

1. Datos Generales de la asignatura

Nombre de la asignatura:	Metodologías Ágiles de Desarrollo de Software
Clave de la asignatura:	PMD-2402
SATCA¹:	2-3-5
Carrera:	Ingeniería en Sistemas Computacionales

2. Presentación

Caracterización de la asignatura

Esta asignatura aporta al perfil del Ingeniero en Sistemas Computacionales la formación de profesionistas con la capacidad de desarrollar, implementar y administrar software de sistemas o de aplicación que cumpla con los estándares de calidad con el fin de apoyar la productividad y competitividad de las organizaciones. La capacidad de coordinar y participar en proyectos interdisciplinarios, ya que en los diferentes proyectos en los que participará el alumno en el ámbito laboral, hará trabajo en equipo con profesionales de diferentes áreas, así como personas involucradas en el proyecto en cuestión, aplicando la metodología Ágil. Además, esta asignatura también capacita a los ingenieros para dominar el proceso de verificación y validación, enfocándose en mejorar la calidad del software. Su importancia radica en realizar pruebas exhaustivas antes del lanzamiento para comprender su impacto en la industria. Las revisiones y pruebas en el ciclo de vida del software son cruciales para detectar defectos, destacando la importancia de las revisiones en este proceso. La verificación y validación emplean diversas herramientas y técnicas simultáneamente al desarrollo para garantizar la calidad desde los requisitos hasta la implementación, con el objetivo de prevenir fallos en el producto final.

La importancia de esta asignatura se basa en la aplicación de metodologías ágiles para desarrollar sistemas, productos o servicios con mayor calidad en el menor tiempo posibles y con costos más justos. La filosofía de estas metodologías ágiles, buscan la satisfacción del cliente y la entrega temprana de software de manera incremental; equipos de proyecto pequeños y con alta motivación; métodos informales; un mínimo de productos de trabajo de la ingeniería del software; y una simplicidad general del desarrollo de un sistema.

Esta asignatura tiene relevancia en la formación del ingeniero en sistemas, proporciona al alumno los conceptos básicos y las metodologías vigentes en el ámbito laboral, mismas que son de apoyo en la gestión o administración del proceso de desarrollo de proyectos de software. La aplicación de los conocimientos teóricos en la práctica de esta asignatura, conduce al alumno a la adquisición de competencias necesarias para llevar satisfactoriamente un proyecto real.

Intención didáctica

La asignatura es teórico – práctica y permite desarrollar en el estudiante la habilidad de la aplicación de las metodologías ágiles y la verificación y validación del software.

En el bloque uno se analizará el concepto de calidad, y las normas ISO aplicables al desarrollo de software, además de hacer la introducción al concepto de metodología ágil.

En el bloque dos, se conocerán las metodologías ágiles actuales más utilizadas en la industria del desarrollo del software y su aplicación e importancia.

En el bloque tres, se verá la metodología SCRUM conociendo sus definiciones y su forma de operación en el proceso del desarrollo de software.

¹ Sistema de Asignación y Transferencia de Créditos Académicos

En el bloque cuatro, Introducción a la verificación y validación del software, revisiones y pruebas de software.

En el bloque 5, Proceso de verificación del software, relación entre modelado y verificación, proceso de implementación y herramientas y proceso de validación del software desde la perspectiva del cliente.

3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Tecnológico Nacional de México Campus Ciudad Hidalgo, enero 2024	Representantes: Miembros de la academia de Ingeniería en Sistemas Computacionales	Análisis y definición por competencias del módulo de la especialidad de "Programación de Móviles" de la carrera de Ingeniería en Sistemas Computacionales.

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
Aplicar la metodología Ágil en el desarrollo de sistemas de software, aplicando métricas de calidad y estándares internacionales para garantizar entregas en tiempo, de igual forma el 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.

5. Competencias previas

<ul style="list-style-type: none"> • Aplica las actividades que involucran cada una de las etapas del ciclo de vida de un proyecto de software. • Identifica y Analiza necesidades de información para su representación, tratamiento y automatización para la toma de decisiones. • Identifica y selecciona estándares y métricas de calidad para ser aplicados a un proyecto de software. • 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 a la Metodología Ágil y Estándares.	1.1 Antecedentes del desarrollo del Software. 1.2 Calidad. 1.3 Mejora de Procesos. 1.4 Estándares ISO (ISO-9000, ISO-12207, ISO-15504 e ISO-14000). 1.5 Estándares CMMI (SW-CMM Y CMMI).



