



Instituto Tecnológico de Hermosillo.



MANUAL DE PRÁCTICAS DE LA MATERIA

Metodologías Ágiles de Desarrollo de Sistemas

Carrera: Ingeniería en Sistemas Computacionales.

Elaborado por
Maestra Adscrita al Departamento de Sistemas y
Computación

M.S.I. Francisca Lorena Zepeda Miramontes

Dictamen No. AS-2-116/2018 de Fecha 01-08-2018

OBJETIVO DE MANUAL:

Ofrecer los recursos necesarios, tanto a alumnos como docentes para desarrollar las competencias pertinentes en este tema tan importante hoy en día como lo son las Metodologías Ágiles, en especial Scrum.

Guiar a los estudiantes, sobre la base teórica y la aplicación adecuada del marco de trabajo en Scrum, desarrollando un proyecto de software a una empresa en específico en un problema planteado por ellos mismos.

INDICE

	Portada	1
	Objetivo del Manual	2
	Introducción	5
	Caracterización de la Materia	6
UNIDAD UNO	INTRODUCCIÓN A LOS PROCESOS ÁGILES	8
Práctica Numero 1 Resumen	Reafirmar los conceptos base de los procesos ágiles debiendo contener al menos 3 temas de la unidad.	14
Práctica Numero 2 Cuestionario	Demostrar el conocimiento sobre los conceptos base de los procesos ágiles vistos en la unidad.	15
Práctica Numero 3 Investigación sobre Problema Real	Investigar y presentar las necesidades de desarrollo de un software para una empresa en específico. (como inicio para proyecto integrador final)	16
Práctica Numero 4 Investigación SoftSkills	Investigar sobre la importancia de las SoftSkills (habilidades blandas) en una empresa	17
UNIDAD DOS	TIPOS DE METODOLOGÍAS, SUS PRINCIPIOS Y CARACTERÍSTICAS	18
Práctica Numero 5 Cuadro comparativo	Elaborar un cuadro comparativo de las diferentes metodologías ágiles vistas en clase.	29
Práctica Numero 6 Debate	Leer, comprender y discutir las ventajas de cada una de las metodologías ágiles.	30
Práctica Numero 7 Investigación uso de una metodología ágil	Conocer y presentar como es aplicada al menos una metodología ágil en una empresa en específico.	31
Práctica Numero 8 Dinámica “Toma la pelota y suéltala”	Fomentar el trabajo en equipo, la comunicación, planeación, toma de decisiones y gestión del tiempo	32
Práctica Numero 9 Cuestionario	Demostrar el conocimiento sobre las características de las diferentes metodologías ágiles.	34
UNIDAD TRES	METODOLOGÍA SCRUM	35
Práctica Numero 10 Dinámica “Organizando mi casa”	Identificar, comprender y aplicar la distribución de tareas a partir de una necesidad en específico. Fomentar el trabajo en equipo que adema apoye a identificar necesidades y habilidades de cada uno de los participantes para de esta manera distribuir de mejor manera el trabajo	46
Práctica Numero 11 Mapa conceptual	Identificar de manera clara todos los conceptos y elementos del framework de Scrum.	48
Práctica Numero 12 Reporte final y exposición	Presentar y evidenciar la aplicación del framework Scrum en el desarrollo de un proyecto integrador (software) a partir de la necesidad de una empresa analizada desde la primera unidad	49
Práctica Numero 13 Revisión de seguimiento al proyecto integrador	Evidenciar en cada sesión de seguimiento el avance del trabajo, demostrando cada integrante del equipo de trabajo su participación y compromiso en el mismo	50
Práctica Numero 14 Investigación sobre diferentes tablonas de tareas	Propiciar el conocimiento de las diferentes formas de implementar Scrum	51
Práctica Numero 15 Investigación sobre herramientas útiles para apoyo de trabajo en Scrum	Conocer diferentes herramientas electrónicas de apoyo a scrum como tablonas de tareas, control de versiones, planning póker, etc	52
Práctica Numero 16 Cuestionario	Demostrar el conocimiento sobre la Metodología Scrum	53
	BIBLIOGRAFÍA	54

TABLA DE ANEXOS

UNIDAD I. INTRODUCCIÓN A LOS PROCESOS ÁGILES	
ANEXO 1	56
ANEXO 2	58
ANEXO 3	67
ANEXO 4	69
UNIDAD II. TIPOS DE METODOLOGÍAS, SUS PRINCIPIOS Y CARACTERÍSTICAS	
ANEXO 5	71
ANEXO 6	74
ANEXO 7	79
ANEXO 8	81
ANEXO 9	87
UNIDAD III. METODOLOGÍAS ÁGILES	
ANEXO 10	89
ANEXO 11	92
ANEXO 12	94
ANEXO 13	96
ANEXO 14	98
ANEXO 15	100
ANEXO 16	102

Introducción.

“Las empresas que apuestan por una Transformación Digital completa, terminan por incluir, implantar y desarrollar **metodologías ágiles** en el interior de sus departamentos para entregar los productos y/o servicios con una mayor calidad y con unos costes y tiempos mucho más reducidos” Vanessa Rosselló Villán (octubre 2018).

La filosofía de estas metodologías, 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. Se concreta su filosofía en lo siguiente:

- **Participación del cliente:** Los clientes deben estar fuertemente implicados en todo el proceso de desarrollo. Su papel es proporcionar y priorizar nuevos requisitos del sistema y evaluar las iteraciones. Sin embargo, el éxito depende de que el cliente esté dispuesto y pueda pasar tiempo con el equipo de desarrollo.
- **Entrega incremental:** El software se desarrolla en incrementos o iteraciones, donde el cliente especifica los requisitos a incluir en cada incremento.
- **Personas, no procesos:** Se deben reconocer y explotar las habilidades del equipo de desarrollo. Se les debe dejar desarrollar sus propias formas de trabajar, sin procesos formales. Esto puede ser un problema si los miembros del equipo no tienen la personalidad apropiada y eso resulte en que no se relacionen adecuadamente con los otros miembros del equipo.
- **Aceptar el cambio:** Se debe contar con que los requisitos del sistema cambian, por lo que el sistema se desarrollará para dar cabida a esos cambios.

Por lo anterior y debido a la necesidad de las empresas instaladas en nuestra región, es que se decidió incluir esta materia en las especialidades para la carrera de Ingeniería en Sistemas Computacionales por lo que en este escrito se presenta el desarrollo de prácticas en la asignatura Metodologías Ágiles de Desarrollo de Software.

Principalmente se enfoca en llevar a cabo un proyecto de Desarrollo de Software dónde se apliquen principios de agilidad específicamente mediante la Metodología Scrum.

En este manual, veremos la manera en cómo se lleva a cabo la práctica, así como la forma de asegurar el aprendizaje y aplicación de los temas de esta Materia. Para ello se presentan 15 prácticas de diferente índole.

CARACTERIZACION DE LA MATERIA

Esta asignatura aporta al perfil del Ingeniero en Sistemas computacionales 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.

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. La asignatura de Metodologías Ágiles de Desarrollo de Software, se relaciona con las asignaturas previas de Ingeniería de Software, Desarrollo de Proyectos de Software I y II, Gestión de proyectos de software, Programación Web, Administración de B.D y Programación Móvil.

Competencias Genéricas:

- Capacidad de análisis y síntesis.
- Capacidad de organizar y planificar.
- Conocimientos básicos de la carrera.
- Comunicación oral y escrita en su propia lengua.
- Habilidades básicas de manejo de la computadora.
- Conocimiento de una segunda lengua.
- Habilidad para gestión de información.
- Habilidad para buscar y analizar información.
- Capacidad de trabajo en equipo.
- Capacidad de aplicar los conocimientos en la práctica.

Competencias previas:

- Aplica las actividades que involucran cada una de las etapas del ciclo de vida de un proyecto de software.
- Conoce procesos de la Ingeniería de software.
- Maneja alguna metodología para el desarrollo de software.
- Identifica y Analiza necesidades de información para su representación, tratamiento y automatización para la toma de decisiones.
- Diseña esquemas de bases de datos para generar soluciones al tratamiento de información.
- Conoce y comprende el entorno de la gestión de Proyectos.
- Identifica y selecciona estándares y métricas de calidad para ser aplicados a un proyecto de software.

UNIDAD 1.- INTRODUCCIÓN A LOS PROCESOS ÁGILES.

Competencia específica a desarrollar: Comprender los conceptos básicos de los procesos ágiles.

Competencias genéricas a adquirir:

- Capacidad de análisis y síntesis.
- Capacidad de organizar y planificar.
- Comunicación oral y escrita en su propia lengua.
- Habilidades básicas de manejo de la computadora.
- Habilidad para gestión de información.
- Habilidad para buscar y analizar información.

Subtemas.

- 1.1. Definición
- 1.2. Manifiesto por el desarrollo de software ágil
- 1.3. Principios de agilidad
- 1.4. Valores en el desarrollo ágil
- 1.5. Políticas del desarrollo ágil
- 1.6. Softskills

En la actualidad, las metodologías ágiles que se usan para el desarrollo de software están posicionándose muy fuertemente en las empresas líderes de este sector como, Microsoft, Oracle, SAP, IBM, CA y HP.

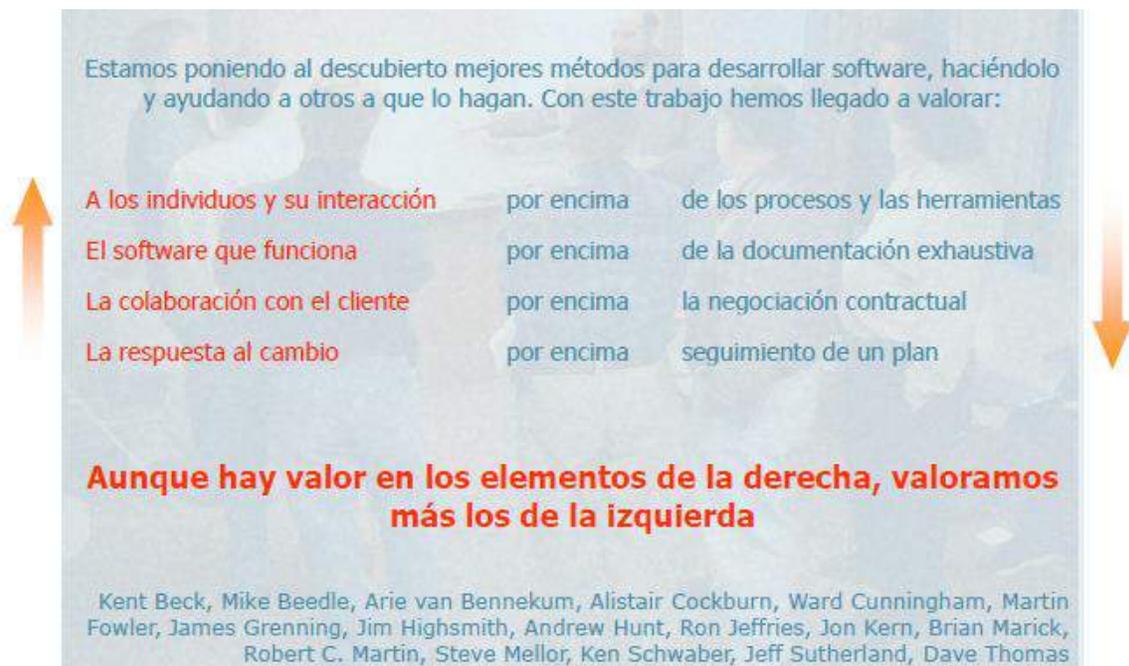
Las metodologías ágiles contemplan el desarrollo de software de manera integral, con un énfasis muy marcado en la entrega de valor al cliente y el retorno de la inversión (ROI).

La colaboración entre el cliente y los desarrolladores es vital e indispensable para la entrega de resultados, siempre se debería fomentar y apoyar.

Es muy importante promover técnicas, procesos y métodos que faciliten esta colaboración. Se debe permitir a los desarrolladores expandir su aportación de valor a los proyectos, y así ofrecer a los clientes transparencia sobre los mismos.

La filosofía de los procesos ágiles es buscar 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 de software; y una simplicidad general del desarrollo.

Esta filosofía se basa en el siguiente manifiesto:



Estamos poniendo al descubierto mejores métodos para desarrollar software, haciéndolo y ayudando a otros a que lo hagan. Con este trabajo hemos llegado a valorar:

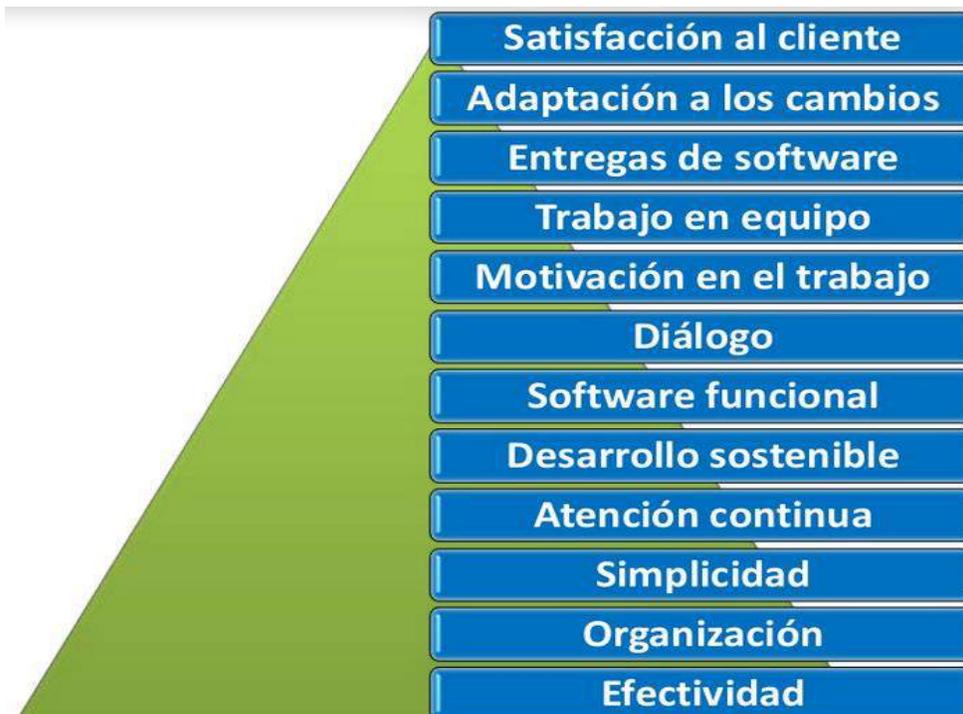
A los individuos y su interacción	por encima	de los procesos y las herramientas
El software que funciona	por encima	de la documentación exhaustiva
La colaboración con el cliente	por encima	la negociación contractual
La respuesta al cambio	por encima	seguimiento de un plan

Aunque hay valor en los elementos de la derecha, valoramos más los de la izquierda

Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas

Manifiesto Ágil presentado en Febrero de 2001.

Principios de Agilidad



Valores en el desarrollo ágil

- Compromiso
- Enfoque
- Apertura y transparencia
- Respeto
- Coraje

Políticas en el desarrollo ágil

Lineamientos Generales

- El proyecto deberá ser ejecutado en iteraciones incrementales con una demostración del producto al finalizar cada iteración (1-4 semanas, idealmente cada 2 semanas): con esta política, se conocerá el estado del proyecto, evaluando si los requisitos cumplen con las expectativas del cliente, si la calidad es la esperada, o si hay retrasos; agilizando la toma de decisiones correctivas.
- Los requisitos se desarrollarán priorizados por el valor aportado al cliente: Esta política permitirá que los objetivos más importantes del proyecto sean atendidos.

- El control y seguimiento del proyecto se basará en los requisitos completados en cada iteración. Se entiende como un requisito, los entregables asociados a: análisis, desarrollo, pruebas, documentación, etc. e integrados con los entregables de las iteraciones anteriores.
- Cada requisito debe ser independiente del resto de los requisitos, en la medida de lo posible.
- Cada requisito debe ser demostrable, permitiendo cómo comprobar con el cliente que el requisito está completado y que se cumplen sus expectativas.

