

Tarea 3

Ciclo de vida del software (Relación 2)

1.- ¿Qué cuatro principios rigen el desarrollo ágil expresados en el Manifiesto Ágil?

- Individuos e interacciones sobre procesos y herramientas
- Software funcionando sobre documentación extensiva
- Colaboración con el cliente sobre negociación contractual
- Respuesta ante el cambio sobre seguir un plan

VALORES ÁGILES

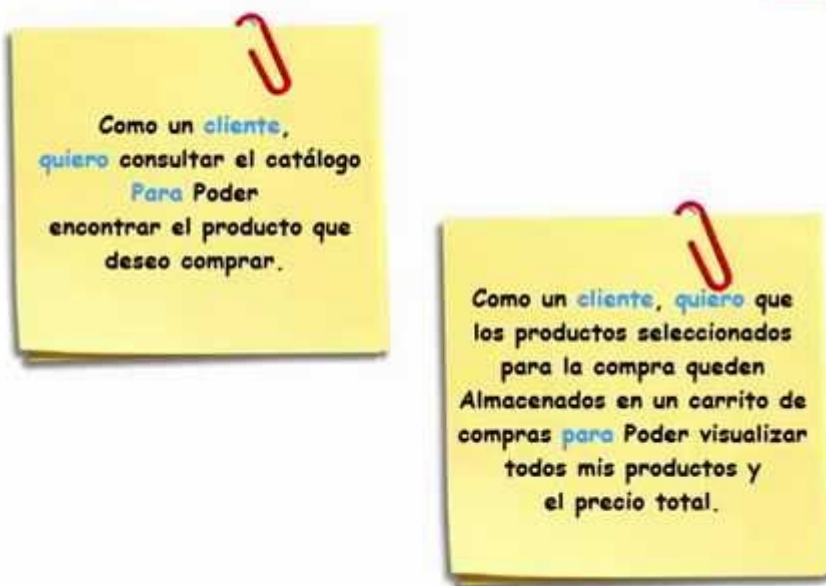


2.- ¿Qué es una historia de usuario? Consulta el siguiente enlace:

Una historia de usuario es una breve descripción de una funcionalidad del sistema desde el punto de vista del usuario final. Está escrita en un lenguaje sencillo y comprensible para cualquier persona. Las historias de usuario son utilizadas en las metodologías de desarrollo ágiles para la especificación de requisitos. Deben regirse a una serie de características que son las siguientes:

1. Independientes unas de otras : Cada tarea que necesitamos hacer debe poder hacerse por sí sola.

2. Negociables: Las tareas no son reglas estrictas. Hablar con los usuarios ayuda a entender qué necesitan realmente
3. Valoradas por los clientes o usuarios: Cada tarea debe ser importante para alguien, ya sea para los clientes o los usuarios finales.
4. Estimables: Hablar sobre las tareas nos ayuda a saber cuánto tiempo tomará hacer cada una de manera que podremos hacer una estimación de cuánto tiempo nos puede tomar un proyecto.
5. Pequeñas: Las tareas no deben ser demasiado largas. Si son muy largas, es difícil saber cuánto tiempo tomarán.
6. Verificables: Las tareas deben ser cosas que se pueden comprobar y probar, especialmente en términos de funciones.



3.- Haz un resumen sobre qué se entiende por Lean software y qué principios lo rigen.

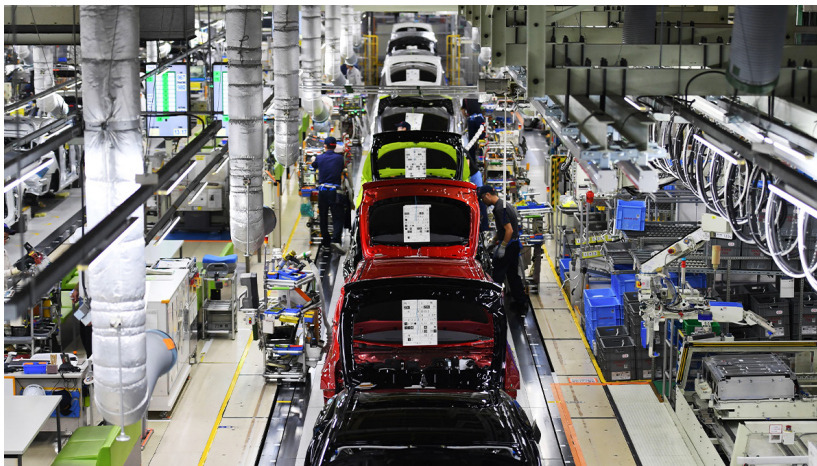
Consulta el siguiente enlace:

Es una metodología de desarrollo de software originado en el Sistema de Producción de Toyota que realiza una traducción de los principios y las prácticas de la forma de producir lean, hacia el área del desarrollo de software.

