**GUÍA PARA DESARROLLAR EL TEXTO GUÍA**

PROGRAMA:

PROCESOS AGILES DE SOFTWARE

(Texto Guía)

DEPARTAMENTO OFERENTE:

INGENIERÍA

AUTOR(A):

JORGE ALEJANDRO AGUIRRE GUTIERREZ

FORMACIÓN:

ESPECIALIZACION DE INGENIERÍA DE SOFTWARE CON ÉNFASIS EN PRUEBAS

Manizales, 2024

**Tabla de Contenidos**

**UNIDAD 1: TEORIA SOBRE METODOLOGIAS AGILES.**

* Conceptos de procesos agiles de software.
* Beneficios de los procesos agiles de software.
* Diferencias entre metodologías agiles y tradicionales.
* Conceptos del SCRUM.
* Roles del SCRUM.
* Organización y épicas.
* Backlog.
* Historias de usuario.
* Sprint.
* Evento Planning.
* Evento daily.
* Evento Review.
* Evento Retrospective.

***Antes de iniciar el texto guía, por favor lea con atención estas recomendaciones generales para el escrito:***

*El texto guía es un escrito académico elaborado por el autor del curso y tiene como principal función “presentar de forma didáctica el conocimiento de las distintas disciplinas…Por su diseño, son textos mixtos que reúnen información textual con otra de tipo visual (uso de imágenes, gráficas, tablas, etc.). También contienen muchos tipos de ayudas didácticas, especialmente seleccionadas y diseñadas, con el propósito de favorecer el aprendizaje (Díaz & Hernández, 2010, p. 158), tales como: videos, simuladores, animaciones, cuestionarios de verificación de lectura, ejercicios interactivos (opción múltiple, falso y verdadero, emparejamiento, clasificar conceptos) o juegos interactivos de Educaplay (*[*Ver tipo de actividades*](https://es.educaplay.com/tipos-de-actividades/)*) con recursos multimedia (video, audio, imagen, texto)*

*Siguiendo a Díaz y Hernández (2010), los textos considerados accesibles se caracterizan por:*

* *Tienen un arreglo estructurado y sistemático de las ideas.*
* *Poseen un buen nivel de coherencia de las ideas expuestas*
* *Son concisos y contienen poca información distractora o irrelevante*
* *Toman en cuenta el conocimiento previo del lector*
* *Presentan al lector la estructura del texto de forma clara (con señalizaciones) para beneficiar su comprensión profunda*

*Por favor tenga en cuenta las siguientes recomendaciones:*

* *Utilizar Verdana tamaño 12, interlineado sencillo.*
* *Las fuentes consultadas para desarrollar el contenido no deben ser de más de cinco años de antigüedad. Si se requiere utilizar fuentes de más años, se debe justificar en las referencias al final de la unidad.*
* *Todas las gráficas, ilustraciones o figuras, deben estar enumeradas con la herramienta de Word REFERENCIAS: INSERTAR TÍTULO. Debe registrarse la FUENTE: Autor, fecha, o sitio web. Cuando la gráfica es de autoría del docente, se debe especificar al final del título de la gráfica: FUENTE: ELABORACIÓN PROPIA.*
* *En el desarrollo del contenido, cuando se trate de ideas fundamentadas en otros autores, se debe hacer la cita correspondiente. Del mismo modo diferenciar claramente las opiniones del docente autor del módulo.*
* *Las citas deben referenciarse con Normas VANCOUVER (PARA CIENCIAS DE LA SALUD) – APA (CIENCIAS SOCIALES E INGENIERÍAS)*

**TÍTULO DE LA UNIDAD:**

**TEORIA SOBRE METODOLOGIAS AGILES**

**INTRODUCCIÓN A LA UNIDAD**

**Indicador de desempeño**

Al finalizar el estudio de esta unidad y desarrollar las actividades propuestas, el estudiante deberá estar en capacidad de: Comprender el concepto de metodología ágil en la industria del software y su diferencia con la metodología tradicional, además de conocer los fundamentos del SCRUM.

**-----------**

**Desarrollar cada uno de los temas propuestos**

***INDICACIONES:***

*Presentación del contenido utilizando estrategias que cumplan con las siguientes funciones:*

* *Detección de la información principal;*
* *Conceptualización de contenidos;*
* *Delimitación de la organización, estructura interna e interrelaciones entre dichos contenidos;*
* *y mantenimiento de la atención y motivación.*

*A lo largo del escrito haga uso de estrategias tales como: ilustraciones, esquemas, gráficas o resúmenes (al iniciar la unidad, al finalizar la unidad, o al finalizar un tema), preguntas intercaladas, analogías, pistas tipográficas (resaltar ideas centrales, conceptos, mapas conceptuales), ejemplificaciones para aclarar conceptos o mostrar aplicaciones, ejercicios.*

*Utilice recursos multimedia como material complementario: audios, videos, animaciones, simuladores, imágenes, entre otros.*

*Recuerde hacer las citas correspondientes con normas VANCOUVER O APA.*

*Recuerde proponer ejercicios interactivos a lo largo del texto para reforzar conocimientos o de verificación de comprensión de la lectura.*

