de elaboración o revisión	Participantes	Observaciones (cambios y justificación)
Dirección General de Institutos Descentralizados, México D.F. Fecha: 12 al 14 de septiembre de 2012.	Representantes: Tecnológico de Estudios Superiores de Coacalco. Instituto Tecnológico Superior de Atlixco, Comalcalco, Fresnillo, Santiago Papasquiaro, Tepexi de Rodríguez, Zapopan.	Análisis y adecuación por competencias del módulo de la especialidad "Ingeniería de Software" de la carrera de Ingeniería en Sistemas Computacionales.

5. OBJETIVO(S) GENERAL(ES) DEL CURSO (competencias específicas a desarrollar en el curso).

Aplicar técnicas para verificaciones y validaciones al software, considerando los aspectos de revisión y pruebas como parte del ciclo de vida para detectar fallas en el desarrollo del mismo.

6. COMPETENCIAS PREVIAS.

- Identifica el proceso de ingeniería de software
- Conoce el ciclo de vida
- Utilizar herramientas de modelado
- Capacidad de análisis
- Capacidad de crítica constructiva

6. TEMARIO.

Unidad	Temas	Subtemas
1	Introducción al proceso de verificación y validación.	1.1 Contextualización de la verificación y validación. 1.2 Terminología del proceso. 1.3 El proceso de la verificación y validación. 1.4 Tipos generales de los errores. 1.5 Responsabilidad de pruebas. 1.6 Organigrama de proceso de testing (un modelo propuesto). 1.7 Costos del error.
2	Pruebas.	2.1 Tipos de pruebas. 2.2 Cobertura de las pruebas. 2.3 Preparación de la prueba. 2.4 Productos de la prueba. 2.5 Criterios para la realización de pruebas. 2.6 Plan Pruebas (validación y verificación). 2.7 Estructura de los casos de Prueba. 2.8 Conceptos Generales, los diseños de las pruebas (validación y verificación). 2.9 Reporte y Seguimiento de errores. 2.10 Informe de la Prueba. 2.11 Fuentes de información de QA para el control estadística o métricas. 2.12 Control estadístico vs métricas. 2.13 Importancia de la calidad, las métricas y el control estadístico.
3	Verificación.	3.1 Marco de Referencia para el desarrollo de software. 3.2 Herramientas para apoyar al proceso y la ejecución de las revisiones de software. 3.3 Manejo de Requerimientos (Verificación). 3.4 Verificación en este proceso. 3.5 Entradas propuestas para el proceso de verificación de requerimientos. 3.6 Método de verificación. 3.7 Aspectos a verificar en esta etapa. 3.8 Entendimiento de problema (Verificación). 3.9 Revisión general de requerimientos. 3.10 Fase de manejo de requerimientos.
4	Modelado.	4.1 Modelado de pruebas con UML. 4.2 Cumplimiento de la especificación en los requerimientos. 4.3 Importancia en la efectividad en el diseño. 4.4 Patrones (tipos de patrones, como utilizar los patrones para validar). 4.5 Contratación y Outsourcing.

5	Implementación.	5.1 Implementación. 5.2 Entradas para pruebas. 5.3 Plan de pruebas (estrategia de prueba, ambientes, test team, atacar y asegurar regresión). 5.4 Ejecución de tipos generales de pruebas. 5.5 Caja negra y caja blanca. 5.6 Otros tipos de test. 5.7 GUI, Funcionalidad, Performance, entre otros. 5.7.1 Documentación (técnica y de usuario). 5.7.2 Seguridad. 5.7.3 Diseño de las pruebas.
6	Validación y logística de pruebas.	6.1 Pruebas y aceptación del cliente 6.2 Entrega de proceso de pruebas. 6.3 Formalización y cierre del proyecto. 6.4 Monitoreo y seguimiento del proyecto. 6.5 Formalización de cambios. 6.6 Administración de defectos.

7. SUGERENCIAS DIDÁCTICAS (desarrollo de competencias genéricas).

El profesor debe:

- Ser conocedor de la disciplina que está bajo su responsabilidad, conocer su origen y situación actual para considerarlo al abordar los temas.
- Desarrollar la capacidad para coordinar y trabajar en equipo; orientar el trabajo del estudiante y desarrollar en él la autonomía, el trabajo cooperativo y la toma de decisiones. Mostrar flexibilidad en el seguimiento del proceso formativo y propiciar la interacción entre los estudiantes. Tomar en cuenta el conocimiento de los estudiantes como punto de partida y como obstáculo para la construcción de nuevos conocimientos.
- Propiciar actividades de metacognición. Ante la ejecución de una actividad, señalar o identificar el tipo de proceso intelectual que se realizó: una identificación de patrones, un análisis, una síntesis, la creación de una heurística, etc. Al principio lo hará el profesor, luego será el alumno quien lo identifique.
- Propiciar actividades de búsqueda, selección y análisis de información en distintas fuentes.
- Fomentar actividades grupales que propicien la comunicación, el intercambio argumentado de ideas, la reflexión, la integración y la colaboración entre los estudiantes.
- Observar y analizar fenómenos y problemáticas propias del campo ocupacional.
- Relacionar los contenidos de esta asignatura con las demás del plan de estudios a las que ésta da soporte para desarrollar una visión interdisciplinaria en el estudiante.
- Propiciar el desarrollo de capacidades intelectuales relacionadas con la lectura, la escritura y la expresión oral.
- Facilitar el contacto directo con herramientas para contribuir a la formación de las competencias para el trabajo experimental como: identificación, manejo y control de variables y datos relevantes, planteamiento de hipótesis, trabajo en equipo.
- Propiciar el desarrollo de actividades intelectuales de inducción-deducción y análisis síntesis, que lo encaminen hacia la investigación.