		<p>1.6 Proceso personal del Software. 1.7 Enfoque de la Metodología Ágil.</p>
2	Tipos de Metodologías Ágil utilizadas en la Industria del desarrollo de SW.	<p>2.1 Definición de Metodología Ágil. 2.2 Ventajas de la Metodología Ágil. 2.3 Manifiesto y principios de la Metodología Ágil. 2.4 SCRUM. 2.5 Kanban. 2.6 Extreme Programming. 2.7 Lean Software Development. 2.8 Soft Skill.</p>
3	Metodología SCRUM.	<p>3.1 Roles. 3.1.1 Product owner. 3.1.2 ScrumMaster. 3.1.3 Team. 3.2 Reuniones. 3.2.1 Sprint planning(User Story). 3.2.2 Sprint review. 3.2.3 Sprint retrospective. 3.2.4 Daily scrum meeting. 3.3 Artefactos. 3.3.1 Product backlog. 3.3.2 Sprint backlog. 3.3.3 Burndown charts. 3.3.4 Tablón de tareas. 3.4 Testing Ágil. 3.4.1 Principios. 3.4.2 Herramientas.</p>
4	Proceso de verificación y validación utilizando pruebas.	<p>4.1 Contextualización de la verificación y Validación, su terminología y proceso. 4.2 Tipos generales de los errores, costos del error y seguimiento de errores. 4.3 Administración y formalización de cambios y defectos. 4.4 Responsabilidad de las pruebas y el organigrama del proceso de testing. 4.5 Tipos de pruebas. 4.6 Pruebas su cobertura, preparación, productos y criterios para la realización de pruebas. 4.7 Documentación de pruebas (plan de pruebas, diseño de casos de pruebas, reporte de pruebas e informe de pruebas). 4.8 Importancia de la calidad, las métricas y el control estadístico.</p>
5	Verificación, Modelado e implementación.	<p>5.1 Marco de referencia para el desarrollo de software con QA y herramientas para revisión y pruebas de software. 5.2 Manejo y revisión de requerimientos para su cumplimiento, métodos de</p>

		verificación, aspectos a verificar y entendimiento del problema. 5.3 Modelado de pruebas con UML y la importancia en la efectividad de su diseño. 5.4 Patrones (tipos de patrones y su uso). 5.5 Fases de la implementación de pruebas. 5.6 Seguridad. 5.7 Formalización y cierre del proyecto para la aceptación del cliente.
--	--	---

7. Actividades de aprendizaje de los temas

1. Introducción a la Metodología Ágil y Estándares.	
Competencias	Actividades de aprendizaje
Específica(s): <ul style="list-style-type: none"> Identifica el estado actual del Software y las normas de calidad aplicables al desarrollo de sistemas. Genéricas: <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. Habilidades de investigación. 	<ul style="list-style-type: none"> Investigar los conceptos de calidad del software. Indagar en la práctica de las disciplinas básicas de la ingeniería del software el uso de los modelos para la evaluación del software. Investigar casos prácticos del uso de estándares internacionales. Realizar mapas conceptuales de los modelos.
2. Tipos de Metodologías Ágil utilizadas en la Industria del desarrollo de SW.	
Competencias	Actividades de aprendizaje
Específica(s): <ul style="list-style-type: none"> Identifica los tipos de metodologías ágiles actuales utilizadas en la industria del Software. Genéricas: <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. 	<ul style="list-style-type: none"> Investigar la historia y el desarrollo de la metodología Ágil. Analizar casos de estudio con proyectos realizados con metodología Ágil. Realizar mapas conceptuales de algunas metodologías Ágiles. Investigar Softskills más solicitadas en el mercado laboral. Realizar un cuadro comparativo de las metodologías Ágiles más utilizadas en México.