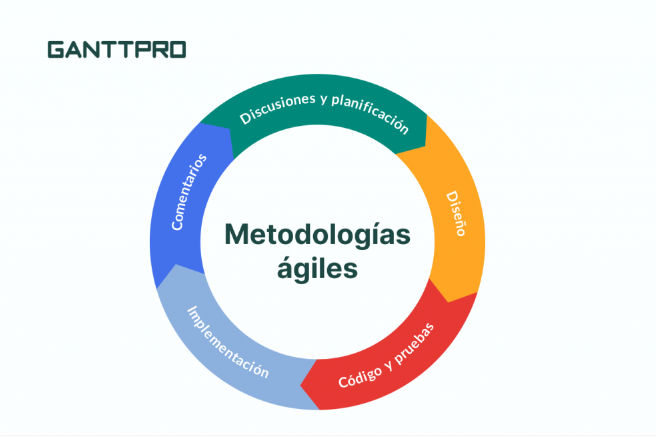
*Recuerde titular y enumerar cada figura, tabla, imagen, gráfica, ilustración y registrar la fuente. Especificar cuando se trate de una gráfica de su propia autoría.*

**TEMA 1: CONCEPTOS DE PROCESOS AGILES DE SOFTWARE**

Un proceso ágil de software es una metodología de desarrollo de software que se basa en la colaboración, la adaptación y la entrega continua de valor. En lugar de seguir un plan rígido y definido desde el principio, los procesos ágiles dividen el trabajo en ciclos cortos de desarrollo, conocidos como iteraciones. En cada iteración, el equipo se enfoca en entregar una funcionalidad específica del software, recibiendo retroalimentación del cliente y adaptándose a los cambios en los requisitos a medida que avanza el proyecto.

Características clave de los procesos ágiles de software:

* Enfoque en el cliente: La satisfacción del cliente es la máxima prioridad. Se busca involucrarlo activamente en el proceso y obtener su retroalimentación de manera constante.
* Adaptabilidad a los cambios: Los procesos ágiles son flexibles y permiten adaptarse a cambios en los requisitos o prioridades del proyecto sin afectar negativamente el desarrollo.
* Entrega incremental: Se entrega software funcional de manera regular, permitiendo al cliente probarlo y usarlo desde las primeras etapas del proyecto.
* Trabajo en equipo: Los equipos ágiles son multifuncionales y colaborativos, donde cada miembro aporta sus habilidades y conocimientos para el éxito del proyecto.
* Mejora continua: Se busca la mejora constante de los procesos y prácticas a través de la reflexión y la retrospectiva al finalizar cada iteración.



**TEMA 2: Beneficios de los procesos agiles en la organización**

Las metodologías ágiles ofrecen una serie de beneficios para las empresas que las implementan, tanto en el ámbito del desarrollo de software como en otros departamentos. Entre los principales beneficios podemos destacar:

1. Mayor satisfacción del cliente:

Al involucrar al cliente en el proceso de desarrollo, se asegura que el producto final cumpla con sus expectativas y necesidades.

La entrega incremental de software funcional permite al cliente probarlo y usarlo desde las primeras etapas del proyecto, proporcionando retroalimentación valiosa para mejorar el producto.

La comunicación constante y la transparencia entre el equipo de desarrollo y el cliente fomentan la confianza y la colaboración.

2. Mayor agilidad y flexibilidad:

Los procesos ágiles permiten adaptarse rápidamente a los cambios en los requisitos o prioridades del proyecto, sin afectar negativamente el desarrollo.

La división del trabajo en ciclos cortos facilita la identificación y corrección de problemas de manera temprana, evitando retrasos costosos al final del proyecto.

La cultura de mejora continua permite al equipo aprender de sus errores y optimizar sus procesos constantemente.

3. Mayor calidad del software:

Las pruebas unitarias continuas y la refactorización constante ayudan a mejorar la calidad del código y reducir la cantidad de errores.

La retroalimentación constante del cliente permite identificar y corregir problemas de usabilidad y funcionalidad antes de que el software llegue a manos del usuario final.

El enfoque en la entrega de valor en cada iteración asegura que el software final sea funcional y útil para el cliente.

4. Mayor motivación del equipo:

Los equipos ágiles suelen estar más motivados y comprometidos con el proyecto, ya que tienen mayor autonomía y responsabilidad sobre su trabajo.

La comunicación abierta y la colaboración entre los miembros del equipo fomentan un ambiente de trabajo positivo y productivo.

El reconocimiento y la celebración de los logros motivan al equipo a seguir mejorando su desempeño.

5. Mayor eficiencia y productividad:

La eliminación de tareas innecesarias y la optimización de los procesos permiten reducir el tiempo y los recursos necesarios para completar el proyecto.