Desarrollo

- Los componente de software, deberán ser desarrollados y liberados por partes, y no entregados al final del proyecto.
- El desarrollo de los componentes de software que conformaran la solución, deberán ser liberados en varias iteraciones.
- Cada iteración deberá producir software con calidad de producción, probado, integrado, y documentado (funcional, técnica).
- Cada iteración deberá cumplir con un subconjunto de requerimientos.
- Cada iteración deberá contemplar (análisis, diseño, implementación, Pruebas, documentación, etc.).

Pruebas

- Cada proyecto debe incorporar las prácticas de TDD (Test Driven Development).
- Cada uno de los entregables, deberá contener scripts de pruebas unitarias, integrales, funcionales, etc; mediante la utilización de frameworks como junit, dbunit, mockObjects, etc.

Documentación

- La documentación del proyectos, específicamente: manual de usuario, manual de operaciones, arquitectura de la solución, especificaciones, etc.; deberán ser entregables parciales para cada una de las iteraciones, es decir, la documentación no se liberara al final del proyecto, sino en entregables parciales.

Control de Calidad

Cada uno de los entregables, serán sometidos a un script de calidad, que ejecutara la organización, y no serán admitidos como productos del proyecto hasta alcanzar un nivel aceptable.

Control de Riesgos

Los riesgos serán identificados en la primera iteración, llevándose a cabo también una valoración inicial de la exposición al riesgo y planes de contingencia. En cada iteración se revisará y actualizará el documento “Lista de Riesgos”, añadiendo además la lista de riesgos más importantes actualizada por cada iteración.

Control de Artefactos

- Cada uno de los artefactos del proyecto, deberán ser mantenidos bajo un sistema de control de versiones.
- La organización dispondrá un sistema de control de versiones, que deberá ser actualizado por el cliente de forma remota.

Soft Skills

Las habilidades blandas, son todas aquellas capacidades que permiten a cualquier persona relacionarse y desempeñarse mejor en su trabajo, además de reaccionar y responder mejor en escenarios cambiantes.

Las “Soft Skills” son transversales a cualquier cargo y organización, complementarias a la formación tradicional de todo profesional, y se han convertido en uno de los aspectos más considerados a la hora de buscar trabajo.

Las Soft Skills se complementan con Hard Skills o habilidades duras (parte del Cociente Intelectual de una persona), que son las exigencias profesionales de un trabajo y muchas otras actividades.

“El desarrollo ágil se centra en los talentos y habilidades de los individuos, puesto que el proceso se ajusta a personas y equipos específicos”. Cockburn y Highsmith.

Los integrantes de un equipo ágil deben tener las siguientes soft Skills :

Competencia. En el contexto de un desarrollo ágil la competencia abarca un talento innato, habilidades específicas relacionadas con el software, y un conocimiento del proceso que el equipo haya elegido aplicar.

Enfoque común. Aunque los miembros del equipo ágil desempeñan tareas distintas y aportan habilidades al proyecto, todos deben enfocarse en una meta: entregar al cliente un incremento de trabajo de software dentro del tiempo establecido.

Los integrantes de un equipo ágil deben tener las siguientes soft Skills:

Colaboración. La ingeniería de software incluye evaluar, analizar y usar información que se comunica al equipo de software y construir información que ofrezca un valor comercial para el cliente. Estas tareas se cumplirán si los miembros del equipo colaboran, entre ellos, con el cliente y con sus gerentes.

Habilidad para la toma de decisiones. El equipo ágil debe tener autonomía, la autoridad para tomar decisiones en cuanto a cuestiones técnicas y del proyecto.

Capacidad de resolución de problemas confusos. En algunos casos, el equipo debe aceptar que el problema que está resolviendo hoy tal vez no sea el problema que debe resolverse mañana. Sin embargo, las lecciones aprendidas en cualquier actividad para la resolución de problemas pueden beneficiar al equipo en fases posteriores del proyecto.

Las soft skills más valoradas por los reclutadores son:

- Motivación
- Gestión del tiempo
- Negociación
- Presentaciones en público
- Cómo dirigir reuniones
- Interacción personal y comunicación
- Diseño de la propia estrategia empresarial
- Delegación
- Gestión del estrés
- Toma de decisiones
- Resolución de conflictos

Importancia

Los profesionales que se comunican de manera asertiva, trabajan en equipo, cuentan con iniciativa, flexibilidad y tolerancia a la frustración, no sólo aportan a mejorar el clima laboral sino también contribuyen en la productividad de la empresa.

PRACTICAS PROPUESTAS
UNIDAD I

Práctica Numero 1. Resumen

Competencias a desarrollar:

- Capacidad de análisis y síntesis.
- Comunicación oral y escrita en su propia lengua.
- Habilidades básicas de manejo de la computadora.
- Habilidad para gestión de información.

Introducción: Llevar a cabo el resumen de un tema permite a quien lo realiza repasar los conceptos con lo cual refuerza el conocimiento y entendimiento de cada uno de ellos.

Esta práctica permitirá al estudiante reafirmar los conceptos base de los procesos ágiles, además de servir al docente, como guía para llevar a cabo su propia instrumentación didáctica

Correlación con el o los temas y subtemas del programa de estudio vigente:
Todos los temas

Material y equipo: Equipo de cómputo

Metodología.

El estudiante realizará y presentará en la fecha indicada, un resumen de los temas vistos en clase de la unidad I; con ello reafirmará la comprensión de cada uno de los temas, los cuales son fundamentales para su aplicación en el desarrollo del proyecto integrador final de esta materia

Para la entrega y posterior evaluación de esta práctica se tendrá como base una lista de cotejo (**Anexo 1**)

Práctica Numero 2.- Cuestionario

Competencias a desarrollar:

- Capacidad de análisis y síntesis.
- Conocimientos básicos de la carrera.
- Comunicación oral y escrita en su propia lengua.
- Habilidades básicas de manejo de la computadora.

Introducción: En un cuestionario (examen) el estudiante deberá demostrar el conocimiento y entendimiento de los conceptos y temas vistos en la unidad I, lo cual junto con el resultado de la evaluación del resumen nos dará la calificación global de la unidad I. El examen es un método de aprendizaje que le permite al estudiante recordar los temas vistos partiendo principalmente de la comprensión de la teoría vista en clase.

Esta práctica permitirá al estudiante reafirmar los conceptos base de los procesos ágiles, además de servir al docente como guía para llevar a cabo su propia instrumentación didáctica

Correlación con el o los temas y subtemas del programa de estudio vigente:

Todos los temas

Material y equipo: Equipo de cómputo

Metodología.

El estudiante deberá presentarse en laboratorio de cómputo y contestar el cuestionario (examen) en la fecha y hora indicada, utilizando la plataforma moodle, para lo cual el docente otorgará clave de acceso al examen y de esta manera demostrar que conoce y comprende los conceptos y temas que integran esta unidad.

Considerando que el cuestionario es un examen, se presenta para apoyo del docente el desarrollo de uno con sus respuestas y el formato de retroalimentación (**Anexo 2**).

Práctica Numero 3. Investigación sobre problema real.

Competencias a desarrollar:

- Capacidad de análisis y síntesis.
- Capacidad de organizar y planificar.
- Conocimientos básicos de la carrera.
- Comunicación oral y escrita en su propia lengua.
- Habilidades básicas de manejo de la computadora.
- Habilidad para gestión de información.
- Habilidad para buscar y analizar información.
- Capacidad de trabajo en equipo.
- Capacidad de aplicar los conocimientos en la práctica

Introducción: A partir de esta unidad iniciamos con el proyecto integrador, en el cuál se aplicarán varios, si no todos, los conceptos y temas que veremos en esta materia. Se le pide a los estudiantes se integren en equipos, con la finalidad de que decidan qué problema de información quieren trabajar durante el semestre; para esto deberán identificar alguna necesidad en una empresa (donde se les facilite la información, mediante alguna persona conocida en la misma).

Esta práctica permitirá al estudiante conocer cómo funciona una empresa o parte de ella, investigando además, quienes intervienen en los procesos y de qué manera lo hacen; que herramientas y documentos utilizan para llevar a cabo sus actividades, etc. Será de utilidad también para el docente, como guía para llevar a cabo su propia instrumentación didáctica.

Correlación con el o los temas y subtemas del programa de estudio vigente:

Todos los temas

Material y equipo: Equipo de cómputo

Metodología.

El estudiante deberá investigar y seleccionar una empresa a la cual tenga acceso por medio de un familiar o conocido, de manera que pueda obtener información sobre algún problema del cual se tenga la intención o sea factible solucionar, mediante el uso de un software. Esta necesidad servirá al estudiante para aplicar los temas de esta materia. Para la presentación y posterior evaluación de esta práctica deberá tomarse como base una lista de cotejo (**Anexo 3**).

Práctica Numero 4. Investigación SoftSkills

Competencias Genéricas:

- Capacidad de análisis y síntesis.
- Conocimientos básicos de la carrera.
- Comunicación oral y escrita en su propia lengua.
- Habilidades básicas de manejo de la computadora.
- Conocimiento de una segunda lengua.
- Habilidad para gestión de información.
- Habilidad para buscar y analizar información.

Introducción: Conocer lo que son las habilidades blandas (softskills), que de forma general, las empresas esperan que su colaboradores tengan. El estudiante mediante este tema y práctica, deberá tener claro en qué aspectos de su personalidad deberá trabajar, para tratar de cumplir con lo que se espera de él en el ámbito de trabajo.

Correlación con el o los temas y subtemas del programa de estudio vigente:
Softskills

Material y equipo: Equipo de cómputo

Metodología.

“Las habilidades blandas o Soft Skills son todos los atributos o capacidades que le permiten a una persona desempeñarse en su trabajo de manera efectiva. Estas habilidades apuntan al lado emocional, interpersonal y a cómo se desenvuelve el personal en una empresa. En este plano entran temáticas como: trabajo en equipo, resolución de problemas, gestión efectiva del tiempo, gestión del cambio, manejo del stress, liderazgo, comunicación efectiva, entre otros.

Los conocimientos duros brindan soluciones a problemas técnicos, pero las crisis no solo se resuelven de manera técnica, sino más bien de forma adaptativa como las que brindan las Soft Skills.”

El estudiante investigará en la empresa que elija, cuáles son las habilidades blandas que se fomentan y trabajan en su empresa y porque son importantes para ellos. Para la presentación y evaluación de esta práctica deberá tomarse como base una lista de cotejo (**Anexo 4**).

UNIDAD 2.- TIPOS DE METODOLOGÍAS ÁGILES, SUS PRINCIPIOS Y CARACTERÍSTICAS

Competencia específica a desarrollar: Conocer y comprender las diferentes metodologías ágiles.

Competencias a adquirir:

- Capacidad de análisis y síntesis.
- Conocimientos básicos de la carrera.
- Comunicación oral y escrita en su propia lengua.
- Habilidades básicas de manejo de la computadora.
- Conocimiento de una segunda lengua.
- Habilidad para buscar y analizar información.

Subtemas.

- 2.1 Programación Extrema(XP)
- 2.2 Scrum
- 2.3 Kanban
- 2.4 Cristal
- 2.5 Proceso unificado ágil (PUA)
- 2.6 Método de desarrollo de sistemas dinámicos (MDSD)

- **PROGRAMACIÓN EXTREMA (XP):**

Según Kent Beck es un proceso ligero, de bajo riesgo, flexible, predecible, científico y divertido de desarrollar software”.

Es una metodología de desarrollo de software utilizada en la actualidad para proyectos de corto plazo, con un equipo de proyecto pequeño, rápida o extrema, utiliza un enfoque orientado a objetivos, cuya particularidad es tener como parte del equipo al usuario final como requisito para llegar al éxito.

Esta metodología consta de 4 actividades del marco de trabajo: Planeación, Diseño, Codificación y pruebas.

Las características más relevantes de la programación extrema son las siguientes:

- Refabricación: Se basa en la utilización repetitiva de código, para lo cual se establecen patrones permitiendo mayor flexibilidad al cambio.
- Programación en pares: Consiste en que dos desarrolladores trabajen para un proyecto en la misma estación de trabajo.
- Pruebas: la fase de prueba se compone de dos tipos, las pruebas de unidad y las pruebas de aceptación. Las **pruebas de unidad** se basa en las pruebas

realizadas a los principales procesos y las **pruebas de aceptación** son realizadas por los clientes y se enfoca en las características generales del sistema.

- Metodología creada a base de prueba y error.
- Surge considerando 4 valores que pueden mejorar cualquier proyecto de software: Simplicidad, Comunicación, Retroalimentación, Coraje.
- Expresada en forma de 12 prácticas (algunas ya existentes desde hace años), que se soportan las unas a las otras y conforman un conjunto completo.

Los 4 valores

- Simplicidad: XP propone el principio de hacer la cosa más simple que pueda funcionar, en relación al proceso y la codificación. Es mejor hacer algo simple hoy, que hacerlo más complicado hoy y probablemente nunca usarlo.
- Comunicación: Algunos problemas en los proyectos tienen su origen en que alguien no dijo algo a alguien más sobre algo importante en algún momento. XP hace casi imposible la falta de comunicación.
- Retroalimentación: retroalimentación concreta y frecuente del cliente, del equipo y de los usuarios finales da una mayor oportunidad de dirigir el esfuerzo.
- Coraje: se requiere coraje para confiar en que la retroalimentación durante el camino es mejor que tratar de adivinar todo con anticipación. Se requiere valor para comunicarse con los demás cuando eso podría exponer la propia ignorancia. Se requiere valor para mantener el sistema simple, dejando para mañana las decisiones de mañana. Y, sin un sistema simple, comunicación constante y retroalimentación, es difícil ser valeroso.

Historia de XP

- Creado por Kent Beck para la plantilla del proyecto C3 en Chrysler
 - Kent fue contratado para dirigir el proyecto
 - Durante el proceso nació una nueva metodología: eXtreme Programming (XP)
 - C3 concluyó exitosamente en 1997

Roles XP

- Programador (*Programmer*)
 - Responsable de decisiones técnicas
 - Responsable de construir el sistema
 - Sin distinción entre analistas, diseñadores o programadores
 - En XP, los programadores diseñan, programan y realizan las pruebas
- Jefe de Proyecto (*Manager*)
 - Organiza y guía las reuniones
 - Asegura condiciones adecuadas para el proyecto

- Cliente (*Customer*)
 - Es parte del equipo
 - Determina qué construir y cuándo
 - Establece las pruebas de aceptación
- Encargado de Pruebas (*Tester*)
 - Ayuda al cliente con las pruebas de aceptación
 - Se asegura de que las pruebas aceptación se superan
- Rastreador (*Tracker*)
 - *Metric Man*
 - Observa sin molestar
 - Conserva datos históricos
 -
- Entrenador (*Coach*)
 - Responsable del proceso
 - Tiende a estar en un segundo plano a medida que el equipo madura

Captura de Requisitos en XP

- Historias del Usuario (*User-Stories*)
 - Establecen los requisitos del cliente
 - Las establece el cliente
 - Son la base para las pruebas de aceptación
 - Se caracterizan por prioridad, riesgo y esfuerzo

Planificación en XP

- Planificación por entregas (*releases*)
- Las Historias de Usuario se priorizan por el cliente seleccionando aquellas más importantes para el negocio
- Entregas:
 - Son lo más pequeñas posibles
 - Se dividen en iteraciones (iteración = 2 o 3 semanas)
 - Cada iteración está compuesta por historias de usuario

Programación en XP

- Cada Historia de Usuario se descompone en tareas de programación
- La programación de tareas se realiza por parejas
- La pareja diseña, prueba, implementa e integra el código de la tarea
- Código dirigido por las pruebas
- Código modular, intentando refactorizar siempre que se pueda