<ul style="list-style-type: none"> • Compromiso ético. • Capacidad de generar nuevas ideas. 	
3. Metodología SCRUM.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> • Analiza y conoce la Metodología SCRUM. <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"> • Realizar un cuadro sinóptico con los roles definidos en SCRUM. • Investigar los principios de SCRUM. • Estudiar en un caso práctico el uso y la aplicación de la metodología SCRUM. • Investigar las herramientas más utilizadas con la metodología SCRUM.
4. Proceso de verificación y validación utilizando pruebas.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> • Identifica el proceso de verificación y validación, las características y los tipos de pruebas que se le aplican al 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. • 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 las características del proceso de verificación y validación, de las revisiones, de la planeación y de la calidad del software en dicho proceso. • Desarrollar un glosario de términos de los conceptos del proceso de verificación y validación de software. • Documenta el seguimiento a cambios del proyecto. • Diseñar un caso de prueba. • Ejecuta pruebas y analiza los resultados. • Realizar un reporte de pruebas y seguimiento de errores. • Utilizar formatos propuestos para la documentación de las pruebas de software. • Conocer el papel de los estándares en el proceso de verificación y validación del software.
5. Verificación, Modelado, implementación y validación.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p>	<ul style="list-style-type: none"> • Investiga el marco de referencia para

<ul style="list-style-type: none"> Analiza y planea el proceso de verificación y validación del software, utiliza herramientas para el modelado en este mismo proceso y aplica métricas para evaluar los resultados finales. <p>Genéricas:</p> <ul style="list-style-type: none"> Capacidad de organizar y planificar. Solución de problemas. Toma de decisiones. Capacidad de aplicar los conocimientos en la práctica. Habilidades de investigación. Capacidad de aprender. Capacidad crítica y autocrítica. Trabajo en equipo. Habilidades interpersonales. Capacidad de comunicarse con profesionales de otras áreas. Compromiso ético. Capacidad de generar nuevas ideas. 	<p>el desarrollo de software con QA y el uso de herramientas para apoyar al proceso, diseño y la ejecución de las revisiones y pruebas de software.</p> <ul style="list-style-type: none"> Analiza el proceso de verificación y validación del software para realizar la integración de este al desarrollo de software. Modelar casos prácticos de pruebas y realizar en un caso práctico el manejo de requerimientos por medio de un método de verificación. Analiza e interpreta los resultados de las pruebas. Elabora carta de aceptación del cliente.
---	--

8. Práctica(s)

1. Desarrollar un procedimiento para un proceso específico en el desarrollo de Software utilizando la norma ISO-9001.
2. Aplicación de test para conocer y aplicar las Softskills de los estudiantes.
3. Análisis de caso práctico de uso de SCRUM en un proyecto de desarrollo de SW.
4. Debate de la importancia de cada ROL en la metodología SCRUM, haciendo equipos aleatorios y simulando un problema real.
5. Seleccionar un caso práctico de desarrollo de software para someterlo al proceso de verificación y validación del software.
6. Utilizar las herramientas para apoyar al proceso de verificación y validación del software.
7. 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.
8. Diseñar un formato de pruebas.
9. Realizar visitas a empresas dedicadas al desarrollo de software.
10. 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 la metodología SCRUM en el desarrollo.
- 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. Beth Chrissis, Mary, Konrad, Mike, Shrum, Sandy, CMMI® Guidelines for Process Integration and Product Improvement,; Addison Wesley.
2. Humphrey, Watts S., (2001), Introducción al Proceso Software Personal (PSP), Addison-Wesley.
3. Kendall, Kenneth E. y Kendall, Julie E., "Análisis y Diseño de Sistemas", Pearson. Octava Edición. 2011
4. Palacio, Juan y Ruata, Claudia., "Scrum Manager Gestión de Proyectos" . SafeCreative. 2011
5. Test Process Improvement, Tim Koomen and Martin Pol. Addison-Wesley. 1999
6. Software Testing and Continuous Quality Improvement, Willam E. Lewis. Aurebach. 2000
7. Effective Methods for Software Testing (second edition), William E. Perry. WileyComputer Publishing. 2000
8. Software Verification and Validation, Steven R. Rakitin. Artech House.1997
9. UML y Patrones Craig Larman. Pearson. 1999
10. Software Testing and Quality Assurance Theory and Practice, Kshirasagar Naik & Priyadarshi Tripathy, Wiley.
11. Verification and Validation in Systems Engineering, Assessing UML/SysML Design Models, Mourad Debbabi, Fawzi Hassaine - Yosr Jarraya, Springer.