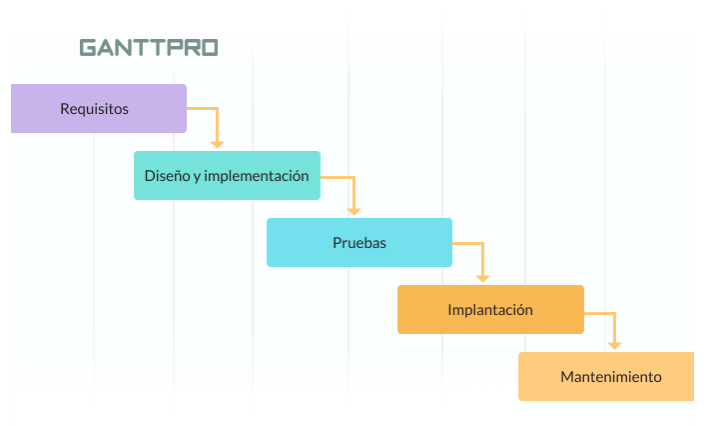
La entrega incremental de software permite liberar valor al cliente de manera más rápida, lo que puede generar ingresos antes de que el proyecto esté finalizado.

La mejora continua de los procesos permite aumentar la eficiencia y la productividad a largo plazo.

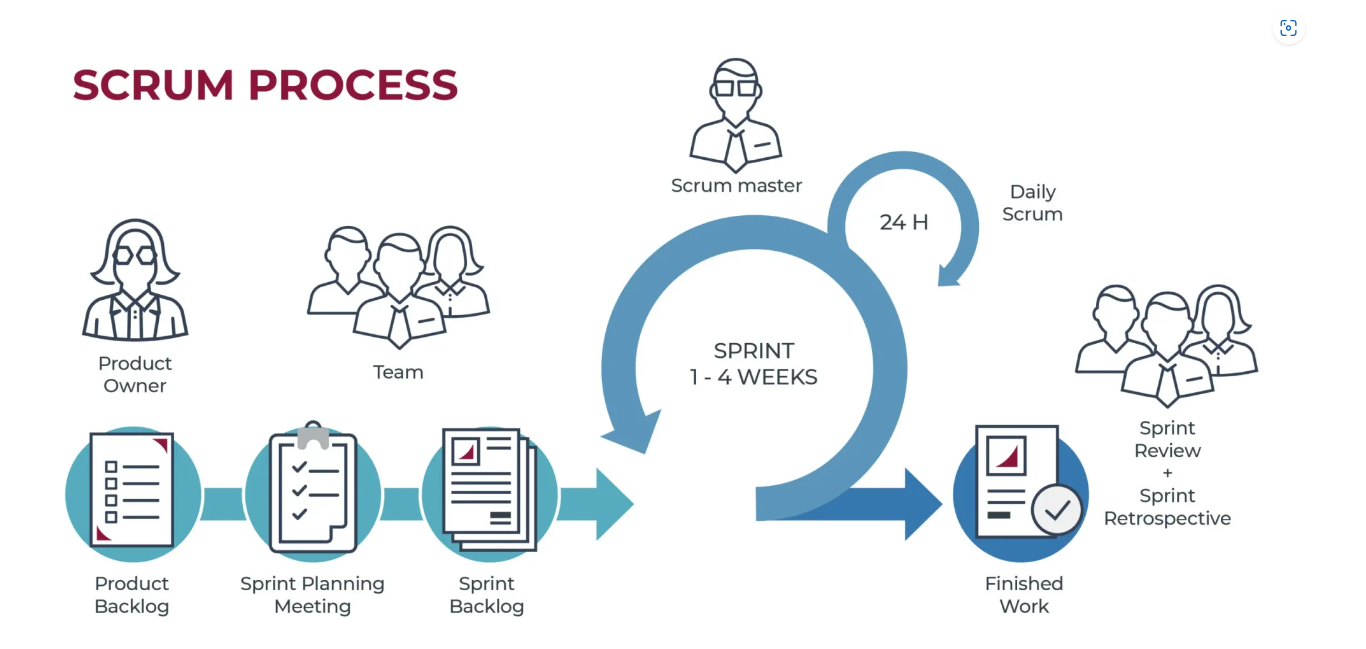


Video sobre los beneficios de las metodologías agiles en diferentes ámbitos empresariales: <https://www.youtube.com/watch?v=j8hNLrnhzFo>

**TEMA 3: DIFERENCIAS ENTRE METODOLOGIAS AGILES Y TRADICIONALES.**

**ESQUEMA METODOLOGIA TRADICIONAL**

**ESQUEMA METODOLOGIA AGIL**



|  |  |  |
| --- | --- | --- |
| **CARACTERÍSTICA** | **METODOLOGÍAS ÁGILES** | **METODOLOGÍAS TRADICIONALES** |
| Enfoque | Adaptativo e iterativo | Predictivo y lineal |
| Planificación | Planificación incremental y flexible | Planificación detallada y por adelantado |
| Entrega de valor | Entrega de valor en ciclos cortos y frecuentes | Entrega de valor al final del proyecto |
| Retroalimentación | Retroalimentación continua y adaptación | Retroalimentación al final de cada fase |
| Cambio | Se fomenta y se abraza el cambio | Se resiste el cambio |
| Colaboración | Se fomenta la colaboración entre equipos | El enfoque es más jerárquico |
| Documentación | Documentación ligera y enfocada | Documentación extensa y detallada |
| Ejemplos | Scrum, Kanban, Extreme Programming (XP) | Waterfall, Ciclo de Vida en Cascada (SDLC) |

**TEMA 4: CONCEPTOS DEL SCRUM**

Scrum es una metodología de gestión de proyectos ágil que se basa en ciclos cortos de trabajo llamados sprints. En cada sprint, el equipo se enfoca en entregar una parte valiosa del producto final.