Espacio de trabajo XP

- Espacio abierto
- Mesas centrales
- Cubículos en el espacio exterior

Prácticas XP

- **El juego de la planificación:**
 - Decisiones de negocio (cliente):
 - **Alcance** → ¿Cuándo debe estar listo el producto para que sea valioso en producción?
 - **Prioridad** → Prioriza la incorporación de las Historias de Usuario
 - **Composición de entregas** → ¿Qué se necesita para que el negocio mejore?
 - **Fechas de entrega** → Fechas cuando el software en operación causarían una gran diferencia
 - Decisiones técnicas (programadores y otros):
 - **Estimaciones** → ¿Cuánto tiempo tardará en implementarse una Historia de Usuario?
 - **Consecuencias** → Tener en cuenta las consecuencias técnicas de determinadas decisiones de negocio
 - **Proceso** → Organización del proceso y el equipo
 - **Planificación detallada** → Dentro de una entrega, qué Historias de Usuario se realizan primero. Intentar trasladar los segmentos de desarrollo más arriesgados al principio, intentando respetar las prioridades del negocio
 - Reunión diaria “Stand-up Meeting”
 - Todo el equipo
 - ✦ Problemas
 - ✦ Soluciones
 - De pie en un círculo
 - ✦ Evitar discusiones largas
 - ✦ Sin conversaciones separadas
- **Entregas pequeñas**
 - Cada entrega es lo más corta posible:
 - Incluir requisitos más valiosos del sistema (básicos)
 - Reducir el riesgo → mayor retroalimentación desde el cliente, y más frecuente
 - Minimizar el nº de Historias de Usuario que componen una entrega → No realizar Historias de Usuario a medias
- **Diseño simple**
 - Se diseña “la cosa más simple que pueda funcionar”
 - Uso de tarjetas CRC
 - Diseño de software correcto, es aquel que:
 - Supera todas las pruebas
 - No tiene lógica duplicada
 - Pone de manifiesto las intenciones importantes de los programadores
 - Tiene el mínimo número de clases y métodos

Pruebas

- Las pruebas unitarias se escriben ANTES que el código
 - Pruebas automatizadas
 - Permiten el desarrollo de proyectos de forma rápida y segura
 - Pruebas unitarias → programadores
 - Pruebas de aceptación → cliente
 - Resultado → Un programa cada vez más seguro
-
- **Refactoring**
 - Refactorización = Mejora de la arquitectura sin cambiar el comportamiento del sistema
 - Intentar eliminar complejidad
 - Código duplicado → Refactorización
 - Se plantea su aplicación antes de enfrentar una Historia de Usuario/Tarea
-
- **Programación en parejas**
 - Toda el código se escribe en parejas
 - Se produce código de mayor calidad
 - Extiende el conocimiento
 - “Se realiza el trabajo de 1 persona en casi la mitad del tiempo y mejor” (cuestionable)
-
- **Propiedad colectiva**
 - Cualquiera puede modificar el código en cualquier momento → Se evitan cuellos de botella en la codificación
 - Todos asume las responsabilidades sobre el conjunto del sistema
 - Todos conocen algo sobre todas las partes y conocen muy bien aquéllas en las que trabajan
-
- **Integración continua**
 - El código se integra y se prueba después de pocas horas
 - Existe un ordenador dedicado para la integración
 - Cada pareja integra su código en dicho ordenador
-
- **Semana de 40 horas**
 - Filosofía: “Los programadores que descansan son más productivos”
 - El exceso de trabajo es un serio problema en un proyecto
 - La gente está más fresca y tiene mejores ideas

- **Cliente in situ**
 - Cliente real = Aquel que usará el sistema cuando esté en producción
 - El cliente real debe estar con el equipo de trabajo:
 - Responder preguntas
 - Resolver disputas
 - Establecer prioridades
 - Discutir mejoras
- **Estándares de programación**
 - Son fundamentales cuando los programadores cambian de pareja o hacen *refactoring* del código de otros
 - Se consigue un código con el mismo estilo, homogéneo, legible
- **SCRUM:**

Es un proceso ágil que nos permite centrarnos en ofrecer el más alto valor de negocio en el menor tiempo. Nos permite rápidamente y en repetidas ocasiones inspeccionar software real de trabajo (cada dos semanas o un mes). El negocio fija las prioridades. Los equipos se auto-organizan a fin de determinar la mejor manera de entregar las funcionalidades de más alta prioridad. Cada dos semanas o un mes, cualquiera puede ver el software real funcionando y decidir si liberarlo o seguir mejorándolo en otro sprint.

Características:

Divide tu organización en equipos pequeños, interdisciplinarios y auto-organizados.

- Divide el trabajo en una lista de entregables pequeños y concretos.
- Ordena la lista por orden de prioridad y estima el esfuerzo relativo de cada elemento.

Que son los sprints?

- En Scrum los proyectos avanzan en una serie de “Sprints”
- Análogo a las iteraciones en XP
- La duración típica es 2–4 semanas o a lo sumo un mes calendario
- La duración constante conduce a un mejor ritmo
- El producto es diseñado, codificado y testeado durante el Sprint

Empresas que han utilizado SCRUM:

- | | |
|---------------------|-------------|
| • Microsoft | Capital One |
| • Yahoo | BBC |
| • Google | Intuit |
| • Nokia | Philips |
| • High Moon Studios | Siemens |

- **KANBAN:**

Es una metodología de trabajo de origen japonés, que permite gestionar de manera más eficiente y dinámica el ciclo de vida de los proyectos, ya que se trabaja de forma más directa y visual, simplificando así la documentación y gestión.

Una de las características más visibles de esta metodología es el uso de tarjetas o etiquetas de instrucción, por lo que se le denomina “sistema de tarjetas”, pues en su implementación más sencilla, utiliza tarjetas que se pegan en los contenedores de materiales y que se despegan cuando estos contenedores son utilizados, para asegurar la reposición de dichos materiales. Las tarjetas actúan de testigo del proceso de producción.

La aplicación del método Kanban implica la generación de un tablero de tareas que permitirá mejorar el flujo de trabajo y alcanzar un ritmo sostenible. Para implantar esta metodología, deberemos tener claro los siguientes aspectos:

Definir el flujo de trabajo de los proyectos: para ello, simplemente deberemos crear nuestro propio tablero, que deberá ser visible y accesible por parte de todos los miembros del equipo.

Cada una de las columnas corresponderá a un **estado concreto del flujo de tareas**, que nos servirá para saber en qué situación se encuentra cada proyecto. El tablero debe tener tantas columnas como estados por los que pasa una tarea, desde que se inicia hasta que finaliza (p.e: diagnóstico, definición, programación, ejecución, testing, etc.).

El origen de la **metodología Kanban** debemos buscarlo en los procesos de producción “just-in-time” (JIT) ideados por Toyota, en los que se utilizaban tarjetas para identificar necesidades de material en la cadena de producción.

Actualmente, el término **Kanban** ha pasado a formar parte de las llamadas **metodologías ágiles**, cuyo objetivo es gestionar de manera general cómo se van completando las tareas.

Las principales ventajas de esta metodología es que es **muy fácil de utilizar, actualizar y asumir por parte del equipo**. Además, destaca por ser **una técnica de gestión de las tareas muy visual**, que permite ver a golpe de vista el estado de los proyectos, así como también pautar el desarrollo del trabajo de manera efectiva.

- **PROCESO UNIFICADO ÁGIL:**

El proceso unificado ágil (AUP) es una versión simplificada del Proceso Unificado de Rational (RUP) desarrollada por Scott Ambler. Describe un enfoque simple, fácil de entender, del desarrollo de software de aplicación de negocios usando técnicas y conceptos ágiles que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo desarrollo orientado a pruebas, modelado ágil, gestión de cambios ágil y refactorización de bases de datos para mejorar la productividad

Ventajas:

El personal sabe lo que está haciendo: no obliga a conocer detalles. La gente no va a leer documentación del proceso detallada, pero quieren algo de orientación a alto nivel y/o formación de vez en cuando.

Simplicidad: está descrito de forma concisa.

Agilidad: procesos simplificados del RUP

Centrarse en actividades de alto valor: esenciales para el desarrollo.

Independencia de herramientas: a disposición del usuario. La recomendación es que se usen las herramientas que mejor se adapten al trabajo, que son con frecuencia herramientas simples.

Fácil adaptación de este producto para que cumpla con las necesidades propias.

Desventajas:

Es **un producto muy pesado** debido a su complejidad

Al ser un **proceso simplificado del RUP**, muchos desarrolladores eligen trabajar con RUP, ya que este posee más detalles en el proceso

- **CRYSTAL:**

Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo. El factor más importante es la comunicación.

Crystal da vital importancia a las personas que componen el equipo de un proyecto, y por tanto sus puntos de estudio son:

- ✓ Tamaño de un equipo (número de componentes)
- ✓ Comunicación entre los componentes
- ✓ Distintas políticas a seguir
- ✓ Espacio físico de trabajo



Principios

- El grado de detalle necesario en documentar requerimientos, diseño, planeamiento, etc., varía según el proyecto.
- Es imposible eliminar toda documentación pero puede ser reducida logrando un modo de comunicación más accesible, informal y precisa, que pueda ser accedido por todos los miembros del equipo.
- El equipo ajusta constantemente su forma de trabajo para lograr que cada personalidad encaje con los otros miembros, con el entorno y las particularidades de cada asignación.

Características:

- Cuando el número de personas aumenta, también aumenta la necesidad de coordinar.
- Cuando el potencial de daños se incrementa, la tolerancia a variaciones se ve afectada.
- La sensibilidad del tiempo en que se debe estar en el mercado varía: a veces este tiempo debe acortarse al máximo y se toleran defectos, otras se enfatiza la auditoria, confiabilidad, protección legal, entre otros.
- Las personas se comunican mejor cara a cara, con la pregunta y la respuesta en el mismo espacio de tiempo.
- El factor más significativo es la “comunicación”.

Valores:

- ⊙ Frecuencia en las entregas
- ⊙ Comunicación
- ⊙ Crecimiento reflexivo
- ⊙ Seguridad personal
- ⊙ Concentración
- ⊙ Usuarios expertos
- ⊙ Entorno técnico para pruebas automatizadas

- **MÉTODO DE DESARROLLO DE SISTEMAS DINÁMICOS (MDSO):**

Provee un framework para el desarrollo ágil de software, apoyado por su continua implicación del usuario en un desarrollo iterativo y creciente que sea sensible a los requerimientos cambiantes, para desarrollar un sistema que reúna las necesidades de la empresa en tiempo y presupuesto.

Principios:

Involucrar al cliente es la clave para llevar un proyecto eficiente y efectivo, donde ambos, cliente y desarrolladores, comparten un entorno de trabajo para que las decisiones puedan ser tomadas con precisión.

El equipo del proyecto debe tener el poder para tomar decisiones que son importantes para el progreso del proyecto, sin esperar aprobación de niveles superiores.

Se centra en la ***entrega frecuente de productos***.

Entregar un sistema que satisface las actuales necesidades de negocio. No está dirigida tanto a proporcionar un sistema perfecto que resuelva todas las necesidades posibles del negocio, si no que centra sus esfuerzos en aquellas funcionalidades críticas para alcanzar las metas establecidas en el proyecto/negocio.

El desarrollo es iterativo e incremental, guiado por la realimentación de los usuarios para converger en una solución de negocio precisa.

Todos los ***cambios*** durante el desarrollo ***son reversibles***.

El ***alcance de alto nivel y los requerimientos deberían ser base-lined*** antes de que comience el proyecto.

Las pruebas son realizadas durante todo el ciclo vital del proyecto. Esto tiene que hacerse para evitar un caro coste extraordinario en arreglos y mantenimiento del sistema después de la entrega.

La comunicación y cooperación entre todas las partes interesadas en el proyecto es un prerequisite importante para llevar un proyecto efectivo y eficiente.

Características:

- Participación del usuario.
- Desarrollo iterativo y creciente.
- Frecuencia de entrega mejorada.
- Pruebas integradas en cada fase.

- La aceptación de los productos entregados dependen directamente del cumplimiento de los requisitos.
- Todos los cambios durante el desarrollo son reversibles.
- El principal criterio de aceptación de entregables reside en entregar un sistema que satisfice las actuales necesidades de negocios.
- El equipo del proyecto debe tener el poder para tomar decisiones que son importantes para el progreso del proyecto, sin esperar aprobación de niveles superiores.

PRÁCTICAS PROPUESTAS UNIDAD II

Práctica Numero 5. Cuadro comparativo.

Competencias a desarrollar:

- Capacidad de análisis y síntesis.
- Capacidad de organizar y planificar.
- Conocimientos básicos de la carrera.
- Comunicación oral y escrita en su propia lengua.
- Habilidades básicas de manejo de la computadora.
- Conocimiento de una segunda lengua.
- Habilidad para gestión de información.
- Habilidad para buscar y analizar información.

Introducción: Elaborar un cuadro comparativo de las diferentes tipos de metodologías ágiles vistas en clase, que permita visualizar, sus principios y características.

Correlación con el o los temas y subtemas del programa de estudio vigente:
Todos los temas

Material y equipo: Equipo de cómputo

Metodología.

En base a la información vista en clase, el estudiante elaborará cuadro comparativo donde se visualice claramente los principios y características, de cada una de las metodologías ágiles para el desarrollo de software. Para la entrega y posterior evaluación de esta práctica, deberá tomarse como base una rúbrica (**Anexo 5**).

Práctica Número 6.- Debate

Competencias a desarrollar:

- Capacidad de análisis y síntesis.
- Capacidad de organizar y planificar.
- Conocimientos básicos de la carrera.
- Comunicación oral y escrita en su propia lengua.
- Habilidades básicas de manejo de la computadora.
- Conocimiento de una segunda lengua.
- Habilidad para gestión de información.
- Habilidad para buscar y analizar información.
- Capacidad de trabajo en equipo.

Introducción: El debate es una forma de comunicación en la que se discute en torno a un tema. Involucra desacuerdo, ya que cada una de las partes que participan en el debate tiene un punto de vista u opinión que debe defender.

La finalidad de este debate es leer, comprender y discutir la utilidad práctica de cada una de las metodologías.

Correlación con el o los temas y subtemas del programa de estudio vigente:

Tipos de metodologías ágiles, principios y características

Material y equipo: Equipo de cómputo para la investigación

Metodología.

Para la presentación del debate instructivo, que será formal, los estudiantes deberán de haberse documentado adecuadamente para el mismo (por lo que el cuadro comparativo anterior será de mucha utilidad para este fin) deberán conformarse en equipos representativos de las metodologías ágiles más utilizadas, para su respectiva defensa en los dos sentidos (acuerdo y desacuerdo). El desarrollo del debate y su evaluación se basará en una rúbrica (**Anexo 6**)

Práctica Número 7.- Investigación sobre uso de una metodología ágil

Competencias a desarrollar:

- Capacidad de análisis y síntesis.
- Comunicación oral y escrita en su propia lengua.
- Habilidades básicas de manejo de la computadora.
- Conocimiento de una segunda lengua.
- Habilidad para gestión de información.
- Capacidad de trabajo en equipo.
- Capacidad de aplicar los conocimientos en la práctica

Introducción: La teoría vista en clase es la base de cada una de las metodologías ágiles, sin embargo, igual que cualquier otra dentro del desarrollo de software, estas metodologías pueden ser (sin perder su base teórica), modificadas en su práctica o mejor dicho adaptadas según sea la forma de trabajar en las empresas. Son adaptadas según sus propias necesidades, por lo tanto se considera, es importante que el estudiante lo vea por sí mismo.

Correlación con el o los temas y subtemas del programa de estudio vigente:
Tipos de metodologías ágiles, principios y características

Material y equipo: Equipo de cómputo

Metodología.

Deberán visitar una empresa en la cual puedan obtener información, ver, conocer y documentar (con fotos, copias, videos, etc.) como esa empresa aplica al menos una metodología ágil, lo que le permitirá saber que la puesta en práctica de una metodología, no necesariamente debe ser como lo vemos en clase y en las lecturas, ya que depende mucho de cada empresa, la adaptación que haga de la misma. Para la presentación de esta investigación deberán los estudiantes conformarse en equipos y basada en una lista de cotejo (**Anexo 7**).

Práctica Número 8.- Dinámica “Toma la pelota y suéltala”

Competencias a desarrollar:

- Capacidad de análisis y síntesis.
- Capacidad de organizar y planificar.
- Comunicación oral y escrita en su propia lengua.
- Capacidad de trabajo en equipo.

Introducción: ¿Que son las dinámicas de grupo? Son actividades estructuradas de **carácter vivencial** cuyo objetivo es conseguir la cohesión del grupo a través de la diversión. ¿Cuáles son sus beneficios?:

- Sentimiento de pertenencia
- Aceptación de la importancia del grupo
- Autoconocimiento
- Comunicación auténtica entre los miembros del grupo
- Participación, motivación para contribuir al bienestar del grupo
- División de tareas
- Colaboración y socialización

Al ser fundamental el trabajo en equipo utilizando metodologías ágiles, la aplicación de esta dinámica ofrece todos los beneficios antes mencionados, además de tratar de instruir en el trabajo que tendrán que realizar durante el desarrollo del proyecto integrador utilizando el framework de Scrum, poniendo en práctica la retroalimentación del equipo con el fin de mejorar en cada sprint. Adicional a los beneficios y de forma precisa con el fin de esta dinámica para el tema de esta materia, contribuye también en los rubros de la planeación, toma de decisiones y gestión del tiempo.

