1. **INTRODUCCIÓN A LAS METODOLOGÍAS ÁGILES**

Las metodologías ágiles surgen como respuesta efectiva a los desafíos actuales a la hora de hacer proyectos, siendo estos la adaptabilidad rápida a los cambios del mercado, satisfacer las demandas de los clientes o la entrega de productos de alta calidad en plazos ajustados. Las metodologías ágiles son un conjunto de enfoques y prácticas de gestión de proyectos que enfatizan la flexibilidad, la colaboración y la adaptación constante a los cambios. Valoran la interacción humana, la entrega incremental y la respuesta ágil a los cambios en los requisitos del proyecto.

Principios y valores fundamentales:

* **Colaboración y comunicación.** Fomentan la colaboración activa entre los miembros del equipo y los interesados, promoviendo la comunicación constante y la toma de decisiones conjuntas.
* **Flexibilidad y adaptabilidad.** Se adaptan de manera fluida a los cambios en los requisitos y prioridades del proyecto, lo que permite responder rápidamente a las demandas del mercado.
* **Entrega incremental.** Dividen el proyecto en entregas pequeñas y manejables, lo que permite obtener resultados tangibles y valiosos de manera temprana.
* **Feedback continuo.** Se basan en la retroalimentación constante de los interesados y usuarios, lo que permite ajustar y mejorar el producto en cada iteración.
* **Enfoque en el cliente.** Colocan al cliente en el centro del proceso, asegurándose de que el producto final satisfaga sus necesidades y expectativas.
* **Autogestión del equipo.** Empoderan al equipo para tomar decisiones y autogestionar su trabajo, lo que aumenta la responsabilidad y la motivación.

Existen varias metodologías ágiles.

* **Scrum.** Se basa en la división del proyecto en iteraciones llamadas sprints. Cada sprint tiene una duración fija y al final se entrega un incremento funcional del producto.
* **Kanban.** Se centra en el flujo de trabajo visual y limita la cantidad de trabajo en progreso. Las tareas se mueven a través de etapas definidas, lo que ayuda a mantener un flujo constante y a detectar cuellos de botella.
* **Extreme Programming (XP).** Se centra en las prácticas de desarrollo de software, como la programación en parejas y las pruebas automatizadas. Prioriza la entrega de software funcional y de alta calidad.
* **Crystal.** Es una familia de metodologías ágiles que se adaptan según el tamaño y la complejidad del proyecto. Se centra en la colaboración, la comunicación y la entrega incremental.
* **Lean Software Development.** Inspirado en los principios de Lean Manufacturing, se enfoca en eliminar el desperdicio y optimizar el flujo de trabajo. Prioriza la entrega rápida y la mejora continua.
  1. **Aparición y descripción general de los métodos ágiles**

El término ágil representa la versatilidad y adaptabilidad a las situaciones que acontecen los proyectos, productos y servicios que ofrece la organización. La agilidad se debe entender como una forma de pensar que aborda herramientas potencialmente beneficiosas para las organizaciones.

Ante demandas y expectativas cada vez más abrumadoras, los directores de proyecto que usaban metodologías tradicionales (las que controlan y anticipan una anticipación completa) han tenido que agilizar sus proyectos de manera progresiva y así poder cubrir las expectativas del mercado.

Las metodologías ágiles están basadas en las planificaciones iterativas donde se realizan entregas tempranas para incrementar el calor progresivamente. Así se procura poder adaptar el producto a las expectativas del cliente con cada iteración. Es necesaria la comunicación entre todos los interesados.

Los proyectos ágiles son un conjunto de metodologías dirigidas al desarrollo de proyectos que requieren rapidez y flexibilidad en su proceso. Suelen estar vinculados con el desarrollo de software y el ámbito digital.

Las metodologías ágiles nacieron gracias al dinamismo del mundo empresarial contemporáneo. En proyectos relacionados con la informática, no se obtenían resultados óptimos con las metodologías tradicionales ya que requerían cambios constantes, comunicación continua y menos planificación.

* 1. **El Manifiesto agile**

Está comprometido con cuatro valores fundamentales.