Caso de uso comparativo: Equipo de futbol.

Imaginemos un equipo deportivo entrenando para un partido importante:

* El entrenador: Es como el Scrum Master, guía al equipo y facilita el proceso.
* Los jugadores: Son como el Equipo de Desarrollo, trabajan juntos para crear el producto final.
* Los entrenamientos: Son como los Sprints, cada uno con un objetivo específico.
* El plan de juego: Es como el Backlog del Producto, una lista de todas las tareas que se deben completar.
* Las reuniones: Son como las Ceremonias Scrum, momentos para planificar, revisar y celebrar el progreso.

En cada sprint:

* Planificación: Se seleccionan las tareas del backlog que se completarán en el sprint.
* Desarrollo: El equipo trabaja en las tareas seleccionadas.
* Revisión diaria: El equipo se reúne diariamente para discutir su progreso y resolver problemas.
* Revisión del sprint: Al final del sprint, el equipo muestra lo que ha completado a las partes interesadas.
* Retrospectiva: El equipo reflexiona sobre el sprint y busca formas de mejorar en el futuro.

Scrum es como un juego de equipo:

* Todos colaboran: Cada miembro del equipo tiene un rol importante que jugar.
* Comunicación constante: La comunicación abierta y fluida es esencial.
* Adaptación a los cambios: Los planes pueden cambiar a medida que se avanza, el equipo se adapta a los nuevos desafíos.
* Enfoque en el valor: Se prioriza la entrega de valor al cliente en cada sprint.
* Mejora continua: El equipo busca constantemente formas de mejorar su proceso y producto.

Scrum es ideal para proyectos:

* Con requisitos cambiantes: El enfoque flexible de Scrum permite adaptarse a los cambios fácilmente.
* Que requieren colaboración: Scrum fomenta el trabajo en equipo y la comunicación abierta.
* Con entregas frecuentes: Scrum permite entregar valor al cliente de forma regular.

**TEMA 5: ROLES DEL SCRUM**

1. Scrum Master: El guía sabio

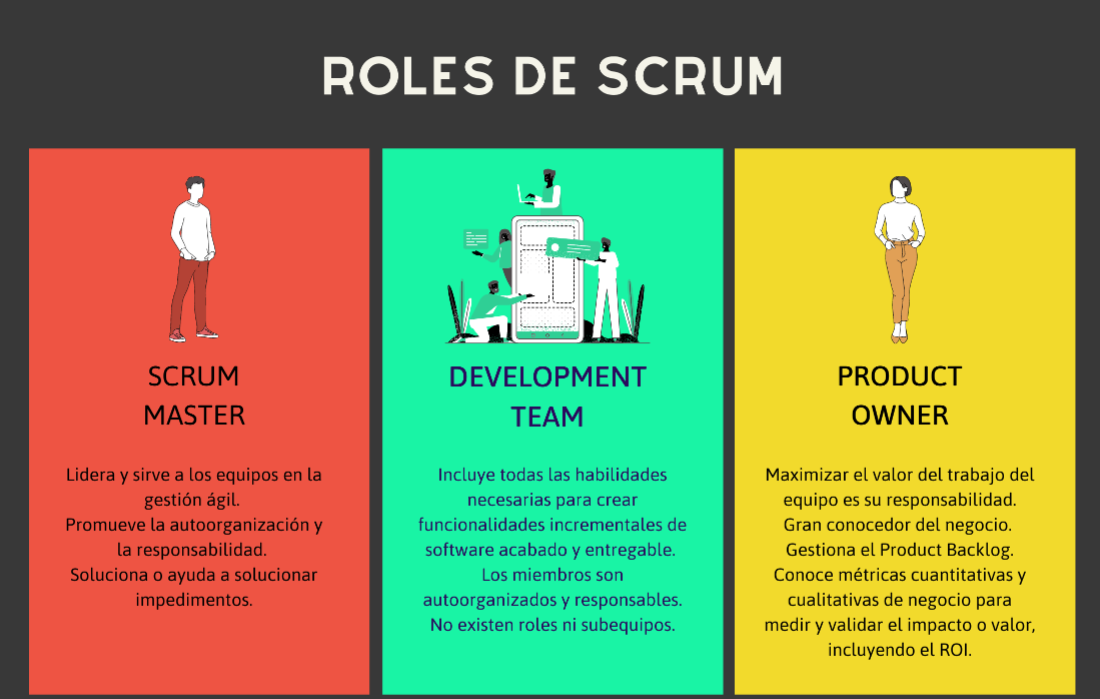
* Misión: Guiar y facilitar el proceso Scrum, eliminando obstáculos y asegurando que el equipo siga los principios Scrum.
* Habilidades: Liderazgo, comunicación, resolución de problemas y conocimiento de Scrum.
* Acciones: Convocar y facilitar las ceremonias Scrum, proteger el tiempo del equipo, resolver conflictos y motivar al equipo.