Correlación con el o los temas y subtemas del programa de estudio vigente:

Tipos de metodologías ágiles, principios y características y Metodología Scrum

Material y equipo: Pelotas de tenis o de tamaño similar, Un cronómetro, Hojas de rotafolio o pizarrón, plumón y 2 cajas

Metodología.

Definir roles de los participantes:

- Un facilitador
- Un controlador de cronómetro y pizarrón
- Grupo de participantes

Para la realización de esta dinámica los participantes forman un círculo. Uno de ellos se hará cargo de la caja con las pelotas, la otra estará vacía al inicio.

Uno de los participantes se ubicara fuera del circulo y tendrá el control del cronómetro, pizarrón y plumón.

La dinámica inicia **tomando** una pelota de la caja y **pasándola** al compañero de a lado derecho y de inmediato la siguiente pelota, de esa manera se hará circular el total de pelotas en la caja. La caja vacía se va llenando con las pelotas que llegan, al llegar todas las pelotas el cronómetro para y se anota el tiempo realizado, en el pizarrón. Se sugiere se lleven a cabo al menos 3 rondas (depende de cómo se desarrolle la dinámica podremos decidir el que sean más rondas). En cada ronda los participantes deben analizar cómo se llevó a cabo esta ronda, que errores se cometieron y llegar a un acuerdo acerca de qué modificaciones llevarán a cabo en la nueva ronda, con la finalidad de desaparecer errores y disminuir el tiempo con respecto a la ronda anterior. Se contabiliza como un error cuando una de las pelotas cae.

La intención es lograr concluir con cero errores en el menor tiempo posible respetando todas las condiciones.

El papel del facilitador:

En su etapa inicial, el facilitador proporciona las instrucciones para la realización del ejercicio.

A lo largo del desarrollo se mantiene como observador del proceso

Si en una ronda no se cumple con el tiempo o no disminuye el número de errores el facilitador deberá proponer un tiempo fuera para propiciar que los participantes dialoguen sobre cómo se sienten, revisen su estrategia, tomen decisiones, generen propuestas, realicen cambios, experimenten.

El facilitador puede realizar comentarios acerca de la energía, fortalezas y oportunidades en el grupo, de acuerdo a lo que observa.

Por ser de las dinámicas de liderazgo para mejorar el trabajo en equipo más completas en cuanto a la oportunidad de practicar y desarrollar habilidades, el facilitador puede sugerir se lleven a la práctica una a una de ellas.

Práctica Número 9.- Cuestionario

Competencias a desarrollar:

- Capacidad de análisis y síntesis.
- Conocimientos básicos de la carrera.
- Comunicación oral y escrita en su propia lengua.
- Habilidades básicas de manejo de la computadora.

Introducción: En un cuestionario (examen) el estudiante deberá demostrar el conocimiento y entendimiento de los conceptos y temas vistos en la unidad II. El examen es un método de aprendizaje que le permite al estudiante recordar los temas vistos partiendo principalmente de la comprensión de la teoría vista en clase.

Esta práctica permitirá al estudiante reafirmar los conceptos base de las diferentes metodologías ágiles, sus principios y características, además de servir al docente como guía para llevar a cabo su propia instrumentación didáctica

Correlación con el o los temas y subtemas del programa de estudio vigente:
Todos los temas de esta unidad

Material y equipo: Equipo de cómputo

Metodología.

El estudiante deberá presentarse en laboratorio de cómputo y contestar el cuestionario (examen) en la fecha y hora indicada, utilizando la plataforma moodle, para lo cual el docente otorgará clave de acceso al examen y de esta manera demostrar que conoce y comprende los conceptos y temas que integran esta unidad.

Considerando que el cuestionario es un examen, se presenta para apoyo del docente el desarrollo de uno con sus respuestas y el formato de retroalimentación (**Anexo 8**).

Unidad III. METODOLOGÍA SCRUM

Competencia específica a desarrollar: Aplicar la metodología Scrum a un problema real

Competencias a adquirir:

- Capacidad de análisis y síntesis.
- Capacidad de organizar y planificar.
- Conocimientos básicos de la carrera.
- Comunicación oral y escrita en su propia lengua.
- Habilidades básicas de manejo de la computadora.
- Conocimiento de una segunda lengua.
- Habilidad para gestión de información.
- Habilidad para buscar y analizar información.
- Capacidad de trabajo en equipo.
- Capacidad de aplicar los conocimientos en la práctica

Subtemas:

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

Scrum es un proceso ágil que nos permite centrarnos en ofrecer el más alto valor de negocio en el menor tiempo.

Nos permite rápidamente y en repetidas ocasiones inspeccionar software real de trabajo (cada dos semanas o un mes).

El negocio fija las prioridades. Los equipos se auto-organizan a fin de determinar la mejor manera de entregar las funcionalidades de más alta prioridad.

Cada dos semanas o un mes, cualquiera puede ver el software real funcionando.

SCRUM FRAMEWORK

ROLES

PRODUCTO OWNER:

- Define las funcionalidades del producto
- Decide sobre las fechas y contenidos de los releases
- Es responsable por la rentabilidad del producto (ROI)
- Prioriza funcionalidades de acuerdo al valor del mercado/negocio
- Ajusta funcionalidades y prioridades en cada iteración si es necesario
- Acepta o rechaza los resultados del trabajo del equipo

SCRUM MASTER:

- Representa a la gestión del proyecto
- Responsable de promover los valores y prácticas de Scrum
- Remueve impedimentos
- Se asegura de que el equipo es completamente funcional y productivo
- Permite la estrecha cooperación en todos los roles y funciones
- Escudo del equipo de interferencias externas

TEAM

- Típicamente de 5 a 9 personas
- Multi-funcional:
 - Programadores, testers, analistas, diseñadores, etc.
- Los miembros deben ser full-time
 - Puede haber excepciones (Ej.: Infraestructura, SCM, etc.)
- Los equipos son auto-organizativos
 - Idealmente, no existen títulos pero a veces se utilizan de acuerdo a la organización
- Solo puede haber cambio de miembros entre los sprints

REUNIONES

SPRINT PLANNING

- El equipo selecciona los temas a partir del Product Backlog que pueden comprometerse a completar
- Se crea el Sprint Backlog
 - Se identifican tareas y cada una es estimada
 - Realizado colaborativamente, no solo por el ScrumMaster
- El diseño de Alto Nivel es considerado

ESTIMACIÓN

Es una labor de equipo: todos los miembros del equipo deben involucrarse en estimar cada historia. ¿Por qué?:

- A la hora de planificar, normalmente no sabemos exactamente quién implementará qué partes de cada historia.
- Las historias normalmente involucran a bastantes personas y de diferentes áreas de experiencia (diseño de interfaz de usuario, programación, pruebas, etc.).
- Debe pedirse que todos participen. Cuando pedimos a todo el mundo que dé estimaciones, muchas veces encontramos discrepancias en las que dos miembros del equipo tienen estimaciones tremendamente distintas sobre la misma historia. Es mejor descubrir tales cosas al principio.

Se sugiere utilizar una herramienta que nos ayude a la estimación como lo es el **Planning Poker**:



- Cada miembro del equipo cuenta con una baraja de 13 cartas, como las que se muestran en la imagen. Cada vez que hay que estimar una historia, cada miembro del equipo selecciona una carta que representa su estimación de tiempo (en puntos de historia) y la coloca bocabajo en la mesa.
- Cuando todos los miembros del equipo han preparado sus cartas, se les da la vuelta al mismo tiempo. Así obligamos a cada miembro del equipo a pensar por sí mismo en lugar de seguir la estimación de otro.
- Si hay mucha discrepancia entre dos estimaciones, el equipo discute las diferencias y trata de construir una imagen común del trabajo necesario para la historia. Pueden hacer algún tipo de división en tareas. Después, el equipo estima de nuevo. Este bucle se repite hasta que la estimación de tiempo converge, es decir, que todas las estimaciones sean *aproximadamente las mismas* para esa historia.
- Se deben estimar el total de tiempo necesario para la historia. No solamente “su” parte del trabajo.

Date cuenta de que la secuencia de números no es lineal. Por ejemplo, no hay nada entre 40 y 100. ¿Por qué?:

- Esto es para evitar una falsa sensación de exactitud para las estimaciones de tiempo más grandes. Si una historia se estima aproximadamente en 20 puntos, no es relevante discutir si deberían ser 20, 18 o 21. Todo lo que sabemos es que es una historia grande y es difícil de estimar. Así que 20 es nuestra idea aproximada.

Algunas cartas especiales sobre las que hablar:

- Q = “esta historia ya está hecha” o “esta historia es prácticamente nada, apenas unos minutos de trabajo”.
- ¿ = “no tengo ni la más remota de las ideas. Nada”.
- Taza de café = “estoy demasiado cansado para pensar. Tomemos un descanso”.

SPRINT REVIEW

- El equipo presenta lo realizado durante el sprint
- Normalmente adopta la forma de un demo
- Informal
 - Regla de 2 hs preparación
- Todo el equipo participa
- Se invita a todo el mundo

SPRINT RETROSPECTIVE

- Normalmente 15 a 30 minutos
- Se realiza luego de cada sprint
- Todo el equipo participa
 - ScrumMaster
 - Product owner
 - Equipo
 - Posiblemente clientes y otros
 - Todo el equipo se reúne y discute lo que les gustaría Comenzar a hacer, Dejar de hacer, Continuar haciendo (Start / Stop / Continue)

** Esto es sólo una de las muchas maneras de hacer una retrospectiva.

DAILY SCRUM MEETING

- Parámetros
 - Diaria
 - Dura 15 minutos
 - Parados
- No para la solución de problemas
 - Todo el mundo está invitado
 - Sólo los miembros del equipo, ScrumMaster y Product Owner, pueden hablar
 - Ayuda a evitar otras reuniones innecesarias

ARTEFACTOS

PRODUCTO BACKLOG

- Los requisitos.
- Una lista de todos los trabajos deseados en el proyecto.
- Idealmente cada tema tiene valor para el usuario o el cliente.
- Priorizada por el Product Owner.
- Repriorizada al comienzo de cada Sprint

SPRINT BACKLOG

- Los individuos eligen las tareas
- El trabajo nunca es asignado
- Cualquier miembro del equipo puede añadir, borrar o cambiar el Sprint Backlog
- El trabajo para el Sprint emerge
- Si el trabajo no está claro, definir un tema del Sprint Backlog con una mayor cantidad de tiempo y subdividirla luego.
- Actualizar el trabajo restante a medida de que más se conoce

BURNDOWN CHARTS

El diagrama de Burndown sirve para saber el tiempo que falta para completar el trabajo. Normalmente se utiliza para saber cuánto falta para terminar las historias comprometidas en un sprint.

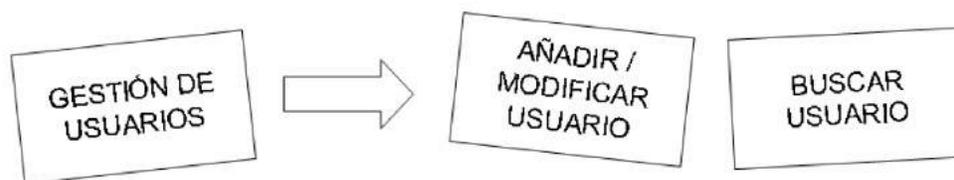
TABLON DE TAREAS

Un Tablón, tablero o pizarra Scrum, es una parte integral de la metodología Scrum. Hace la carga de trabajo visible y ayuda a los equipos a gestionar el desarrollo del producto de principio a fin. Permite visualizar como las historias de usuario son divididas en tareas y de esta manera definir el sprint backlog.

¿Cuál es la diferencia entre historias y tareas?

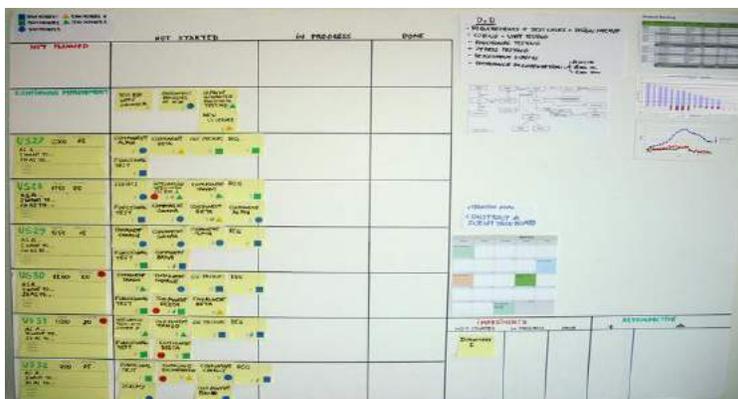
La diferencia es que las historias son entregables de los que el Dueño de Producto se preocupa. Las tareas son no-entregables, o aspectos de los que el Dueño de Producto no se preocupa.

Ejemplo de división de una historia en tareas:



En términos generales un tablón de tareas debe contener las siguientes columnas o apartados:

Por hacer, haciendo o en proceso, terminado, imprevistos



Este es un ejemplo de pizarra, Al inicio de un proyecto



Este es el mismo tablón, con el proyecto ya en proceso

TESTING ÁGIL (Agile Testing)

Agile Testing es una práctica de pruebas de software que sigue los principios del desarrollo ágil de software.

Involucra a todos los miembros de un equipo ágil multifuncional, en el cual el rol del tester es el de un experto multifuncional, garante que se entregue el valor de negocio deseado por el cliente a un ritmo sostenible y continuo.

Las metodologías ágiles no ven al software testing como una fase separada, sino como parte integral del Desarrollo de software al igual que la programación.

Agile Testing, incorpora una serie prácticas, como por ejemplo Testing de “todo el equipo”, Testing independiente (opcional), Integración continua, Testing guiado por pruebas (**Test Driven Development – TDD**), Desarrollo guiado por comportamiento (Behaviour Driven Development – BDD), Desarrollo guiado por pruebas de aceptación (Acceptance Test Driven Development – ATDD), entre otros.

Los equipos ágiles utilizan un enfoque de “todo el equipo” al testing, con la finalidad de integrar la calidad al desarrollo del producto, al contrario de un enfoque de primero fabricar el producto y luego inspeccionar para determinar su nivel de calidad.

El rol del tester en un equipo ágil es el de un experto, garante que se entregue el valor de negocio deseado por el cliente a un ritmo sostenible y continuo.

Para ello, utiliza la “especificación mediante ejemplos” para capturar los comportamientos deseados y no deseados para guiar la codificación.

El foco del Tester en un entorno Agile está en la aplicación de enfoques tipo Behaviour Driven Development (BDD), usualmente trabajando en paralelo con el equipo de desarrollo y no en la fase final.

Esto se contrapone con el rol convencional del tester, en el cual es un profesional encargado de elaborar diseños de prueba a partir de diseños funcionales, y luego ser un simple ejecutor de guiones preestablecidos.

En un entorno Agile el rol del tester es de mayor especialización técnica, considerando que debe manejar herramientas de automatización, gestión ágil y metodologías.

Además, el rol también posee mayor interacción con otras personas como por ejemplo el cliente o los desarrolladores, por lo que también necesitará habilidades blandas de comunicación, orientación al cliente, negociación, entre otras.