El desarrollo lean puede resumirse en siete principios.

1. Eliminar el desperdicio: Eliminar cualquier actividad o proceso que no agregue valor al cliente final.

2. Amplificar el aprendizaje: Aprender de los errores y adaptarse rápidamente es fundamental para mejorar el proceso y el producto.
3. Decidir lo más tarde posible: No tomar decisiones prematuras que no sean absolutamente necesarias. Retrasar las decisiones hasta que se tengan datos y se pueda tomar una decisión certera.
4. Entregar tan rápido como sea posible: Reducir el tiempo entre la concepción de una idea y su implementación.
5. Capacitar al equipo: proporcionar motivación y un propósito superior para el cual trabajar: un objetivo alcanzable dentro de la realidad con la garantía de que el equipo puede elegir sus propios compromisos. Los desarrolladores deberían tener acceso a los clientes; el jefe de equipo debe proporcionar apoyo y ayuda en situaciones difíciles, así como asegurarse de que el escepticismo no arruine el espíritu de equipo.
6. Construir integridad intrínseca: Enfocarse en la calidad desde el principio y asegurarse de que cada parte del proceso contribuya a la integridad del producto final. Esto implica pruebas frecuentes, revisión continua y mejora constante del proceso.
7. Véase todo el conjunto: Ver el proceso de desarrollo como un sistema completo y optimizar todo el sistema en lugar de partes individuales. Esto incluye la colaboración efectiva entre equipos, departamentos y partes interesadas para maximizar el valor entregado al cliente.





4. KANBAN. Estudia las ventajas e inconvenientes de tener una pizarra web digital para la metodología Kanban. Puedes consultar los siguientes enlaces:

Algunas de las ventajas de usar una pizarra kanban son:

- Es una herramienta visual que nos permite ver de manera clara y en tiempo real el proceso de trabajo.
- Kanban es altamente adaptable a diferentes tipos de proyectos. Los equipos pueden introducir cambios y ajustes en el proceso sin interrupciones drásticas.
- Los tableros Kanban eliminan en gran medida la necesidad de reuniones de estado al transmitir información de estado a través de la información y la posición de la tarjeta.
- Identificación rápida de problemas, una pizarra kanban permite la identificación de problemas de manera rápida gracias a que si se invierte más tiempo del estimado en una tarjeta, el equipo de trabajo podrá tomar medidas correctivas para mantener el flujo de trabajo.
- Al visualizar el trabajo y alentar la transparencia, Kanban fomenta la colaboración entre los miembros del equipo.

Algunas desventajas de usar kanban son:

- En equipos grandes o con proyectos complejos, el tablero Kanban puede conllevar a una sobrecarga visual.

- Kanban no proporciona roles específicos, pudiendo llevar a una falta de estructura, especialmente para equipos nuevos en Kanban, y dificultando la organización y la toma de decisiones.
- Kanban se centra en la gestión del trabajo a corto plazo y puede no ser óptimo para proyectos que requieren una planificación a largo plazo

En conclusión la metodología Kanban es muy útil y ofrece grandes ventajas aunque no es aplicable a todos los proyectos

5. KANBAN. Haz un resumen de la metodología Kanban e indica sus diferencias frente a SCRUM. Puedes consultar el siguiente enlace:

