

1. Datos Generales de la asignatura

Nombre de la asignatura:	Verificación y Validación del Software
Clave de la asignatura:	AEC-2106
SATCA¹:	2 – 2 – 4
Carrera:	Ingeniería en Sistemas Computacionales

2. Presentación

Caracterización de la asignatura

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.

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.

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.

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.

Intención didáctica

¹ Sistema de Asignación y Transferencia de Créditos Académicos

La asignatura es teórico – 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.

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.

El bloque dos apoyará en la definición de las revisiones al software así como el proceso de pruebas de software.

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

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

En el bloque cinco, se aborda el proceso de implementación y las diferentes herramientas que se pueden utilizar en el proceso.

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

3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
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.

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
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.

5. Competencias previas

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

6. Temario


No	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 diseño 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.

		<p>3. 5 Entradas propuestas para el proceso de verificación de requerimientos.</p> <p>3. 6 Método de verificación.</p> <p>3. 7 Aspectos a verificar en esta etapa.</p> <p>3. 8 Entendimiento de problema (Verificación).</p> <p>3. 9 Revisión general de requerimientos.</p> <p>3. 10 Fase de manejo de requerimientos.</p>
4.	Modelado.	<p>4. 1 Modelado de pruebas con UML.</p> <p>4. 2 Cumplimiento de la especificación en los requerimientos.</p> <p>4. 3 Importancia en la efectividad en el diseño.</p> <p>4. 4 Patrones (tipos de patrones, como utilizar los patrones para validar).</p> <p>4. 5 Contratación y Outsourcing.</p>
5.	Implementación.	<p>5. 1 Implementación.</p> <p>5. 2 Entradas para pruebas.</p> <p>5. 3 Plan de pruebas (estrategia de prueba, ambientes, test team, atacar y asegurar regresión).</p> <p>5. 4 Ejecución de tipos generales de pruebas.</p> <p>5. 5 Caja negra y caja blanca.</p> <p>5. 6 Otros tipos de test.</p> <p>5. 7 GUI, Funcionalidad, Performance, entre otros.</p> <p>5. 8 Documentación (técnica y de usuario).</p> <p>5. 9 Seguridad.</p> <p>5. 10 Diseño de las pruebas.</p>
6.	Validación y logística de pruebas.	<p>6.1 Pruebas y aceptación del cliente</p> <p>6.2 Entrega de proceso de pruebas.</p> <p>6.3 Formalización y cierre del proyecto.</p> <p>6.4 Monitoreo y seguimiento del proyecto.</p> <p>6.5 Formalización de cambios.</p> <p>6.6 Administración de defectos.</p>

7. Actividades de aprendizaje de los temas

Unidad 1: Introducción al proceso de la verificación y validación.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <p>Identifica el proceso de verificación y validación del software.</p>	<ul style="list-style-type: none"> ● Investigar los conceptos de calidad del software. ● Analiza el proceso de verificación y validación del software.

<p>Genéricas:</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. ● Habilidades de investigación. 	<ul style="list-style-type: none"> ● 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.
<p>Unidad 2: Pruebas.</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica(s):</p> <p>Identifica las características y los tipos de pruebas que se le aplican al software.</p> <p>Genéricas:</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. ● Habilidades de investigación. ● Capacidad crítica y autocrítica. ● Trabajo en equipo. ● Habilidades interpersonales. ● Capacidad de comunicarse con profesionales de otras áreas. ● Compromiso ético. ● Capacidad de generar nuevas ideas. 	<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.
<p>Unidad 3: Verificación.</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>

<p>Específica(s):</p> <p>Analiza el proceso de verificación al software.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> ● Capacidad de organizar y planificar. ● Solución de problemas. ● Toma de decisiones. ● Habilidades de investigación. ● Capacidad crítica y autocrítica. ● Trabajo en equipo. ● Habilidades interpersonales. ● Capacidad de comunicarse con profesionales de otras áreas. ● Compromiso ético. ● Capacidad de generar nuevas ideas. 	<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.
<p>Unidad 4: Modelado.</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica(s):</p> <p>Utiliza herramientas para el modelado en el proceso de verificación y validación del software.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> ● Capacidad de organizar y planificar. ● Solución de problemas. ● Toma de decisiones. ● Habilidades de investigación. ● Capacidad crítica y autocrítica. ● Trabajo en equipo. ● Habilidades interpersonales. ●  Compromiso ético. 	<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.
<p>Unidad 5: Implementación.</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>

<p>Específica(s):</p> <p>Planea el proceso de verificación y validación del software, y aplicar métricas para evaluar los resultados finales.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> ● Capacidad de organizar y planificar. ● Solución de problemas. ● Habilidades de investigación. ● Capacidad crítica y autocrítica. ● Trabajo en equipo. ● Habilidades interpersonales. ● Capacidad de comunicarse con profesionales de otras áreas. ● Compromiso ético. ● Capacidad de generar nuevas ideas. 	<ul style="list-style-type: none"> ■ Investiga el proceso de planeación. ■ Ejecuta pruebas y analiza los resultados
<p>Unidad 6: Validación y logística de pruebas.</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica(s):</p> <p>Analizar el proceso de verificación y validación del software, y aplicar métricas para evaluar los resultados finales.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> ● Capacidad de aplicar los conocimientos en la práctica. ● Habilidades de investigación. ● Capacidad de aprender. ● Capacidad de generar nuevas ideas. 	<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.

8. Práctica(s)

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.

9. Proyecto de asignatura

Dar continuidad a algún proyecto iniciado en materias previas de la especialidad o realizar un caso práctico de desarrollo de software de acuerdo a lo siguiente:

- Descripción del proyecto (si es un proyecto nuevo).
- Realizar el ciclo de vida del desarrollo de software.
- Aplicar las herramientas para el proceso de verificación y validación del software.
- Implementar métricas para la evaluación de resultados de la verificación y validación del software.

10. Evaluación por competencias

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.

11. 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.