2. Product Owner: El visionario

* Misión: Representar las necesidades del cliente y definir la visión del producto.
* Habilidades: Comprensión del negocio, comunicación, priorización y visión estratégica.
* Acciones: Crear y mantener el backlog del producto, priorizar las tareas, colaborar con el equipo y recopilar feedback de las partes interesadas.

3. Equipo de Desarrollo: Los héroes valientes

* Misión: Desarrollar el producto de acuerdo con el backlog del producto.
* Habilidades: Habilidades técnicas, trabajo en equipo, autoorganización y resolución de problemas.
* Acciones: Estimar tareas, diseñar, desarrollar, probar y entregar el producto.



**TEMA 6: LA ORGANIZACIÓN Y LAS EPICAS**

En el mundo de Scrum, la organización y las épicas son dos piezas fundamentales que permiten a los equipos abordar proyectos complejos de manera eficiente y efectiva.

**Organización:**

Imagina una biblioteca gigante con miles de libros. Sin un sistema de organización, encontrar el libro que buscas sería una tarea titánica. Lo mismo sucede en Scrum. La organización es clave para:

* Estructurar el trabajo: El backlog del producto es la biblioteca de Scrum, donde se encuentran todas las historias de usuario, funcionalidades y mejoras que se deben implementar. Una buena organización del backlog permite priorizar las tareas, facilitar la planificación y garantizar que el equipo se enfoque en lo que realmente importa.
* Desglosar grandes objetivos: Las épicas son como capítulos dentro de la historia del producto. Representan grandes objetivos que se descomponen en historias de usuario más pequeñas y manejables. Esto facilita la comprensión del proyecto, la estimación del esfuerzo y la gestión de las dependencias.
* Visualizar el progreso: El tablero Kanban es como un mapa del tesoro para el equipo Scrum. Permite visualizar el flujo de trabajo, identificar cuellos de botella y celebrar los avances. Una buena organización del tablero Kanban proporciona transparencia y facilita la colaboración entre los miembros del equipo.

**Épicas:**

Las épicas son como las grandes batallas en una guerra. Son objetivos a gran escala que abarcan varias historias de usuario y funcionalidades relacionadas. Dividir el proyecto en épicas ofrece múltiples beneficios:

* Agrupar historias relacionadas: Las épicas permiten agrupar historias de usuario que comparten un objetivo común. Esto facilita la comprensión del proyecto, la estimación del esfuerzo y la planificación de las entregas.
* Gestión de dependencias: Las épicas pueden tener dependencias entre sí, lo que significa que algunas deben completarse antes que otras. Identificar y gestionar estas dependencias es crucial para evitar retrasos y garantizar el éxito del proyecto.
* Comunicación efectiva: Las épicas son una excelente herramienta de comunicación para las partes interesadas. Permiten comunicar los objetivos generales del proyecto y el progreso general.

**TEMA 7: EL BACKLOG**

En Scrum, el backlog es como un mapa del tesoro que guía al equipo hacia el éxito.

¿Qué es el backlog?

El backlog es una lista priorizada de todas las tareas, funcionalidades y mejoras que se deben implementar en un producto. Es como una lista de compras para un chef: contiene todos los ingredientes necesarios para crear una deliciosa comida.

¿Quién crea el backlog?

El Product Owner es el responsable de crear y mantener el backlog. Es como el chef de la aventura, quien decide qué ingredientes se necesitan y en qué orden se deben cocinar.

¿Cómo se organiza el backlog?

El backlog se organiza en historias de usuario, que son descripciones breves y claras de las necesidades o deseos de los usuarios. Cada historia de usuario debe tener un valor claro y ser estimable en términos de tiempo y esfuerzo.

¿Cómo se prioriza el backlog?

El Product Owner prioriza el backlog en función del valor que cada historia de usuario aporta al producto. Las historias de usuario con mayor valor se colocan en la parte superior del backlog, mientras que las menos importantes se colocan en la parte inferior.

¿Cómo se utiliza el backlog?

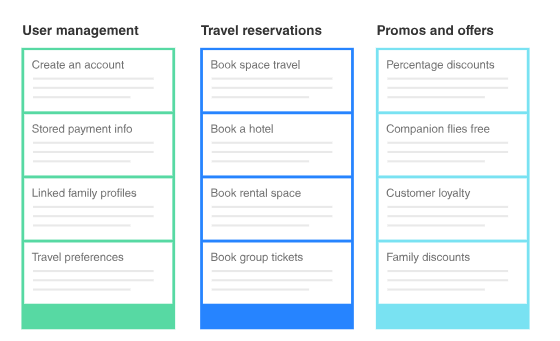
El backlog se utiliza durante la planificación del sprint para seleccionar las historias de usuario que se completarán en el siguiente sprint. El equipo de desarrollo estima el tiempo que llevará completar cada historia de usuario y se compromete a entregarlas al final del sprint.