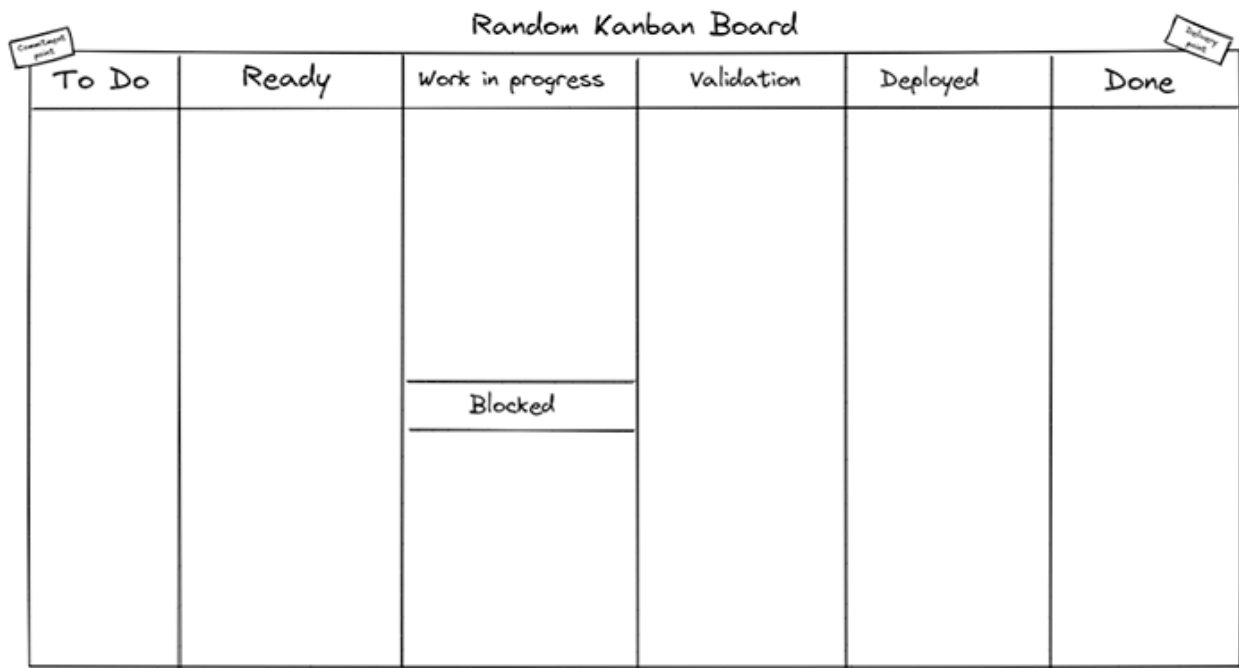
Kanban es una metodología ágil que procede del japonés que significa letrero o señal visual, esta metodología deriva de Lean manufacturing aplicado en la producción de Toyota. La idea general es que todos puedan ver el proceso de trabajo en cada uno de los procesos de producción definidos por el equipo. En Kanban el trabajo se visualiza en un tablero físico o digital. Este tablero está dividido en columnas que representan diferentes etapas del proceso de trabajo, como: por hacer, en proceso y terminado.

Los elementos de una pizarra kanban son:

- Carriles, que representan pasos definidos en el proceso.
- Tarjetas, que representan elementos de trabajo que avanzan a lo largo del proceso.

Kanban es una forma de organizar el trabajo visualmente. Los equipos que usan Kanban limitan cuántas tareas pueden hacer al mismo tiempo y se esfuerzan por trabajar de manera más eficiente. Usan un tablero para ver y organizar el trabajo, y siempre están buscando maneras de hacerlo mejor.

Por otro lado, Scrum es otra forma de gestionar el trabajo. Los equipos de Scrum se comprometen a hacer ciertas tareas en un tiempo específico llamado "sprint". Su objetivo es aprender rápido y mejorar escuchando a los clientes. Tienen roles, documentos y reuniones específicas para ayudarles a mantenerse en el camino correcto.



6. SCRUM. Explica cómo funciona Scrum. Consulta los siguientes enlaces:

Scrum es una metodología de trabajo ágil en la que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto.

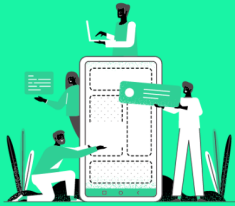
En Scrum un proyecto se ejecuta en ciclos temporales cortos y de duración fija denominados sprints que suelen durar entre una y cuatro semanas. Cada sprint tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

ROLES DE SCRUM



SCRUM MASTER

Lidera y sirve a los equipos en la gestión ágil.
Promueve la autoorganización y la responsabilidad.
Soluciona o ayuda a solucionar impedimentos.