PRINCIPIOS

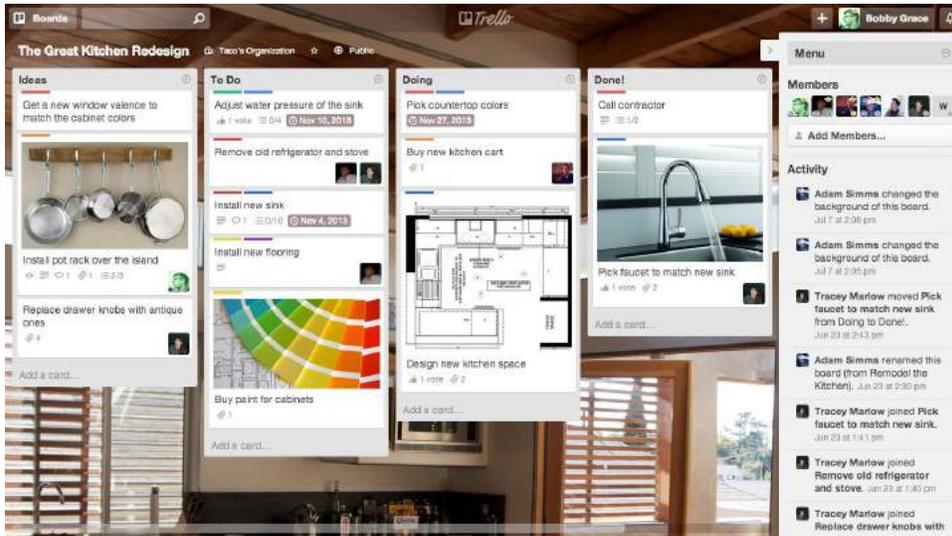
De forma similar a que el Manifiesto Ágil contiene principios que se aplican al desarrollo ágil de software, el Agile Testing engloba los siguientes principios:

- **El Testing no es una fase:** El testing continuo es la única forma de garantizar avance continuo, por esto, el testing se realiza continuamente junto con el desarrollo de software y demás actividades.
- **El Testing hace avanzar el proyecto:** Bajo métodos convencionales, el testing es una alcabala, en cambio en Agile Testing se proporciona retroalimentación continua, permitiendo corregir el rumbo continuamente durante el desarrollo de software.
- **Todo el equipo realiza pruebas:** en Agile Testing, los Analistas de negocio y Desarrolladores de software también ejecutan pruebas, no sólo los testers como en métodos convencionales.
- **Reducir el tiempo para recibir retroalimentación:** En Agile Testing, los equipos del área de negocio (el cliente) están involucrados en cada iteración, no solo al final durante la fase de aceptación, como resultado, el tiempo de retroalimentación se reduce y el costo de correcciones también es menor.
- **Código limpio:** Los defectos en el código se corrigen en la misma iteración, por lo que se mantiene el código limpio.
- **Reducir la documentación de pruebas:** Los Agile Testers usan listas de chequeo reusables en lugar de documentación extensa, se enfocan en la esencia de la prueba en lugar de detalles. Siguiendo principios ágiles estas listas de chequeo son el inicio de las definiciones de las pruebas y no el final, y el tester cuenta con libertad para aportar valor.
- **Guiado por pruebas:** El Agile Testing, las pruebas se hacen “durante” el desarrollo y no después del desarrollo como en métodos convencionales.

Herramientas

Aquí algunas herramientas para la Gestión de proyectos Ágiles

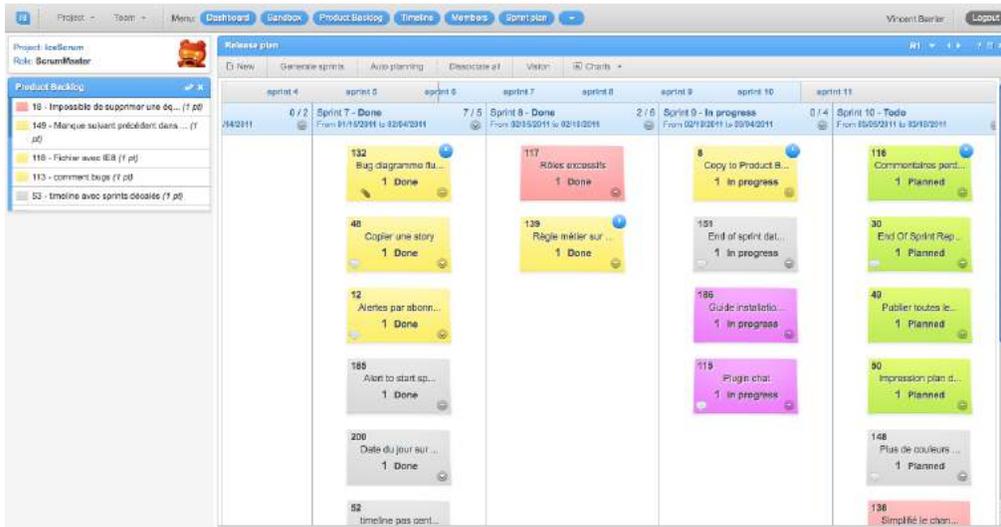
TRELLO



ASSANA



ICE SCRUM



LEANKIT



Herramientas para Agile Testing

1)QMetry:

<http://www.qmetry.com/>

2)TestRail:

<http://www.gurock.com/testrail/>

3)JIRA:

a)Zephyr:

<https://marketplace.atlassian.com/plugins/com.thed.zephyr.je>

b)Go2GroupSynapseRT:

<https://marketplace.atlassian.com/plugins/com.go2group.jira.plugin.synapse>

4)qTest:

<http://www.qasymphony.com/qtest.html>

5)TestLodge:

<http://www.testlodge.com/>

6)Zephyr:

<http://www.getzephyr.com/>

8)Testuff:

<http://www.testuff.com/>

9)TestCollab:

<http://testcollab.com/>

10)Gemini:

<http://www.countersoft.com/solutions/testing/>

11)PractiTest:

<http://www.practitest.com/>

12)TestLink:

<http://sourceforge.net/projects/testlink/>

13)QAComplete:

<http://smartbear.com/products/qa-tools/test-management/>

14)SilkCentral:

<http://www.borland.com/products/silkcentral/>

Práctica Número 10.- Dinámica “Organizando mi casa”

Competencias a desarrollar:

- Capacidad de análisis y síntesis.
- Capacidad de organizar y planificar.
- Comunicación oral y escrita en su propia lengua.
- Habilidad para gestión de información.
- Capacidad de trabajo en equipo.

Introducción: Esta dinámica pretende que el estudiante identifique, comprenda y aplique la distribución de tareas a partir de una necesidad en específico, esto le servirá para saber el nivel de detalle que debe observar cuando se vaya a planear el trabajo a realizar para dar solución al problema planteado usando scrum, además de fomentar el trabajo en equipo, que lo apoye a identificar necesidades y habilidades de cada uno de los participantes para de esta manera distribuir de mejor manera el trabajo. Importante es que analice, se tomen decisiones y lleguen a acuerdos a partir del conocimiento que tienen de las tareas a desarrollar y de las capacidades de sus compañeros.

Correlación con el o los temas y subtemas del programa de estudio vigente:

Con todos los temas pero más con el tema de las Reuniones

Material y equipo: Equipos de trabajo (equipo de estudiantes conformados desde un inicio de semestre), papel y lápiz

Metodología.

El facilitador plantea al grupo lo siguiente:

Partiendo de la suposición que los integrantes del equipo viven en una casa, la cual comparten ya que su lugar de origen es otro, otra ciudad, entonces tienen el reto de limpiar su casa, ya que sus padres vendrán a visitarlos (y ellos, sus padres deben ver que si son capaces de mantener su casa “muy” limpia), “muy limpia significa observar los detalles, no solo hacer la limpieza superficialmente”, esto es algo que los integrantes del equipo deben de pensar (ellos solos tienen que llegar a eso, el facilitador no se los hará ver) o sea no solo deben poner limpiar cocina sino todo lo que eso conlleva por ejemplo limpiar estufa, limpiar refrigerador, etc.

Se les debe definir el tiempo que tienen para llevar a cabo la tarea de “limpiar la casa”, por ejemplo poner un tiempo de 4 hrs. o 5, etc. El facilitador plantea la definición del tamaño de la casa, esto con la finalidad de que los estudiantes “imaginen” el tamaño que se está definiendo, por ejemplo: “Su casa es de 2 pisos, tamaño **mediano**, en la planta baja esta la sala, cocina, comedor, un baño completo, cochera, jardín y patio.

En la planta alta se encuentran 3 recámaras y un baño completo” no se les da más detalle, ellos deben imaginar que es lo que hay dentro de cada habitación, los utensilios que utilizaran y lo demás que necesitan.

Se les pide también desglosar por cada habitación las tareas o actividades que necesitan llevar a cabo para cumplir y dejarla muy limpia (barrer, trapear, sacudir, limpiar refrigerador, lavar azulejo, lavar sanitario, etc), así como calcular el tiempo que lleva realizar cada actividad y en total el tiempo en cada habitación. Deben también indicar qué actividad hará cada uno de los integrantes, debiendo totalizar el tiempo que “trabajó” cada uno según las actividades y tiempos asignados, obviamente cubriendo toda la casa.

Durante el desarrollo del ejercicio el facilitador debe ir observando y guiando a los equipos.

Al final, cada equipo realiza una exposición indicando actividades y tiempos por cada habitación, y distribución del trabajo entre los integrantes. Se tomará como ejemplo para la exposición, una habitación en particular y se analizará el trabajo realizado.

Es importante hacer ver a los estudiantes, que la definición de los problemas tienen que ser más precisos, no puede quedar en duda el tamaño del proyecto, deben como profesionistas conocer a detalle el problema, observar cómo se llevan a cabo las cosas, los procesos y procedimientos dentro de una empresa, conocer exactamente qué es lo que el cliente quiere y espera, no se puede trabajar superficialmente porque dentro de los “gabinetes” se esconderán los verdaderos detalles y de ahí seguramente surgirán problemas

Práctica Número 11.- Mapa conceptual

Competencias a desarrollar:

- Capacidad de análisis y síntesis.
- Capacidad de organizar y planificar.
- Comunicación oral y escrita en su propia lengua.

Introducción: El mapa conceptual permitirá al estudiante Identificar de manera clara todos los conceptos y elementos del framework de Scrum, lo cual es muy importante ya que todo el marco de trabajo de esta metodología lo aplicará en su proyecto integrador y quedará plasmado en su trabajo final.

Correlación con el o los temas y subtemas del programa de estudio vigente:

Con todos los temas

Material y equipo: Papel y lápiz, equipo de cómputo

Metodología.

Tomando como base los temas vistos en clase, el estudiante realizará un mapa conceptual donde demuestre que identifica y comprende los elementos que conforman la metodología Scrum.

Para la presentación y posterior evaluación de esta práctica deberá tomarse como base una lista de cotejo (**Anexo 9**).

Práctica Número 12.- Reporte final y exposición

Competencias a desarrollar:

- Capacidad de organizar y planificar.
- Conocimientos básicos de la carrera.
- Comunicación oral y escrita en su propia lengua.
- Habilidades básicas de manejo de la computadora.
- Conocimiento de una segunda lengua.
- Habilidad para gestión de información.
- Capacidad de trabajo en equipo.
- Capacidad de aplicar los conocimientos en la práctica

Introducción: En este reporte se debe presentar y evidenciar la aplicación del framework Scrum en el desarrollo de un proyecto integrador (software) a partir de la necesidad de una empresa, analizada desde la primera unidad. La exposición se lleva a cabo con la intención de que el resto del grupo conozca el problema analizado y la solución del problema permitiéndole conocer de otros tipos de problemas y soluciones además del desarrollado con su equipo de trabajo. Para el desarrollo de este reporte se debe tomar como base las anotaciones hechas durante el semestre, de las actividades realizadas en relación al seguimiento que dimos (tanto los estudiantes en sus horas dedicadas a cumplir con cada una de las actividades, así como las observaciones hechas por el facilitador en las revisiones de avances del proyecto).

Correlación con el o los temas y subtemas del programa de estudio vigente:

Con todos los temas de la materia en especial con la unidad III

Material y equipo: Equipo de cómputo y herramientas de gestión elegida

Metodología.

Los estudiantes por equipo deberán entregar un reporte del trabajo realizado expresando el desarrollo y la aplicación de cada uno de los pasos que se siguen en Scrum para de esta manera ir cumpliendo las tareas que den como resultado el software del cliente. Este trabajo se presenta basado en una lista de cotejo (**Anexo 10**).

Deberán también realizar una exposición donde expliquen y presenten al resto del grupo los resultados de su trabajo y como fueron resolviendo los problemas durante su desarrollo. La exposición y evaluación se llevará cabo basado en una guía de observación (**Anexo 11**).

Práctica Número 13.- Revisión de seguimiento al proyecto integrador.

Competencias a desarrollar:

- Capacidad de organizar y planificar.
- Conocimientos básicos de la carrera.
- Comunicación oral y escrita en su propia lengua.
- Habilidades básicas de manejo de la computadora.
- Habilidad para gestión de información.
- Capacidad de trabajo en equipo.
- Capacidad de aplicar los conocimientos en la práctica

Introducción: El equipo de trabajo, en cada sesión de asesoría, debe evidenciar el avance del trabajo demostrando cada integrante su compromiso en el mismo. A partir de la tercera revisión de seguimiento, la revisión de seguimiento se deberá ir realizando y revisando conforme al índice que se presenta en la lista de cotejo. En clase se informará que puntos abarcara el avance, según lo que corresponda al seguimiento.

Correlación con el o los temas y subtemas del programa de estudio vigente:
Todos los temas de la unidad I y III

Material y equipo: Papel, lápiz, equipo de cómputo, trabajo de campo en empresa

Metodología.

Todos los integrantes del equipo de trabajo deberá asistir a cada sesión de seguimiento en donde se presentará avance de su proyecto en la fecha que se le indique, representando para el docente la presentación de este avance, una forma más de evaluar y asegurarse que cada uno de los integrantes está participando activamente en el desarrollo del mismo y desde entonces tratar de encauzarlos para asegurar se alcance la competencia.

La presentación del avance se basará en una lista de cotejo (**Anexo 12**) y la evaluación a cada uno de los estudiantes en una guía de observación (**Anexo 13**)

Práctica Número 14.- Investigación sobre diferentes tablonos de tarea

Competencias a desarrollar:

- Capacidad de análisis y síntesis.
- Capacidad de organizar y planificar.
- Comunicación oral y escrita en su propia lengua.
- Habilidades básicas de manejo de la computadora.
- Conocimiento de una segunda lengua.
- Habilidad para gestión de información.
- Habilidad para buscar y analizar información.

Introducción: El estudiante, al investigar sobre este tema, encontrará que existen diferentes tipos de tablonos o pizarras de tareas, lo que le permitirá conocer diferentes formas de implementar scrum. Las empresas toman la filosofía y la adaptan a conveniencia o su forma propia de trabajar.

Correlación con el o los temas y subtemas del programa de estudio vigente:

Todos los temas de esta unidad

Material y equipo: Equipo de cómputo

Metodología.

El estudiante deberá investigar sobre el contenido y diseño de diferentes tablonos de tareas, esto con la intención de que ellos se den cuenta que no existe solo uno, sino que puede adaptarse según las necesidades de cada empresa o cada proyecto. Para la presentación y posterior evaluación de esta práctica deberá tomarse como base una lista de cotejo (**Anexo 14**).

Práctica Número 15.- Investigación sobre herramientas útiles para apoyo de trabajo en Scrum

Competencias a desarrollar:

- Capacidad de análisis y síntesis.
- Capacidad de organizar y planificar.
- Conocimientos básicos de la carrera.
- Comunicación oral y escrita en su propia lengua.
- Habilidades básicas de manejo de la computadora.
- Conocimiento de una segunda lengua.
- Habilidad para gestión de información.
- Habilidad para buscar y analizar información.

Introducción: Actualmente existen varias herramientas electrónicas que apoyan en la gestión de proyectos ágiles y en específico varias para trabajo con scrum por ejemplo, control de versiones, planning póker, agile testing, tablon de tareas, etc

Correlación con el o los temas y subtemas del programa de estudio vigente:

Todos los temas de esta unidad

Material y equipo: Equipo de cómputo

Metodología.

El estudiante debe investigar sobre diferentes herramientas electrónicas que existen y que sirven como apoyo para diversas etapas del desarrollo con metodologías ágiles y algunas específicamente para el trabajo con Scrum. Debe presentar funciones, características, ventajas y desventajas.

Para la presentación y posterior evaluación de esta práctica deberá tomarse como base una lista de cotejo (**Anexo 15**).

Práctica Número 16.- Cuestionario (ESTA ES UNA PRÁCTICA ADICIONAL)**

Competencias a desarrollar:

- Capacidad de análisis y síntesis.
- Conocimientos básicos de la carrera.
- Comunicación oral y escrita en su propia lengua.
- Habilidades básicas de manejo de la computadora.