Beneficios del backlog:

* Transparencia: El backlog proporciona transparencia a todos los miembros del equipo y a las partes interesadas sobre el trabajo que se debe realizar.
* Enfoque: Ayuda al equipo a enfocarse en las tareas más importantes y de mayor valor.
* Flexibilidad: El backlog es flexible y puede adaptarse a los cambios en las necesidades del cliente o del mercado.
* Mejora continua: El backlog se revisa y actualiza constantemente para reflejar las nuevas prioridades y los comentarios de las partes interesadas.

Video: Que es el Backlog para la empresa Atlassian dueño de la plataforma Jira <https://youtu.be/zkZJxvKzILY>

Ejemplo de Backlog



**TEMA 8: HISTORIAS DE USUARIO**

Las Historias de usuario son ¡La voz del cliente en Scrum!

En el mundo de Scrum, las historias de usuario son como las voces de los clientes que guían al equipo hacia la creación de un producto que realmente satisfaga sus necesidades.

¿Qué son las historias de usuario?

Las historias de usuario son descripciones breves y claras de las necesidades o deseos de los usuarios. Son como pequeñas historias que describen cómo un usuario específico quiere interactuar con el producto.

¿Cómo se escriben las historias de usuario?

Las historias de usuario se escriben siguiendo una estructura simple:

Como: Esta frase describe al usuario que está realizando la acción.

Yo quiero: Esta frase describe la necesidad o el deseo del usuario.

Para que: Esta frase explica el beneficio o el objetivo que el usuario quiere lograr.

Ejemplo de una historia de usuario:

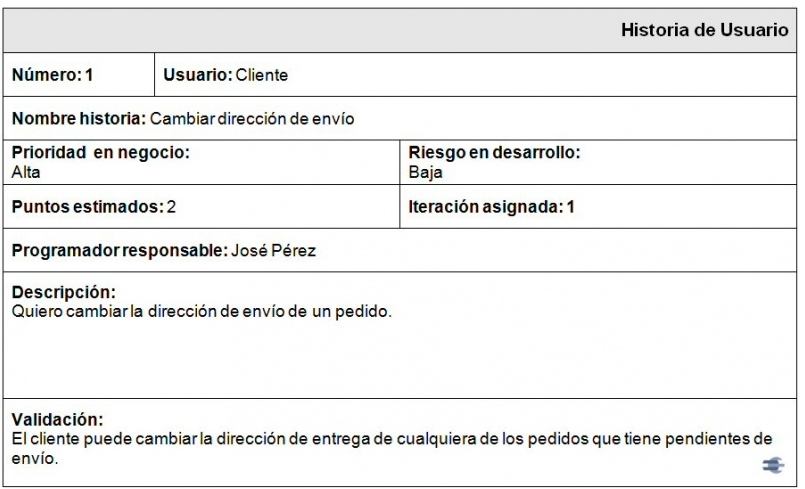
Como cliente, yo quiero poder buscar productos por categoría para que pueda encontrar fácilmente lo que busco.

¿Por qué son importantes las historias de usuario?

Las historias de usuario son importantes por varias razones:

* Capturan las necesidades del cliente: Las historias de usuario se basan en las necesidades y deseos reales de los clientes, lo que garantiza que el producto esté diseñado para satisfacer sus necesidades.
* Promueven la colaboración: Las historias de usuario facilitan la colaboración entre el Product Owner y el equipo de desarrollo, asegurando que todos estén en la misma página.
* Facilitan la estimación: Las historias de usuario son fáciles de entender y estimar, lo que ayuda al equipo a planificar los sprints de manera efectiva.
* Guían el desarrollo: Las historias de usuario sirven como guía para el equipo de desarrollo, asegurando que el producto se desarrolle de acuerdo a las necesidades del cliente.

EJEMPLO DE HISTORIA DE USUARIO:



**TEMA 9: SPRINT**

Imagina una carrera de relevos emocionante, donde cada miembro del equipo corre una parte de la distancia para alcanzar la meta juntos. En Scrum, los sprints son como esas carreras de relevos, pero en lugar de correr, el equipo trabaja en conjunto para desarrollar y entregar una parte valiosa del producto final.

¿Qué es un sprint?

Un sprint es un período corto de tiempo (generalmente de 1 a 4 semanas) durante el cual el equipo de Scrum se enfoca en completar un conjunto específico de historias de usuario. Es como una carrera corta con un objetivo definido: entregar valor al cliente al final del sprint.

¿Qué sucede en un sprint?

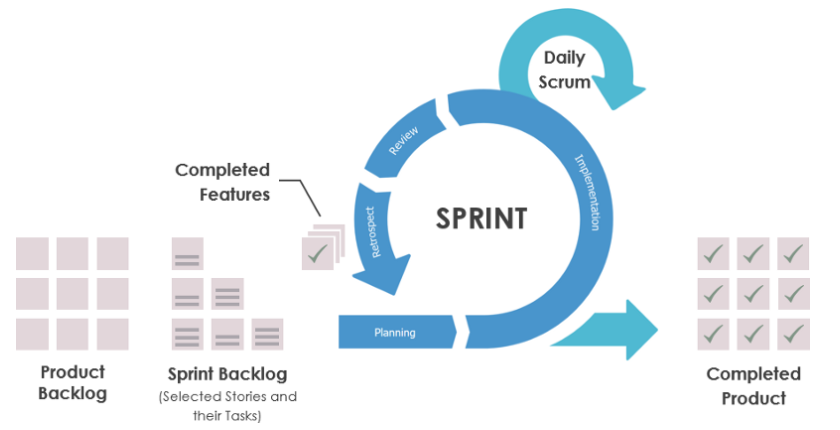
Un sprint se divide en varias etapas:

1. Planificación del sprint: Al inicio del sprint, el equipo selecciona las historias de usuario del backlog del producto que se completarán en el sprint.
2. Desarrollo: Durante el sprint, el equipo de desarrollo trabaja en las historias de usuario seleccionadas, utilizando sus habilidades y experiencia para crear un producto que funcione y satisfaga las necesidades del cliente.
3. Revisión diaria: Cada día, el equipo se reúne para discutir su progreso, identificar problemas y ajustar su plan según sea necesario.
4. Revisión del sprint: Al final del sprint, el equipo presenta lo que ha completado a las partes interesadas.
5. Retrospectiva del sprint: Después de la revisión del sprint, el equipo reflexiona sobre lo que salió bien y lo que no, y busca formas de mejorar su proceso para el próximo sprint.

¿Por qué son importantes los sprints?

Los sprints son importantes por varias razones:

* Entrega de valor frecuente: Los sprints permiten al equipo entregar valor al cliente de forma regular, lo que ayuda a obtener feedback temprano y a mejorar el producto de forma continua.
* Enfoque y motivación: Los sprints ayudan al equipo a mantenerse enfocado en un objetivo específico durante un período corto de tiempo, lo que aumenta la motivación y la productividad.
* Adaptación a los cambios: Los sprints permiten al equipo adaptarse a los cambios en las necesidades del cliente o del mercado de forma rápida y flexible.
* Mejora continua: Los sprints promueven una cultura de mejora continua, donde el equipo aprende de sus experiencias y busca formas de mejorar su proceso en cada sprint.



**TEMA 10: EVENTO PLANNING**

El planning, también conocido como planificación del sprint, es una ceremonia esencial en Scrum que se realiza al inicio de cada sprint. Es como una reunión crucial donde el equipo se prepara para la aventura que les espera.

¿Qué sucede en el planning?

En el planning, el equipo se reúne con el Product Owner para:

1. Revisar el backlog del producto: Se analizan las historias de usuario priorizadas y se seleccionan las que se completarán en el sprint.
2. Estimar el esfuerzo: El equipo estima el tiempo y esfuerzo que requerirá completar cada historia de usuario seleccionada.
3. Crear el backlog del sprint: Se crea una lista de las historias de usuario seleccionadas para el sprint, junto con su estimación de esfuerzo.
4. Definir el objetivo del sprint: Se establece un objetivo claro y conciso para el sprint, que describe lo que el equipo se compromete a entregar al final del mismo.

¿Por qué es importante el planning?

* Alineación del equipo: Asegura que todos los miembros del equipo estén en la misma página y comprendan el objetivo del sprint.
* Enfoque y priorización: Ayuda al equipo a enfocarse en las historias de usuario más importantes y priorizadas.
* Planificación realista: Permite al equipo planificar el sprint de manera realista, teniendo en cuenta el tiempo y esfuerzo disponibles.
* Mejora continua: Brinda la oportunidad de reflexionar sobre los sprints anteriores y buscar formas de mejorar el proceso de planificación.

**TEMA 11: EVENTO DAILY**

El Daily Scrum es una reunión breve (máximo 15 minutos) que se realiza todos los días durante un sprint. Es como una breve parada dentro de la cueva para que los exploradores se comuniquen y planifiquen su siguiente paso.

¿Qué sucede en el Daily Scrum?

En el Daily Scrum, cada miembro del equipo responde tres preguntas:

* ¿Qué hice ayer para completar el objetivo del sprint?
* ¿Qué haré hoy para completar el objetivo del sprint?
* ¿Hay algún impedimento que me impida alcanzar el objetivo del sprint?
* ¿Por qué es importante el Daily Scrum?

El Daily Scrum es importante por varias razones:

* Mejora la comunicación: Fomenta la comunicación diaria entre los miembros del equipo, lo que ayuda a evitar malentendidos y mantener a todos en la misma página.
* Identifica y resuelve problemas: Permite al equipo identificar y resolver problemas de manera temprana, evitando que se conviertan en obstáculos mayores.
* Promueve la colaboración: Fomenta la colaboración entre los miembros del equipo, permitiéndoles compartir ideas y ayudarse mutuamente.
* Adaptación al cambio: Permite al equipo adaptarse a los cambios en las prioridades o en el entorno del proyecto de manera rápida y flexible.
* Mejora continua: Brinda la oportunidad de reflexionar sobre el progreso diario y buscar formas de mejorar el proceso en el siguiente sprint.

**TEMA 12: EVENTO REVIEW**

La review es una ceremonia esencial en Scrum que se realiza al final de cada sprint. Es como una reunión final donde los aventureros comparten sus logros y aprenden de sus experiencias.

¿Qué sucede en la review?