DEVELOPMENT TEAM

Incluye todas las habilidades necesarias para crear funcionalidades incrementales de software acabado y entregable.
Los miembros son autoorganizados y responsables.
No existen roles ni subequipos.



PRODUCT OWNER

Maximizar el valor del trabajo del equipo es su responsabilidad.
Gran conocedor del negocio.
Gestiona el Product Backlog.
Conoce métricas cuantitativas y cualitativas de negocio para medir y validar el impacto o valor, incluyendo el ROI.

En un equipo de trabajo existen los siguientes roles:

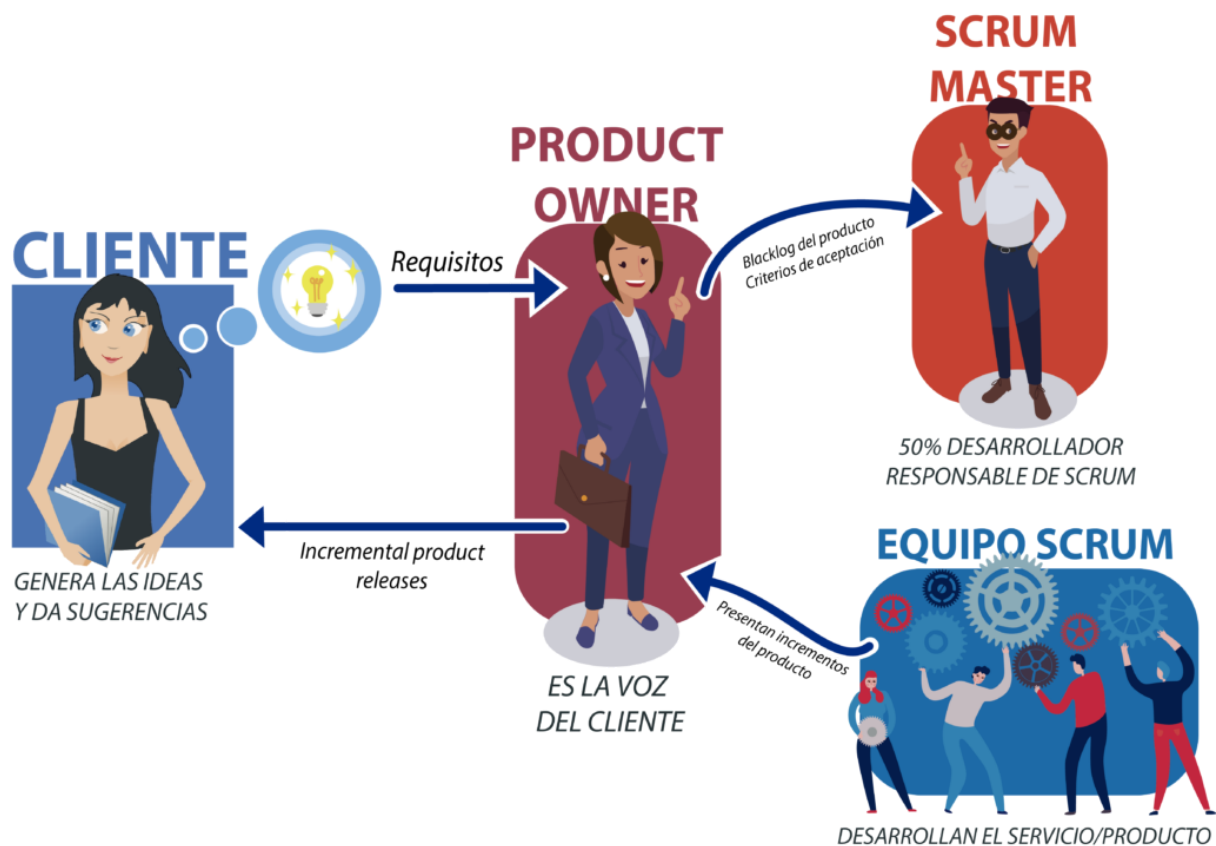
- **Product Owner:** Su principal misión es encargarse de que exista una priorización clara de los objetivos a conseguir, con el propósito de maximizar el valor del trabajo que lleva a cabo el equipo. Las responsabilidades del Product Owner son:
 - Conocer el mercado y los comportamientos de los clientes o usuarios finales, con muy buena visión de Negocio.
 - Ser el representante de todas las personas interesadas para conseguir una buena definición de los objetivos del producto o proyecto y de los resultados esperados.