Introducción: En un cuestionario (examen) el estudiante deberá demostrar el conocimiento y entendimiento de los conceptos y temas vistos en la unidad III. El examen es un método de aprendizaje que le permite al estudiante recordar los temas vistos partiendo principalmente de la comprensión de la teoría vista en clase.

Esta práctica permitirá al estudiante reafirmar los conceptos base de la Metodología Scrum, sus roles, reuniones y artefactos así como el Agile Testing y herramientas de apoyo. además de servir al docente como guía para llevar a cabo su propia instrumentación didáctica

Correlación con el o los temas y subtemas del programa de estudio vigente:

Todos los temas de esta unidad

Material y equipo: Equipo de cómputo

Metodología.

El estudiante deberá presentarse en laboratorio de cómputo y contestar el cuestionario (examen) en la fecha y hora indicada, utilizando la plataforma moodle, para lo cual el docente otorgará clave de acceso al examen y de esta manera demostrar que conoce y comprende los conceptos y temas que integran esta unidad.

Considerando que el cuestionario es un examen, se presenta para apoyo del docente el desarrollo de uno con sus respuestas y el formato de retroalimentación (**Anexo 16**).

BIBLIOGRAFÍA:

[1] Pressman, Roger S., “Ingeniería de Software un enfoque práctico”, Mac Graw Hill. Séptima edición. 2010

[2] Sommerville, Ian., “Ingeniería de Software”, Pearson. Novena Edición. 2011

[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]<https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>

ANEXOS

ANEXO 1
Asignatura: "Metodologías ágiles de desarrollo de Software"

Nombre del alumno:			
Carrera:	CLAVE:	GRUPO:	
No. Control:		Fecha:	

Instrucciones para el evaluador: Marque con una "✓" en la columna SI, cuando el(los) alumno(s) muestre las evidencias de los apartados correspondientes y en la columna NO cuando no muestre las evidencias señaladas.

Lista de cotejo Trabajo # 1 Tema: Introducción a los procesos

No.	Aspecto a evaluar	Valor	Cumple		Pts.
			SI	No	
1	Portada	5			
2	Introducción (breve explicación del tema)	5			
3	Índice	5			
4	Desarrollo (presenta Resumen del tema mínimo 3 cuartillas)	55			
5	Conclusiones (opinión personal)	10			
6	Bibliografía. (Al menos 3 fuentes confiables y formato Apa)	10			
7	Presentación del documento (Justificar y formato)	5			
8	Sin errores ortográficos.	5			
		100	Total		

Nivel de desempeño mínimo aceptable: 80 pts.



RETROALIMENTACION
ASPECTOS POSITIVOS:
AREAS DE OPORTUNIDAD:
SUGERENCIAS DE MEJORA:
NOMBRE Y FIRMA DE CONFORMIDAD DEL ALUMNO: _____



ANEXO 2
Asignatura: "Metodologías ágiles de desarrollo de Software"

Nombre del alumno:		
Carrera:	CLAVE:	GRUPO:
No. Control:		Fecha:

Cuestionario Unidad 1. Introducción a los procesos

Con el presente instrumento se pretende evaluar los conocimientos necesarios para acreditar la Unidad 1.

Instrucciones:

1. El cuestionario se hará en la plataforma de apoyo en línea www.materias.ith.mx/moodle
2. El cuestionario se aplicará en el laboratorio de sistemas y computación en el horario de clases.
3. El maestro proporcionará la contraseña y acceso en la fecha y hora de aplicación indicado en el grupo.
4. Para el llenado de la aplicación en línea el maestro se recomienda que lea con atención las instrucciones correspondientes a cada sección o reactivo.
5. Conteste las preguntas que se presentan en esta evaluación escribiendo su respuesta directamente en el cuestionario.
6. En caso de que tenga cualquier duda sobre la evaluación, pregunte al facilitador antes de contestar la respuesta en el cuestionario.
7. Se estima que el tiempo aproximado para contestar la evaluación es de 30 minutos.



CUESTIONARIO

1. Seleccione los principios de las metodologías ágiles según corresponda:

Seleccione una o más de una:

- a. En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.
- b. Cada uno de los entregables, serán sometidos a un script de calidad, que ejecutara la organización, y no serán admitidos como productos del proyecto hasta alcanzar un nivel aceptable.
- c. Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
- d. La documentación no se libera al final del proyecto, sino en entregables parciales.
- e. En cada iteración se revisará y actualizará el documento "Lista de Riesgos", añadiendo además la lista de riesgos más importantes actualizada por cada iteración.
- f. La simplicidad es esencial.
- g. La atención continua a la calidad técnica y al buen diseño mejora la agilidad.

2. Los procesos ágiles de desarrollo de software, también son conocidos como

Respuesta:

3. La Motivación, Gestión del tiempo, Negociación, Presentaciones en público, Cómo dirigir reuniones, Interacción personal, Comunicación, Diseño de la propia estrategia empresarial, Delegación, Gestión del estrés, Toma de decisiones y Resolución de conflictos son algunas habilidades blandas más valoradas en las organizaciones.

Elija una;

- Verdadero
- Falso



4. Organización dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos.

Respuesta:

5. Uno de los principios ágiles nos dice que las mejores arquitecturas, requisitos y diseños surgen de los equipos que se dirigen organizadamente.

Elija una;

- Verdadero
- Falso

6. Es fundamental que en las metodologías tradicionales la gente del negocio y los desarrolladores trabajen juntos a lo largo del proyecto.

Elija una;

- Verdadero
- Falso

7. El manifiesto valora: Seleccione una o más de una:

- a. Más la negociación contractual que la colaboración con el cliente
- b. Más el seguimiento de un plan que la respuesta al cambio
- c. Menos la negociación contractual que la colaboración con el cliente
- d. Más la respuesta al cambio que el seguimiento de un plan.
- e. Más el software que funciona que la documentación exhaustiva.
- f. Menos el software que funciona que la documentación exhaustiva.
- g. Menos a los individuos y su interacción que a los procesos y las herramientas.
- h. Menos a los procesos y las herramientas que a los individuos y su interacción.



8. Para que una organización requiera agilidad en la ejecución de sus proyectos de software es necesario contar con:

Respuesta:

9. Es el documento que resume la filosofía "ágil".

Respuesta:

10. Le dan un valor agregado al producto a entregar, mismo que asegura calidad y dedicación para su confiable implementación.

Respuesta:

11. El objetivo principal de un proceso ágil, es no hacer documentación de desarrollo.

Elija una;

Verdadero

Falso

12. Seleccione las políticas del desarrollo ágil según corresponda:

Seleccione una o más de una:

a. El control y seguimiento del proyecto se basará en los requisitos completados en cada iteración.

b. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.

c. Cada iteración deberá producir software con calidad de producción, probado, integrado, y documentado (funcional, técnica).

d. El proyecto deberá ser ejecutado en iteraciones incrementales con una demostración del producto al finalizar cada iteración

- e. Cada uno de los entregables, deberá contener scripts de pruebas unitarias, integrales, funcionales, etc; mediante la utilización de frameworks como junit, dbunit, mockObjects, etc.
- f. Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
- g. Cada requisito debe ser demostrable, permitiendo cómo comprobar con el cliente que el requisito está completado y que se cumplen sus expectativas.
- h. El software que funciona es la medida principal de progreso

13. El Compromiso, enfoque, apertura, transparencia, respeto y coraje son algunos de los valores ágiles.

Elija una;

- Verdadero
- Falso

14. Relacione según corresponda:

- | | |
|--|-----------------------------------|
| Se centra en los talentos y habilidades de los individuos, puesto que el proceso se ajusta a personas y equipos específicos. | Respuesta 1
Soft skills |
| Aunque los miembros del equipo ágil desempeñan tareas distintas y aportan habilidades al proyecto, todos deben tener una meta: entregar al cliente un incremento de trabajo de software dentro del tiempo establecido. | Respuesta 2
El desarrollo ágil |
| Extreme Programming (XP), Scrum, Adaptive Software Development (ASD), Crystal, DSDM, Feature Driven Development y Lean software development. | Respuesta 3
Agile Alliance, |
| Es el documento que resume la filosofía "ágil". | Respuesta 4
Hard Skills |
| Parte del Cociente Intelectual de una persona, son las exigencias profesionales de un trabajo y muchas otras actividades. | Respuesta 5
Enfoque común |



Es el conjunto de rasgos de personalidad, habilidades sociales, comunicación, lenguaje, hábitos personales, la amistad y el optimismo que caracteriza a las relaciones con otras personas. Respuesta 6
Manifiesto Ágil

En el contexto de un desarrollo ágil abarca un talento innato, habilidades específicas relacionadas con el software, y un conocimiento del proceso que el equipo haya elegido aplicar. Respuesta 7
Métodos ágiles

Organización dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. Respuesta 8
Competencia

15. Todas las metodologías de desarrollo ágil; minimiza riesgos desarrollando software en cortos lapsos de tiempo.

Elija una;

- Verdadero
- Falso

16. La metodología tradicional tiene como filosofía dar mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental.

Elija una;

- Verdadero
- Falso

17. Solo en desarrollo tradicional es necesario incorporar y aplicar políticas.

Elija una;

- Verdadero
- Falso

18. Es lo que necesitan los profesionales y emprendedores para poder desempeñar eficazmente su trabajo y llevar adelante con éxito los proyectos.





Respuesta:

19. Una de las características principales de las metodologías ágiles es entregar software funcionando en intervalos de 2-4 meses de tiempo. Es el menor tiempo posible entre entregas.

Elija una:

- Verdadero
- Falso



Cuestionario unidad 1 "Introducción a los procesos"

Marque con una "✓" en la columna SI, cuando el candidato respondió correctamente el reactivo y en la columna NO cuando su respuesta no sea la indicada en el presente documento.

Nombre del Alumno					
Valor 50 puntos		Tipo de	Respuesta		
Pregunta	Respuesta Correcta	Reactivo	SI	NO	Valor
1	Metodologías livianas	Pregunta Abierta			1
2	Agile Alliance	Pregunta Abierta			1
3	a,d,e,g,h	Opción múltiple			1
4	Manifiesto ágil	Pregunta abierta			1
5	Verdadero	Falso-Verdadero			1
6	Falso	Falso-Verdadero			1
7	Falso	Falso-Verdadero			1
8	Falso	Falso-Verdadero			1
9	Falso	Falso-Verdadero			1
10	Verdadero	Falso-Verdadero			1
11	Falso	Falso-Verdadero			1
12	2,5,7,6,4,1,8,3	Relacionales			8
13	Falso	Falso-Verdadero			1
14	Políticas	Pregunta abierta			1
15	Soft skills	Pregunta abierta			1
16	b,e,g,h	Opción múltiple			4
17	Falso	Falso-Verdadero			1
18	a,c,d,f	Opción múltiple			4
19	Valores ágiles	Pregunta abierta			1
T o t a l e a					32 ->100 ->50 Puntos CAL= ____ Valor: ____ puntos





Nivel de desempeño mínimo aceptable: 70/100

RETROALIMENTACION
ASPECTOS POSITIVOS:
AREAS DE OPORTUNIDAD:
SUGERENCIAS DE MEJORA:
NOMBRE Y FIRMA DE CONFORMIDAD DEL ALUMNO: <hr/>



ANEXO 3

Asignatura: "Metodologías ágiles de desarrollo de Software"

Nombre del alumno:		
Carrera:	CLAVE:	GRUPO:
No. Control:		Fecha:

Instrucciones para el evaluador: Marque con una "✓" en la columna SI, cuando el(los) alumno(s) muestre las evidencias de los apartados correspondientes y en la columna NO cuando no muestre las evidencias señaladas.

Lista de cotejo INVESTIGACION SOBRE PROBLEMA REAL

No.	Aspecto a evaluar	Valor	Cumple		Pta.
			SI	No	
1	Portada	5			
2	Introducción (descripción de la empresa seleccionada y mencionar el problema a analizar)	30			
3	Desarrollo (Presentar mediante el desarrollo de preguntas (al menos 5) previamente formuladas y aplicadas para obtener la información)	40			
4	Conclusiones (indicar si es factible desarrollar o no y porque)	15			
5	Presentación del documento (Justificar y formato)	5			
6	Sin errores ortográficos.	5			
		100	Total		

Nivel de desempeño mínimo aceptable: 80 pts.



RETROALIMENTACION
ASPECTOS POSITIVOS:
AREAS DE OPORTUNIDAD:
SUGERENCIAS DE MEJORA:
NOMBRE Y FIRMA DE CONFORMIDAD DEL ALUMNO: _____



ANEXO 4
Asignatura: "Metodologías ágiles de desarrollo de Software"

Nombre del alumno:			
Carrera:	CLAVE:	GRUPO:	
No. Control:		Fecha:	

Instrucciones para el evaluador: Marque con una "√" en la columna SI, cuando el(los) alumno(s) muestre las evidencias de los apartados correspondientes y en la columna NO cuando no muestre las evidencias señaladas.

Lista de cotejo INVESTIGACIÓN SOFTSKILLS

No.	Aspecto a evaluar	Valor	Cumple		Pts.
			SI	No	
1	Portada	5			
2	Introducción (Descripción de lo que son las softskills y de lo que significan para la empresa seleccionada)	20			
3	Desarrollo (presenta la descripción de las softskills que se emplean en la empresa y de qué manera se aplican)	40			
4	Conclusiones (opinión personal sobre la aplicación de estas habilidades en la empresa)	15			
5	Bibliografía. (Fuente confiable y formato Apa)	10			
6	Presentación del documento (Justificar y formato)	5			
7	Sin errores ortográficos.	5			
		100	Total		

Nivel de desempeño mínimo aceptable: 80 pts.



RETROALIMENTACION
ASPECTOS POSITIVOS:
AREAS DE OPORTUNIDAD:
SUGERENCIAS DE MEJORA:
NOMBRE Y FIRMA DE CONFORMIDAD DEL ALUMNO: _____



ANEXO 5

Asignatura: "Metodologías Ágiles de desarrollo de software"

Nombre del alumno:			
Carrera:	CLAVE:	GRUPO:	
No. Control:		Fecha:	

**Rúbrica Cuadro Comparativo
UNIDAD II: Tipos de Metodologías Ágiles, sus principios y características**

Aspecto	Excelente	Bueno	Suficiente	Insuficiente	Puntos obtenidos
Establece los elementos y las características a comparar	Identifica todas las metodologías ágiles, sus principios y características. Las características elegidas son suficientes y pertinentes	Incluye la mayoría de las metodologías ágiles, las características son suficientes para realizar una buena comparación	Incluye algunas metodologías ágiles (al menos 2), sin embargo las características son mínimas	Incluye solo una metodología ágil y no enuncia adecuadamente las características	
Puntaje	35%	30%	25%	0%	
Identifica las semejanzas y diferencias de las metodologías ágiles	Identifica de manera clara y precisa las semejanzas y diferencias de las metodologías a comparar	Identifica la mayor parte de las semejanzas y diferencias, entre las metodologías a comparar	Identifica varias de las semejanzas y diferencias, entre las metodologías a comparar	No identifica las semejanzas y diferencias de los modelos comparados	
PUNTAJE	30%	25%	20%	0%	



Representación esquemática de la información	La gráfica presenta las metodologías ágiles a comparar, principios y características y sus diferencias de forma clara y precisa	La gráfica que presenta muestra con cierta claridad y precisión varias metodologías ágiles, principios, características y establece diferencias	La gráfica elaborada presenta las metodologías solicitadas aunque no es del todo claro y preciso	La gráfica no representa esquemáticamente los elementos a ser comparados	
PUNTAJE	25%	20%	15%		
Ortografía y presentación	Excelente presentación sin errores ortográficos	Existen errores ortográficos (menos de 3)	Varios errores ortográficos (más de 3 pero menos de 5)	Muchos errores ortográficos (más de 5)	
PUNTAJE	10%	8%	5%	7%	
Calificación					





RETROALIMENTACION
ASPECTOS POSITIVOS:
AREAS DE OPORTUNIDAD:
SUGERENCIAS DE MEJORA:
NOMBRE Y FIRMA DE CONFORMIDAD DEL ALUMNO:



ANEXO 6

Asignatura: "Metodologías ágiles de desarrollo de Software"

Nombre del alumno:			
Carrera:	CLAVE:	GRUPO:	
No. Control:		Fecha:	