En la review, el equipo:

* Presenta lo que ha completado durante el sprint: Demostrará las funcionalidades desarrolladas, las historias de usuario finalizadas y cualquier otro avance significativo.
* Recibe feedback de las partes interesadas: Las partes interesadas brindarán sus comentarios sobre el trabajo realizado, identificando aspectos positivos y áreas de mejora.
* Responde preguntas y aclara dudas: El equipo responderá a las preguntas de las partes interesadas y aclarará cualquier duda que puedan tener.
* Reflexiona sobre el sprint: El equipo analizará el proceso de desarrollo, identificando lo que funcionó bien y lo que se puede mejorar en el futuro.

¿Por qué es importante la review?

La review es importante por varias razones:

* Transparencia: Permite a las partes interesadas ver el progreso del proyecto y brindar feedback valioso.
* Inspección y adaptación: Brinda la oportunidad de inspeccionar el producto y adaptar el proceso según sea necesario.
* Celebración de los logros: Es un momento para celebrar los logros del equipo y reconocer su esfuerzo.
* Mejora continua: Promueve la mejora continua del proceso de desarrollo, identificando áreas de mejora y buscando nuevas formas de trabajar de manera más efectiva.

**TEMA 13: EVENTO RETROSPECTIVE**

En el mundo del desarrollo ágil, una retrospectiva es una reunión o taller que se lleva a cabo al final de cada ciclo de trabajo (sprint o iteración) para reflexionar sobre lo que salió bien, lo que no funcionó y cómo se puede mejorar el proceso en el futuro. Es un espacio para que el equipo aprenda de sus experiencias y trabaje en conjunto para identificar áreas de mejora.

Objetivos de una retrospectiva:

* Promover la mejora continua: El objetivo principal de una retrospectiva es identificar oportunidades para mejorar el proceso de desarrollo de software. Esto se logra mediante la reflexión sobre lo que salió bien y lo que no en el ciclo anterior, y luego generando acciones concretas para abordar las áreas problemáticas.
* Fomentar el trabajo en equipo: Las retrospectivas son una excelente manera de fortalecer el trabajo en equipo y la comunicación entre los miembros del equipo. Al discutir abiertamente los desafíos y éxitos del ciclo anterior, los miembros del equipo pueden aprender unos de otros y desarrollar una mayor comprensión de las diferentes perspectivas.
* Aumentar la responsabilidad: Las retrospectivas ayudan a crear un entorno en el que los miembros del equipo se sienten responsables de su propio trabajo y del éxito del equipo en general. Al participar activamente en la discusión y la generación de soluciones, los miembros del equipo se comprometen a realizar los cambios necesarios para mejorar.

Estructura de una retrospectiva:

Las retrospectivas no tienen una estructura fija, pero por lo general siguen un proceso similar que incluye los siguientes pasos:

1. Reunir al equipo: El primer paso es reunir a todos los miembros del equipo en un lugar cómodo donde puedan hablar libremente.
2. Establecer un ambiente seguro: Es importante crear un ambiente seguro y de confianza donde los miembros del equipo se sientan cómodos para compartir sus opiniones honestas sin temor a represalias.
3. Recopilar información: El siguiente paso es recopilar información sobre lo que sucedió durante el ciclo anterior. Esto se puede hacer utilizando una variedad de técnicas, como lluvia de ideas, encuestas o simplemente tener una discusión abierta.
4. Analizar la información: Una vez que se ha recopilado la información, el equipo debe analizarla para identificar los patrones y temas comunes. Esto ayudará al equipo a comprender mejor las áreas que necesitan mejorar.
5. Generar acciones: El último paso es generar acciones concretas que el equipo puede tomar para abordar las áreas problemáticas identificadas. Estas acciones deben ser específicas, medibles, alcanzables, relevantes y con plazos definidos (SMART).

Consejos para una retrospectiva efectiva:

* Involucrar a todos los miembros del equipo: Es importante que todos los miembros del equipo tengan la oportunidad de participar en la retrospectiva. Esto ayudará a garantizar que se escuchen todas las perspectivas y que la discusión sea completa.
* Ser respetuoso: Es importante ser respetuoso con todas las opiniones, incluso si no está de acuerdo con ellas. El objetivo de la retrospectiva es aprender y mejorar, no culpar o criticar.
* Centrarse en el futuro: La retrospectiva debe centrarse en cómo mejorar el futuro, no en culpar a las personas por los errores del pasado.
* Tomar medidas: Es importante que el equipo tome medidas concretas para abordar las áreas problemáticas identificadas. Las acciones deben ser SMART y deben tener un propietario asignado.

**CIERRE DE LA UNIDAD**

***INDICACIONES:***

*Una vez terminado el desarrollo del contenido, procure formar una visión sintética, integradora e incluso crítica de la unidad, a través de estrategias como: resumen, mapa conceptual, preguntas o conclusiones generales de los temas presentados en el escrito.*

*Proponga material o lecturas complementarias si lo considera necesario.*

**REFERENCIAS BIBLIOGRÁFICAS**

***INDICACIONES:***

*Registre la documentación (NORMAS APA O VANCOUVER) consultada para desarrollar el contenido.*

*Utilice bibliografía con una antigüedad no superior a los cinco años.*