El Product Owner es responsable de definir las historias de usuario, priorizar el trabajo de la lista de tareas pendientes del proyecto y tomar decisiones sobre qué funcionalidades se desarrollan y en qué orden. Además, el Product Owner está disponible para el equipo de desarrollo durante el sprint para proporcionar claridad sobre los requisitos y tomar decisiones rápidas.

- **Scrum Master:** lidera al equipo y se encarga de conseguir que el equipo conozca los principios y valores ágiles, así como la teoría y prácticas de Scrum, con el objetivo de que los

usen en sus procesos de toma de decisiones. El Scrum Master es un facilitador y un líder servicial que trabaja para ayudar al equipo de Scrum a alcanzar su máximo potencial y a seguir las prácticas de Scrum de manera efectiva. De este modo, es el coach y líder al servicio del equipo, llevando a cabo las siguientes responsabilidades:

- o El Scrum Master actúa como entrenador para el equipo de Scrum velando por que todos los participantes del proyecto sigan los valores y principios ágiles, las reglas y proceso de Scrum y guiar la colaboración dentro del equipo y con Product Owner.
 - o Proporcionar orientación y apoyo para mejorar el rendimiento del equipo.
 - o Quitar impedimentos que el equipo tiene en su camino para conseguir el objetivo de cada iteración y poder finalizar el proyecto con éxito.
 - o El Scrum Master facilita las reuniones y eventos de Scrum, como las reuniones de planificación del sprint, las reuniones diarias, la revisión del sprint y la retrospectiva del sprint. Ayuda a organizar estas reuniones y a garantizar que se sigan las reglas y los procesos de Scrum.
 - o El Scrum Master lidera la mejora continua del equipo. Facilita la retrospectiva del sprint, una reunión al final de cada sprint donde el equipo reflexiona sobre su desempeño y busca maneras de mejorar en el próximo sprint.
- El equipo de desarrollo es el conjunto de personas que de manera conjunta desarrollan el producto del proyecto. Tienen un objetivo común, comparten la responsabilidad del trabajo que realizan así como de su calidad en cada iteración y en el proyecto. Son multidisciplinarios y autoorganizados, lo que significa que se organizan y asignan tareas entre ellos según las habilidades y las necesidades del proyecto. El tamaño del equipo debe ser de 5 a 9 personas y trabajan conjuntamente



7. SCRUM. Define los siguientes términos:

- **Product backlog:** En la metodología de trabajo scrum, el product owner realiza una lista priorizada de requisitos que recibe el nombre de Product backlog. Los requisitos en el Product Backlog están representados en historias de usuario que describen funcionalidades o características del producto desde la perspectiva del usuario. Estos requisitos adquieren mayor prioridad en función del valor que aporten para el cliente.
- **Sprint backlog:** es una lista de tareas identificadas por el equipo de scrum que deberá ser completada durante cada sprint. El sprint backlog es representado a través de un tablero de tareas que hace visible todo el trabajo necesario para alcanzar el compromiso que se hizo con el product owner para el sprint.

8. SCRUM. En la terminología Scrum qué términos se utilizan como sinónimo de:

- Jefe de proyecto: Scrum Master
- Cliente: Product Owner
- Equipo de desarrollo: equipo de desarrollo