Rúbrica Debate

Criterios de Desempeño		Excelente	Bueno	Suficiente	Insuficiente	Puntos obtenidos
F O R M A	Argumentación	<ul style="list-style-type: none"> Los contenidos de la argumentación sobre el tema evidencian una investigación previa realizada en diversas fuentes de consulta. Expone las ideas principales y secundarias en torno al tema de debate. Identifica y articula puntos de desacuerdo que no son obvios sobre el tema, y que inciden en la discusión del mismo. 	<ul style="list-style-type: none"> Los contenidos de la argumentación sobre el tema son generales y evidencian la investigación en diversas fuentes de consulta. Expone las ideas principales y no identifica las ideas secundarias en torno al tema de debate. Identifica algunos puntos de desacuerdo que no son obvios sobre el 	<ul style="list-style-type: none"> Los contenidos de la argumentación sobre el tema son pobres y evidencian la investigación en algunas fuentes de consulta. Expone sólo ideas secundarias en torno al tema de debate. Identifica algún punto de desacuerdo sobre el tema que es obvio en el contenido de la discusión. 	<ul style="list-style-type: none"> Los contenidos de la argumentación sobre el tema son deficientes y evidencian la falta de investigación sobre el tema. Expone ideas no precisas sobre el tema. Ignora los puntos de desacuerdos implícitos y no 	



	cuenta las conclusiones de los demás.				
PUNTAJE	10%	8%	5%	0%	
MATERIAL DE APOYO (GRUPAL)	<ul style="list-style-type: none"> • Para la argumentación o desarrollo del debate los participantes muestran material de apoyo que presenta, al menos, las siguientes características: <ul style="list-style-type: none"> • Contiene información primordial para respaldar los puntos de vista o posiciones. • Las imágenes u objetos representan claramente la idea que se pretende reforzar. • Tiene una presentación ordenada, uniforme y es legible desde cualquier parte del salón. • Los textos no presentan errores ortográficos. 	<ul style="list-style-type: none"> • Para la argumentación o desarrollo del debate los participantes muestran material de apoyo que presenta, las siguientes características: <ul style="list-style-type: none"> • Contiene información primordial para respaldar los puntos de vista o posiciones • Las imágenes u objetos representan claramente la idea que se pretende reforzar. Además presenta alguna de las siguientes características. <ul style="list-style-type: none"> • Tiene una presentación ordenada, uniforme y es legible desde cualquier parte del salón. • Los textos si presentan errores ortográficos. 	<ul style="list-style-type: none"> • Para la argumentación o desarrollo del debate los participantes muestran material de apoyo que presenta, las siguientes características: <ul style="list-style-type: none"> • Contiene información primordial para respaldar los puntos de vista o posiciones. • Las imágenes u objetos representan claramente la idea que se pretende reforzar. • Tiene una presentación poco ordenada y uniforme. Aunque es legible desde cualquier parte del salón. • Los textos presentan muchos errores ortográficos. 	<ul style="list-style-type: none"> • Para la argumentación o desarrollo del debate los participantes muestran material de apoyo que presenta, las siguientes características: <ul style="list-style-type: none"> • Contiene información que respalda pocos de los puntos de vista o posiciones • Las imágenes u objetos representan vagamente la idea que se pretende reforzar. • Tiene una presentación sin orden y uniformidad, y no es legible desde cualquier parte del salón. • Los textos presentan bastantes errores ortográficos. 	
PUNTAJE	10%	8%	5%	2%	
ESTRUCTURA DEL DEBATE	En el desarrollo del debate se observan	En el desarrollo del debate se observan	En el desarrollo del debate se observan	En el desarrollo del debate se observa	



PROCESO DE ANÁLISIS	(Grupal)	<p>todos los siguientes puntos:</p> <ul style="list-style-type: none"> • Una distribución explícita de tiempos de participación para los individuos o grupos. • Respeto por los tiempos destinados tanto para el trabajo de reflexión como para el debate. • Un moderador designado que desempeña en forma adecuada su rol con el grupo. • Se manifiesta la organización de las distintas fases del debate: introducción, argumentación, refutación y conclusión. • Se establecieron previamente las reglas de participación de acuerdo a la estructura definida para el debate 	<p>todos los siguientes puntos:</p> <ul style="list-style-type: none"> • Una distribución explícita de tiempos de participación para los individuos o grupos. • Respeto por los tiempos destinados tanto para el trabajo de reflexión como para el debate. • Un moderador designado que desempeña en forma adecuada su rol con el grupo. • Se manifiesta la organización de las distintas fases del debate: introducción, argumentación, refutación y conclusión. 	<p>todos los siguientes puntos:</p> <ul style="list-style-type: none"> • Una distribución explícita de tiempos de participación para los individuos o grupos. • Respeto por los tiempos destinados tanto para el trabajo de reflexión como para el debate. • Un moderador designado que desempeña en forma adecuada su rol con el grupo. 	<p>claramente uno de los siguientes puntos:</p> <ul style="list-style-type: none"> • Una distribución explícita de tiempos de participación para los individuos o grupos • Respeto por los tiempos destinados tanto para el trabajo de reflexión como para el debate. • Sin moderador 	
	PUNTAJE	10%	8%	5%	3%	
		<ul style="list-style-type: none"> • Durante el debate se observa: • El equipo reflexiona sobre los tópicos tratados. • Emite respuestas que invitan al grupo a plantearse nuevos cuestionamientos. • Las respuestas dependen de las frases precedentes. • Las respuestas pueden modificar sus opiniones. 	<ul style="list-style-type: none"> • Durante el debate se observa: • El equipo reflexiona sobre los tópicos tratados. • Emite respuestas que invitan al grupo a plantearse nuevos cuestionamientos. • Las respuestas dependen de las frases 	<ul style="list-style-type: none"> • Durante el debate se observa: • El equipo reflexiona sobre los tópicos tratados. • Emite respuestas que invitan al grupo a plantearse nuevos cuestionamientos. • Las respuestas dependen de las frases precedentes. 	<ul style="list-style-type: none"> • Durante el debate se observa: • El equipo reflexiona sobre los tópicos tratados • Emite respuestas que invitan al grupo a plantearse nuevos cuestionamientos. • No hay respuestas congruentes solo 	



	•Las respuestas manifiestan congruencia y objetividad	precedentes, •Las respuestas pueden modificar sus opiniones. •Las respuestas son incongruentes y sin objetividad		exposición y muy poca participación de los temas.	
PUNTAJE	20%	15%	10%	5%	
CONCLUSIÓN Y SÍNTESIS	•El equipo realiza una exposición breve, escrita u oral, que a modo de resumen contiene un conjunto de ideas fundamentales, • Las ideas están relacionadas con el/los tópicos e ideas que estaban dispersas. •La conclusión se da con una sentencia que resuelve el fondo del asunto. •La conclusión da la certeza de que se han resuelto todos los tópicos de buena manera.	•El equipo realiza una exposición breve, escrita u oral, que a modo de resumen contiene un conjunto de ideas fundamentales, • Las ideas están relacionadas con el/los tópicos e ideas que estaban dispersas. •La conclusión se da con una sentencia que resuelve el fondo del asunto. •La conclusión genera incertidumbre de la solución de todos los tópicos.	•El equipo realiza una exposición breve, escrita u oral, que a modo de resumen contiene un conjunto de ideas fundamentales, • Las ideas están relacionadas con el/los tópicos e ideas que estaban dispersas. •La conclusión se da con una sentencia que resuelve el fondo del asunto. •No hay una conclusión clara	•El equipo realiza una exposición breve, escrita u oral, que a modo de resumen contiene un conjunto de ideas fundamentales, • Las ideas están relacionadas con el/los tópicos e ideas que estaban dispersas. •Sin conclusión.	
PUNTAJE	20%	15%	10%	5%	
Calificación					





RETROALIMENTACION
ASPECTOS POSITIVOS:
AREAS DE OPORTUNIDAD:
SUGERENCIAS DE MEJORA:
NOMBRE Y FIRMA DE CONFORMIDAD DEL ALUMNO: <hr/>



ANEXO 7
Asignatura: "Metodologías Ágiles de Desarrollo de Software"

EQUIPO NÚMERO		
GRUPO:		
Fecha:	Participaron todos?	Si la respuesta es no, quien falló?

Instrucciones para el evaluador: Marque con una "✓" en la columna SI, cuando el(los) alumno(s) muestre las evidencias de los apartados correspondientes y en la columna NO cuando no muestre las evidencias señaladas.

Lista de cotejo INVESTIGACIÓN SOBRE UNA METODOLOGIA ÁGIL EN UNA EMPRESA

No.	Aspecto a evaluar	Valor	Cumple		Pts.
			SI	No	
1	Portada	5			
2	Introducción (Definir lo que son las metodologías ágiles y la razón de esta investigación)	20			
3	Desarrollo (Describir cómo la empresa utiliza las metodologías ágiles y centrarse de preferencia en una de ellas, detallando su forma de uso)	40			
4	Conclusiones (Expresar la opinión por equipo sobre su experiencia al realizar esta visita)	15			
5	Bibliografía. (Fuente confiable y formato Apa)	10			
6	Presentación del documento (Justificar y formato)	5			
7	Sin errores ortográficos.	5			
		100	Total		

Nivel de desempeño mínimo aceptable: 80 pts.



RETROALIMENTACION	
ASPECTOS POSITIVOS:	
AREAS DE OPORTUNIDAD:	
SUGERENCIAS DE MEJORA:	
NOMBRE Y FIRMA DE CONFORMIDAD DE LOS ESTUDIANTES:	
_____	_____
_____	_____
_____	_____



ANEXO 8
Cuestionario Unidad 2. Tipos de Metodologías Ágiles, sus principios y características

Asignatura: "Metodologías Ágiles de Desarrollo de Software"

Nombre del alumno:			
Carrera:	CLAVE:	GRUPO:	
No. Control:		Fecha:	

Con el presente instrumento se pretende evaluar los conocimientos necesarios para acreditar la Unidad 2.

Instrucciones:

1. El cuestionario se hará en la plataforma de apoyo en línea www.materias.ith.mx/moodle
2. El cuestionario se aplicará en el laboratorio de sistemas y computación en el horario de clases.
3. El maestro proporcionará la contraseña y acceso en la fecha y hora de aplicación indicado en el grupo.
4. Para el llenado de la aplicación en línea el maestro recomienda que se lean con atención las instrucciones correspondientes a cada sección o reactivo.
5. Conteste las preguntas que se presentan en esta evaluación escribiendo su respuesta directamente en el cuestionario.
6. En caso de que tenga cualquier duda sobre la evaluación, pregunte al maestro antes de contestar la respuesta en el cuestionario.
7. Se estima que el tiempo aproximado para contestar la evaluación es de 30 minutos.



Cuestionario

Pregunta 1

"Un proceso ligero, de bajo riesgo, flexible, predecible, científico y divertido de desarrollar software". Es la definición que Kent Beck da a...???

Seleccione una:

- a. Extreme Programming
- b. Scrum
- c. Kanban

Pregunta 2

Se basa en la utilización repetitiva de código, para lo cual se establecen patrones permitiendo mayor flexibilidad al cambio.

Seleccione una:

- a. Reprogramación
- b. Refabricación
- c. pruebas

Pregunta 3

Valores de XP

Seleccione una:

- a. SIMPLICIDAD, COMUNICACIÓN, RESPETO
- b. SIMPLICIDAD, COMUNICACIÓN, CORAJE

Pregunta 4

Es una metodología de trabajo de origen japonés, que permite gestionar de manera más eficiente y dinámica el ciclo de vida de los proyectos, ya que se trabaja de forma más directa y visual, simplificando así la documentación y gestión.

Seleccione una:

- a. KANBAN
- b. PUA
- c. SCRUM



Pregunta 6

Estas pruebas son realizadas por los clientes y se enfoca en las características generales del sistema.
Seleccione una:

- a. DE UNIDAD
- b. DE ACEPTACIÓN

Pregunta 6

Es un proceso ágil que nos permite centrarnos en ofrecer el más alto valor de negocio en el menor tiempo. Nos permite rápidamente y en repetidas ocasiones inspeccionar software real de trabajo (cada dos semanas o un mes).
Seleccione una:

- a. KANBAN
- b. SCRUM
- c. MSDS

Pregunta 7

Provee un framework para el desarrollo ágil de software, apoyado por su continua implicación del usuario en un desarrollo iterativo y creciente

Seleccione una:

- a. KANBAN
- b. SCRUM
- c. MSDS

Pregunta 8

Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo. El factor más importante es la comunicación.

Seleccione una:

- a. KANBAN
- b. CRYSTAL
- c. MSDS



Pregunta 9

Aplica técnicas ágiles incluyendo desarrollo orientado a pruebas, modelado ágil, gestión de cambios ágil y refactorización de bases de datos para mejorar la productividad

Seleccione una:

- a. EXTREME PROGRAMMING
- b. CRYSTAL
- c. PROCESO UNIFICADO ÁGIL

Pregunta 10

El origen de esta metodología debemos buscarlo en los procesos de producción "just-in-time" (JIT) ideados por Toyota

Seleccione una:

- a. KANBAN
- b. CRYSTAL
- c. SCRUM



Cuestionario unidad 2 "Tipos de Metodologías Ágiles, sus principios y características"

Marque con una "SI" en la columna SI, cuando el candidato respondió correctamente el reactivo y en la columna NO cuando su respuesta no sea la indicada en el presente documento.

Nombre del Alumno					
		Tipo de	Respuesta		
Pregunta	Respuesta Correcta	Reactivo	SI	NO	Valor
1	EXTREME PROGRAMMING	Opción múltiple			10
2	REFABRICACIÓN	Opción múltiple			10
3	INCISO b)	Opción múltiple			10
4	KANBAN	Opción múltiple			10
5	DE ACEPTACION	Opción múltiple			10
6	SCRUM	Opción múltiple			10
7	MODEO	Opción múltiple			10
8	CRYSTAL	Opción múltiple			10
9	PROCESO UNIFICADO ÁGIL	Opción múltiple			10
10	KANBAN	Opción múltiple			10
Totales					CAL= ____ Valor: ____ puntos

Nivel de desempeño mínimo aceptable: 70 Puntos





RETROALIMENTACION
ASPECTOS POSITIVOS:
AREAS DE OPORTUNIDAD:
SUGERENCIAS DE MEJORA:
NOMBRE Y FIRMA DE CONFORMIDAD DEL ALUMNO: <hr/>



ANEXO 9
Asignatura: "Metodologías ágiles de desarrollo de Software"

Nombre del alumno:		
Carrera:	CLAVE:	GRUPO:
No. Control:		Fecha:

Instrucciones para el evaluador: Marque con una "✓" en la columna SI, cuando el(los) alumno(s) muestre las evidencias de los apartados correspondientes y en la columna NO cuando no muestre las evidencias señaladas.

Lista de cotejo Mapa Conceptual del Framework de Scrum

No.	Aspecto a evaluar	Valor	Cumple		Pts.
			SI	No	
1	Portada	5			
2	Contenido	70			
	➤ Identifica y anota los conceptos principales	35			
	➤ Identifica conceptos secundarios	10			
	➤ Jerarquiza en orden de importancia	15			
	➤ Hace los enlaces adecuados utilizando preposiciones conectivas en forma correcta	10			
7	Presentación del documento limpio y en tiempo	15			
8	Sin errores ortográficos.	10			
		100	Total		

Nivel de desempeño mínimo aceptable: 80 pts.





RETROALIMENTACION
ASPECTOS POSITIVOS:
AREAS DE OPORTUNIDAD:
BUGERENCIAS DE MEJORA:
NOMBRE Y FIRMA DE CONFORMIDAD DEL ALUMNO: _____



ANEXO 10
Asignatura: "Metodologías Ágiles de desarrollo de software"

EQUIPO NO.		
GRUPO		
Fecha	Participaron todos?	Si la respuesta es no, quien falló?

Instrucciones para el evaluador: Marque con una "X" en la columna SI, cuando el(los) alumno(s) muestre las evidencias de los apartados correspondientes y en la columna NO cuando no muestre las evidencias señaladas.