- Desarrollar actividades de aprendizaje que propicien la aplicación de los conceptos, modelos y metodologías que se van aprendiendo en el desarrollo de la asignatura.
- Proponer problemas que permitan al estudiante la integración de contenidos de la asignatura y entre distintas asignaturas, para su análisis y solución.
- Cuando los temas lo requieran, utilizar medios audiovisuales para una mejor comprensión del estudiante.
- Propiciar el uso de las nuevas tecnologías en el desarrollo de la asignatura.

8. SUGERENCIAS DE EVALUACIÓN.

La evaluación de la asignatura debe ser continua y formativa por lo que se hará con base en el siguiente desempeño para cada una de las actividades:

- Solución de casos prácticos solicitados durante las actividades, así como sus conclusiones de forma escrita.
- Reportes de investigación.
- Reportes de prácticas.
- Ejercicios realizados.
- Tareas.
- Exposición.
- Participación en clase.
- Proyecto integral.
- Exámenes teóricos y/o prácticos.

9. UNIDADES DE APRENDIZAJE.

Unidad 1: Introducción al proceso de la verificación y validación.

Competencia específica a desarrollar	Actividades de Aprendizaje
Identifica el proceso de verificación y validación del software.	<ul style="list-style-type: none"> • Investigar los conceptos de calidad del software. • Analiza el proceso de verificación y validación del software. • Desarrollar un glosario de términos • Integración de la verificación y validación al proceso de desarrollo de software. • Conocer el papel de los estándares en el proceso de verificación y validación del software.

Unidad 2: Pruebas.

Competencia específica a desarrollar	Actividades de Aprendizaje
Identifica las características y los tipos de pruebas que se le aplican al software.	<ul style="list-style-type: none"> • Investigar la definición y características de las revisiones de software. • Diseñar un caso de prueba • Realizar un reporte de pruebas y seguimiento de errores. • Utilizar formatos propuestos para la documentación de las pruebas de software.

Unidad 3: Verificación.

Competencia específica a desarrollar	Actividades de Aprendizaje
Analiza el proceso de verificación al software.	<ul style="list-style-type: none"> • Investigar el proceso de la verificación. • Realizar en un caso práctico el manejo de requerimientos por medio de un método de verificación.

Unidad 4: Modelado.

Competencia específica a desarrollar	Actividades de Aprendizaje
Utiliza herramientas para el modelado en el proceso de verificación y validación del software.	<ul style="list-style-type: none"> • Investigar el uso de herramientas para apoyar al proceso y la ejecución de las revisiones de software. • Investigar el uso de herramientas para apoyar al proceso, el diseño y la ejecución de las pruebas de software. • Modelar casos prácticos de pruebas.

Unidad 5: Implementación.

Competencia específica a desarrollar	Actividades de Aprendizaje
Planea el proceso de verificación y validación del software, y aplicar métricas para evaluar los resultados finales.	<ul style="list-style-type: none"> • Investiga el proceso de planeación. • Ejecuta pruebas y analiza los resultados.

Unidad 6: Validación y logística de pruebas.

Competencia específica a desarrollar	Actividades de Aprendizaje
<p>Analizar el proceso de verificación y validación del software, y aplicar métricas para evaluar los resultados finales.</p>	<ul style="list-style-type: none"> • Analiza e interpreta los resultados de las pruebas. • Elabora carta de aceptación del cliente. • Documenta el seguimiento a cambios del proyecto.

10. FUENTES DE INFORMACIÓN.

1. Test Process Improvement, Tim Koomen and Martin Pol. Addison-Wesley. 1999
2. Software Testing and Continuous Quality Improvement, Willam E. Lewis. Aurebach. 2000
3. Effective Methods for Software Testing (second edition), William E. Perry. WileyComputer Publishing. 2000
4. Software Verification and Validation, Steven R. Rakitin. Artech House.1997
5. UML y Patrones Craig Larman. Pearson. 1999
6. Software Testing and Quality Assurance Theory and Practice, Kshirasagar Naik & Priyadarshi Tripathy, Wiley.
7. Verification and Validation in Systems Engineering, Assessing UML/SysML Design Models, Mourad Debbabi, Fawzi Hassaine - Yosr Jarraya, Springer.

11. PRÁCTICAS PROPUESTAS.

1. Seleccionar un caso práctico de desarrollo de software para someterlo al proceso de verificación y validación del software.
2. Utilizar las herramientas para apoyar al proceso de verificación y validación del software.
3. Diseñar un plan de verificación y validación del software, así como aplicar las métricas para evaluar los resultados arrojados por la verificación y validación del software.
4. Diseñar un formato de pruebas.
5. Realizar visitas a empresas dedicadas al desarrollo de software.
6. Desarrollar una mesa de debates, en la cual se cuente con la presencia de un experto en desarrollo, a fin de conocer las experiencias con respecto a los procesos de verificación y validación del software.