9. SCRUM. Haz un resumen de los requisitos para poder utilizar Scrum. Consulta el siguiente enlace:

- La cultura de la empresa debe fomentar el trabajo en equipo, la colaboración entre todas las personas implicadas en un proyecto, la creatividad, la transparencia y la mejora continua.
- Requiere una alta implicación y dedicación del cliente en la dirección de los resultados del proyecto, gestión del ROI y disponibilidad para poder colaborar.
- La dirección debe comprometerse a apoyar Scrum, identificar y eliminar obstáculos existentes y futuros, y tomar decisiones para facilitar la transición.
- Compromiso de la Dirección de la organización para resolver problemas endémicos y realizar cambios organizativos, formando equipos autogestionados y multidisciplinarios y fomentando una cultura de gestión basada en la colaboración y en la facilitación llevada a cabo por líderes al servicio del equipo.
- Compromiso conjunto y colaboración de los miembros del equipo.
- La relación entre el cliente y el proveedor se basa en un enfoque ganar-ganar, donde se asume que habrá cambios controlados para satisfacer las necesidades reales del cliente
- Scrum permite la incorporación incremental de requisitos y cambios controlados en el producto del proyecto.
- Tamaño de cada equipo entre 5 y 9 personas
- Equipo trabajando en un mismo espacio común para maximizar la comunicación.
- Los miembros del equipo deben dedicarse al proyecto a tiempo completo para evitar la pérdida de productividad debido a cambios constantes de tarea.
- El equipo debe ser estable durante el proyecto para aprovechar las relaciones interpersonales y la organización del trabajo que se han establecido.

10. XP. Explica los 5 valores de la Programación Extrema.

La programación extrema es una metodología ágil de gestión de proyectos que se centra en la velocidad y la simplicidad con ciclos de desarrollo cortos y con menos documentación. La estructura del proceso está determinada por 5 valores fundamentales, 5 reglas y 12 prácticas

Sus cinco valores fundamentales son:

1. Simplicidad: la programación extrema aboga por mantener las cosas lo más simples posible. Los equipos deben buscar soluciones simples y directas para los problemas en lugar de crear soluciones complejas y excesivamente detalladas.

2. Comunicación: la programación extrema se basa en una respuesta rápida y una comunicación efectiva. Para trabajar de manera efectiva, el equipo debe ser abierto y honesto. Ante los problemas todos aportan sus comentarios e ideas, ya que probablemente alguno de ellos ya tenga una solución adecuada. Y si no la tiene, podrán resolver el problema más rápidamente como grupo de lo que podrían hacerlo solos.
3. Comentarios o retroalimentación: El enfoque de XP es producir trabajo de forma rápida y sencilla, para luego compartir los resultados para obtener comentarios de forma casi inmediata. Los equipos recopilan comentarios de los clientes y las pruebas de software para ajustar y mejorar el producto.
4. Valentía: la valentía en XP se refiere a la disposición del equipo para tomar decisiones audaces y arriesgadas en beneficio del producto y del cliente. Los miembros del equipo deben ser valientes para admitir errores, probar nuevas ideas y hacer cambios significativos cuando sea necesario para mejorar el producto
5. Respeto: para que los equipos se comuniquen y colaboren de manera efectiva, deben aprender a estar en desacuerdo. El respeto es una base importante que promueve la bondad y la confianza, incluso cuando se expresan las opiniones con total honestidad.

11. XP. ¿Cuáles son las características distintivas de XP frente a otras metodologías ágiles?

Explícalas. Puedes consultar el siguiente enlace:

Las diferencias principales entre eXtreme Programming y otras metodologías ágiles son las siguientes:

- En XP no existen rangos no hay rangos jerárquicos formales como metodologías ágiles. XP promueve un enfoque colaborativo, en el que cada miembro del equipo es valorado por sus habilidades.
- En la Programación Extrema, el equipo de trabajo y el cliente colaboran para decidir qué funcionalidades o tareas deben priorizarse. La toma de decisiones sobre la priorización es un proceso colaborativo y participativo que involucra tanto a los miembros del equipo de desarrollo como al cliente. En otras metodologías ágiles se lleva a cabo de otras formas.
- XP es conocida por su flexibilidad extrema para adaptarse a los cambios en los requisitos. La toma de decisiones en XP puede ser altamente dinámica y puede ajustarse rápidamente según las necesidades cambiantes del cliente. XP se destaca por su capacidad para realizar cambios significativos incluso durante una iteración frente al resto de metodologías ágiles

- XP se enfoca en la entrega continua e incremental de software al cliente. Otras metodologías, como Kanban, también pueden permitir un flujo de trabajo continuo pero XP es conocida por su enfoque en la entrega rápida y regular.
- XP enfatiza la retroalimentación inmediata a través de prácticas como las revisiones de código y las pruebas automatizadas. Otras metodologías también valoran la retroalimentación, pero la frecuencia y el enfoque pueden variar.