Lista de cotejo REPORTE FINAL DE PROYECTO INTEGRADOR

No.	Aspecto a evaluar	Valor	Cumple		Pts.
			Si	No	
1	Portada	3			
2	Introducción (Describe lo que es son los procesos ágiles y las metodologías que siguen este concepto, importancia y características)	3			
3	Descripción General del Proyecto				
	<ul style="list-style-type: none"> • Objetivo General y objetivos específicos • Ambito del proyecto(empresa a la que se le desarrolla y departamento al fuera el caso) 	5			
4	Desarrollo del proyecto				
	<ul style="list-style-type: none"> • Organización del equipo (roles desempeñados por cada uno) 	5			
	<ul style="list-style-type: none"> • Producto backlog 	15			
	<ul style="list-style-type: none"> • Metodología de trabajo <ul style="list-style-type: none"> ○ Sprint 1n (Objetivo del Sprint) ○ Sprint backlog (tarear) ○ Pizarra de tareas: Presentar historias de usuario (user stories) y tareas, donde se vea el avance y término de las mismas, evidenciando que tareas hizo cada quien ○ Planeado vs Real (tiempo planeado vs tiempo real en horas), explicando las causas de la diferencia ○ Resultado de las pruebas ○ Resultados y evidencias de reuniones Diarias y de Revisión ○ Resultados y Ajustes a partir de las reuniones de retrospectiva 	5			
		5			
		5			
		5			
		5			
		5			
		5			



	• Tecnologías, Técnicas y Herramientas utilizadas	5			
6	Análisis del impacto del proyecto en la Organización	5			
6	Conclusiones • Referente a lo realizado: Conclusiones de aplicar Scrum en el proyecto; <Comunicación del equipo>; <Recomendaciones resultado de su experiencia con Scrum>	5			
7	Presentación del documento (Justificar y formato)	3			
8	Sin errores ortográficos.	3			
8	Entrega en tiempo (fecha y hora)	3			
		100	Total		

Nota : Todo el punto de Metodología del trabajo se repite según el número de sprint

Nivel de desempeño mínimo aceptable: 80 pts.

RETROALIMENTACION
ASPECTOS POSITIVOS:
AREAS DE OPORTUNIDAD:
SUGERENCIAS DE MEJORA:





NOMBRE Y FIRMA DE CONFORMIDAD DE LOS ESTUDIANTES:

_____	_____
_____	_____
_____	_____
_____	_____



ANEXO 11

Asignatura: "Metodologías ágiles de desarrollo de Software"

Nombre del alumno:			
Camera:	CLAVE:	GRUPO:	
No. Control:	Fecha:		

Gua de observación de Exposición final

Instrucciones para el llenado del instrumento:

Observe cuidadosamente la ejecución de las actividades que se enuncian y marque con una "v" en la columna SI cuando el candidato cumpla con el desempeño correspondiente y en la columna NO cuando no realice las actividades señaladas.

Presentación del Tema:	
Unidad no.	

No.	Al inicio de la sesión:	Valor	SI	NO	1	2	3	4	5
1	Presentación ante el grupo	5							
2	Presentación del tema	5							
Durante la sesión:									
3	Utilización de material de apoyo	10							
4	Desarrollo completo de los temas	15							
5	Información complementaria de los temas	5							
6	Mantiene claridad en los temas	10							
7	Mantiene contacto visual con el grupo	5							
8	Interactúa con el grupo	10							
9	Mantiene la atención del grupo en el tema	10							
Cierre de la sesión:									
10	Presenta Resumen o Conclusiones del tema.	15							
11	Responde a las preguntas de forma amable	5							
12	Cumple con el tiempo planeado para la sesión	5							
Total por participante									
							Puntuación Total		

Nota: En las columnas enumeradas se anotará el puntaje en la casilla del participante correspondiente, teniendo como máximo de puntos, el valor indicado.

Nivel de desempeño mínimo aceptable: 80 pts.



RETROALIMENTACION						
ASPECTOS POSITIVOS:						
AREAS DE OPORTUNIDAD:						
SUGERENCIAS DE MEJORA:						
NOMBRE Y FIRMA DE CONFORMIDAD DEL ALUMNO:						
<table style="width: 100%; border: none;"> <tr> <td style="width: 50%; border: none;"> 1. _____ </td> <td style="width: 50%; border: none;"> 4. _____ </td> </tr> <tr> <td style="border: none;"> 2. _____ </td> <td style="border: none;"> 5. _____ </td> </tr> <tr> <td style="border: none;"> 3. _____ </td> <td style="border: none;"></td> </tr> </table>	1. _____	4. _____	2. _____	5. _____	3. _____	
1. _____	4. _____					
2. _____	5. _____					
3. _____						

- | | |
|--|-------|
| 1 *integrante uno del equipo | _____ |
| 2 *integrante dos del equipo | _____ |
| 3 *integrante tres del equipo | _____ |
| 4 *integrante cuatro del equipo | _____ |
| 5 *integrante cinco del equipo | _____ |



ANEXO 12
Asignatura: "Metodologías Ágiles de desarrollo de software"

EQUIPO NO.		
GRUPO		
Fecha:	Participaron todos?	Si la respuesta es no, quien falló?

Instrucciones para el evaluador: Marque con una "X" en la columna SI, cuando el(los) alumno(s) muestre las evidencias de los apartados correspondientes y en la columna NO cuando no muestre las evidencias señaladas.

Importante: Se utilizará un formato de lista de cotejo en cada revisión. El equipo deberá traerlo impreso

Lista de cotejo REVISIÓN DE SEGUIMIENTO AL PROYECTO INTEGRADOR

No.	Aspecto a evaluar	Valor	Cumple		Pts.
			Si	No	
1	Introducción (Describe lo que es son los procesos ágiles y las metodologías que siguen este concepto, importancia y características)	5			
2	Descripción General del Proyecto				
	▪ Objetivo General y objetivos específicos	10			
	▪ Ambito del proyecto(empresa a la que se le desarrolla y departamento si fuera el caso)	5			
8	Desarrollo del proyecto				
	▪ Organización del equipo (roles desempeñados por cada uno)	5			
	▪ Producto backlog	10			
	▪ Metodología de trabajo				
	○ Sprint 1:n (Objetivo del Sprint)	5			
	○ Sprint backlog (tareas)	10			
	○ Pizarra de tareas: Presentar historias de usuario (user stories) y tareas, donde se vea el avance y término de las mismas, evidenciando que tareas hizo cada quien	10			
	○ Planeado vs Real (tiempo planeado vs tiempo real en horas), explicando las causas de la diferencia	5			
	○ Resultado de las pruebas	5			
	○ Resultados y evidencias de reuniones Diarias y de Revisión	10			
	○ Resultados y Ajustes a partir de las reuniones de retrospectiva	10			



4	Conclusiones <ul style="list-style-type: none"> Referente a lo realizado Conclusiones de aplicar Scrum en el proyecto, <Comunicación del equipo> <Recomendaciones resultado de su experiencia con Scrum>	5			
6	Entrega en tiempo (fecha y hora)	5			
		100	Total		

Nota : *Todos los puntos que abarca la Metodología del trabajo se repite según el número de sprint.

Nivel de desempeño mínimo aceptable: 80 pts.

RETROALIMENTACION
ASPECTOS POSITIVOS:
AREAS DE OPORTUNIDAD:
SUGERENCIAS DE MEJORA:
NOMBRE Y FIRMA DE CONFORMIDAD DE LOS ESTUDIANTES: <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div style="width: 45%; border-bottom: 1px solid black;"></div> <div style="width: 45%; border-bottom: 1px solid black;"></div> </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div style="width: 45%; border-bottom: 1px solid black;"></div> <div style="width: 45%; border-bottom: 1px solid black;"></div> </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div style="width: 45%; border-bottom: 1px solid black;"></div> <div style="width: 45%; border-bottom: 1px solid black;"></div> </div>



ANEXO 13

Asignatura: "Metodologías Ágiles de desarrollo de software"

EQUIPO NO.:		
TEMA A REVISIÓN		
Grupo:		Fecha:

Guía de observación Revisión de Seguimiento a Proyecto Integrador

Instrucciones para el llenado del instrumento:

Observe cuidadosamente la ejecución de las actividades que se enuncian y marque con una "✓" en la columna SI cuando el candidato cumpla con el desempeño correspondiente y en la columna NO cuando no realice las actividades señaladas.

Sesión no.	
Observaciones	

No.	Asistencia	Valor	1	2	3	4	5
1	Asiste puntualmente a la Sesión	20					
	Durante la sesión:						
	Está atento a las indicaciones y observaciones del docente	15					
	Expresa corporalmente tener conocimiento de lo que se habla	10					
	Participa activamente con preguntas o respuestas	20					
	Busca la participación de sus compañeros mostrando trabajo de equipo	10					
	Se expresa de manera adecuada	10					
	Toma nota de las indicaciones y observaciones hechas por el docente	15					
	Total por participante						
			Puntuación Total				

Nivel de desempeño mínimo aceptable: 80 pts.



RETROALIMENTACION
ASPECTOS POSITIVOS:
AREAS DE OPORTUNIDAD:
SUGERENCIAS DE MEJORA:

- 1 *Integrante uno del equipo _____
- 2 *Integrante dos del equipo _____
- 3 *Integrante tres del equipo _____
- 4 *Integrante cuatro del equipo _____
- 5 *Integrante cinco del equipo _____



ANEXO 14
Asignatura: "Metodologías ágiles de desarrollo de Software"

Nombre del alumno:		
Carrera:	CLAVE:	GRUPO:
No. Control:		Fecha:

Instrucciones para el evaluador: Marque con una "V" en la columna SI, cuando el(los) alumno(s) muestre las evidencias de los apartados correspondientes y en la columna NO cuando no muestre las evidencias señaladas.

Lista de cotejo INVESTIGACIÓN SOBRE DIFERENTES TABLONES DE TAREA

No.	Aspecto a evaluar	Valor	Cumple		Pts.
			SI	No	
1	Portada	5			
2	Introducción (Descripción de lo que son los tablon de tareas)	20			
3	Desarrollo (presenta diferentes ejemplos de tablón, tablero o pizama de tareas de scrum dando una breve descripción de las características de cada una)	40			
4	Conclusiones (opinión personal sobre lo aprendido)	10			
5	Bibliografía. (Fuente confiable y formato Apa)	5			
6	Presentación del documento (Justificar y formato)	5			
7	Sin errores ortográficos.	10			
8	Entrega en fecha y hora indicada	5			
		100	Total		

Nivel de desempeño mínimo aceptable: 80 pts.



RETROALIMENTACION
ASPECTOS POSITIVOS:
AREAS DE OPORTUNIDAD:
SUGERENCIAS DE MEJORA:
NOMBRE Y FIRMA DE CONFORMIDAD DEL ALUMNO: <hr/>



ANEXO 15

Asignatura: "Metodologías ágiles de desarrollo de Software"

Nombre del alumno:		
Carrera:	CLAVE:	GRUPO:
No. Control:		Fecha:

Instrucciones para el evaluador: Marque con una "v" en la columna SI, cuando el(los) alumno(s) muestre las evidencias de los apartados correspondientes y en la columna NO cuando no muestre las evidencias señaladas.

Lista de cotejo INVESTIGACIÓN SOBRE DIFERENTES HERRAMIENTAS PARA APOYO DE TRABAJO CON SCRUM

No.	Aspecto a evaluar	Valor	Cumple		Pts.
			SI	No	
1	Portada	5			
2	Índice de contenido	5			
3	Enlistar diferentes herramientas debiendo presentar funciones, características, ventajas y desventajas de cada una. (*Deberán ser al menos 5 herramientas como requisito primordial para la evaluación de este trabajo)	40			
4	Conclusiones (opinión personal sobre lo aprendido)	15			
5	Bibliografía. (Fuente confiable y formato Apa)	10			
6	Presentación del documento (Justificar y formato)	5			
7	Sin errores ortográficos.	10			
8	Entrega en fecha y hora indicada	10			
		100	Total		

Nivel de desempeño mínimo aceptable: 80 pts.



RETROALIMENTACION
ASPECTOS POSITIVOS:
AREAS DE OPORTUNIDAD:
SUGERENCIAS DE MEJORA:
NOMBRE Y FIRMA DE CONFORMIDAD DEL ALUMNO: <hr/>



ANEXO 16
Cuestionario Unidad 3. Metodología Scrum
Asignatura: "Metodologías Ágiles de Desarrollo de Software"

Nombre del alumno:			
Carrera:	CLAVE:	GRUPO:	
No. Control:		Fecha:	

Con el presente instrumento se pretende evaluar los conocimientos necesarios para acreditar la Unidad 3.

Instrucciones:

1. El cuestionario se hará en la plataforma de apoyo en línea www.materias.ith.mx/moodle
2. El cuestionario se aplicará en el laboratorio de sistemas y computación en el horario de clases.
3. El maestro proporcionará la contraseña y acceso en la fecha y hora de aplicación indicado en el grupo.
4. Para el llenado de la aplicación en línea el maestro recomienda que se lean con atención las instrucciones correspondientes a cada sección o reactivo.
5. Conteste las preguntas que se presentan en esta evaluación escribiendo su respuesta directamente en el cuestionario.
6. En caso de que tenga cualquier duda sobre la evaluación, pregunte al maestro antes de contestar la respuesta en el cuestionario.
7. Se estima que el tiempo aproximado para contestar la evaluación es de 30 minutos.



Cuestionario

Pregunta 1

EN LA REUNIÓN DE RETROSPECTIVA SE UTILIZAN ESTAS PREGUNTAS.....

Seleccione una:

- a. ¿Qué hiciste ayer?, ¿Que vas a hacer hoy?, ¿Hay obstáculos en tu camino?
- b. ¿Comenzar a hacer?, ¿Dejar de hacer?, ¿Continuar haciendo?

Pregunta 2

- No para la solución de problemas
- Todo el mundo está invitado
- Sólo los miembros del equipo, ScrumMaster y Product Owner, pueden hablar

Respuesta:

Pregunta 3

- Es quien
- Define las funcionalidades del producto
- Ajusta funcionalidades y prioridades en cada iteración si es necesario
- Es responsable por la rentabilidad del producto

Seleccione una:

- a. SCRUM MASTER
- b. PRODUCT OWNER
- c. TEAM



Pregunta 4

«En que reunión el equipo presenta lo realizado durante el sprint?»

Respuesta:

[Siguiente](#)

Pregunta 5

Es quien....

«Representa a la gestión del proyecto

«Remueve Impedimentos

«Permite la estrecha cooperación en todos los roles y funciones

Seleccione una:

a. SCRUM MASTER

b. PRODUCTO OWNER

c. TEAM

Pregunta 6

ROLES EN SCRUM

Seleccione una:

a. PRODUCT OWNER, SCRUM MASTER, TEAM

b. PRODUCT BACKLOG, SPRINT BACKLOG, BURNDOWN CHART

Pregunta 7

«Es una lista de todos los trabajos deseados en el proyecto.....»

Respuesta:



Pregunta 8

• Permite visualizar como las historias de usuario son divididas en tareas y de esta manera definir el sprint backlog.

Respuesta:

Pregunta 9

• involucra a todos los miembros de un equipo ágil multifuncional, en el cual el rol del _____ es el de un experto multifuncional

Respuesta:

Pregunta 10

Se hacen "durante" el desarrollo y no después del desarrollo como en métodos convencionales.
Seleccione una:

* Respuesta:



Cuestionario unidad 3 "Metodología Scrum"

Marque con una "SI" en la columna SI, cuando el candidato respondió correctamente el reactivo y en la columna NO cuando su respuesta no sea la indicada en el presente documento.

Nombre del Alumno					
		Tipo de	Respuesta		
Pregunta	Respuesta Correcta	Reactivo	SI	NO	Valor
1	Inciso b)	Opción múltiple			10
2	Daily Scrum o reunión diaria	Opción múltiple			10
3	Inciso b)	Opción múltiple			10
4	Sprint Review	Respuesta corta			10
5	Scrum Master	Opción múltiple			10
6	Inciso a)	Opción múltiple			10
7	Product backlog	Respuesta corta			10
8	Tablón de tareas	Respuesta corta			10
9	Tester	Respuesta corta			10
10	Pruebas	Respuesta corta			10
Totales					CAL= ____ Valor: ____ puntos

Nivel de desempeño mínimo aceptable: 70 Puntos



RETROALIMENTACION
ASPECTOS POSITIVOS:
AREAS DE OPORTUNIDAD:
SUGERENCIAS DE MEJORA:
NOMBRE Y FIRMA DE CONFORMIDAD DEL ALUMNO: <hr/>