* **Individuos e interacciones por encima de procesos y herramientas.** Valorar a las personas por encima de los procesos es esencial porque son ellas quienes responden a las necesidades del negocio y dirigen el desarrollo. Si el proceso o las herramientas conducen el desarrollo, el equipo es menos sensible al cambio y tiene menos probabilidad de conocer las necesidades del cliente. La comunicación es un ejemplo claro: con los individuos, la comunicación es fluida y ocurre cuando surge una necesidad; con el proceso, la comunicación se programa y requiere de un contenido específico.
* **Software que funcione por encima de una documentación extensiva.** Se ha gastado una enorme cantidad de tiempo en documentar el producto para el desarrollo y la última entrega de un producto. El listado de documentación necesaria es enorme y era una de las causas de los grandes retrasos en el desarrollo de un producto. Ágil no elimina la documentación, sino que la reestructura de forma que proporciona al desarrollador lo que necesita esencialmente para hacer el trabajo. Ágil documenta los requisitos como historias de usuario, que son suficientes para empezar la tarea de construir una nueva funcionalidad. Se valora la documentación, pero más el software que funciona.
* **Colaboración con el cliente por encima de la negociación contractual.** Con modelos de desarrollo como waterfall, los clientes negocian los requisitos del producto, a menudo con mucho detalle, antes de empezar cualquier trabajo. Así, el cliente se involucra en el proceso de desarrollo antes y después, pero no durante el proceso explícito de desarrollo. En Ágil el cliente está comprometido y colabora en el proceso de desarrollo, haciendo más fácil el poder satisfacer sus necesidades. Pueden incluir al cliente en demostraciones periódicamente, por lo que tener a un usuario final como parte del equipo y asistiendo a todas las reuniones puede asegurar que el producto cumple con las necesidades.
* **Respuesta ante el cambio en lugar de seguir un plan.** En el desarrollo de software tradicional, los cambios se evitan porque se consideran costosos. La intención es desarrollar de forma detallada y elaborar planes con un conjunto definido de funcionalidades, prioridades y un gran número de dependencias en la entrega. Con ágil, el tiempo de una iteración significa que las prioridades pueden ser empaquetadas de iteración en iteración y se pueden añadir nuevas funcionalidades en la siguiente iteración. Los cambios siempre mejoran un proyecto y proporcionan un valor adicional al producto. Los participantes deben estar preparados para realizarlos.
  1. **Aproximación tradicional vs agile**

Una aproximación agile al proyecto comprende cambiar el foco del Project Manager, su equipo y los distintos elementos que comprenden sus trabajos y tareas. Al desarrollar un sitio web, el cliente preferirá tener alguna página relacionada con su web lo antes posible, aunque el resto no estén completas. Si esperamos a tener todas hechas, no tendrá nada que mostrar en su sitio web. La aproximación agile a este proyecto se centra en aportar valor al cliente lo antes posible, mediante la entrega de partes 100% funcionales del entregable final.

* 1. **Aproximación al marco Scrum**

En las metodologías ágiles buscan abordar las restricciones de otra forma, manteniendo fijo el coste y el tiempo, variando solo el alcance (en waterfall ocurre al revés). El alcance se itera con el cliente para buscar la mejor versión que se pueda entregar. El entorno colaborativo y la retroalimentación forman el eje fundamental de esta forma de pensar y desarrollar proyectos.

Scrum es un marco de gestión flexible diseñado para entregar rápidamente valor considerable a lo largo del proyecto. Pretende garantizar transparencia en la comunicación y crear un ambiente de responsabilidad colectiva. El marco Scrum está estructurado de tal manera que es compatible con el desarrollo de productos y servicios en todo tipo de industrias y en cualquier tipo de proyecto, sin importar su tamaño y complejidad.

Una fortaleza clave de Scrum radica en el uso de sprints, ciclos de trabajo cortos en los que se divide el trabajo.

Tres pilares de Scrum:

* **Transparencia:** Todo el equipo debe saber dónde estamos y a dónde vamos.
* **Inspección:** Se revisa siempre el trabajo realizado para identificar mejoras.
* **Adaptación:** Se cambia lo planificado si es necesario para alcanzar los objetivos.

1. **GESTIÓN DE EQUIPOS ÁGILES**

El enfoque Scrum funciona bien con Agile porque ayuda a los miembros del equipo a abordar problemas complejos y a lograr entregas productivas y creativas con el máximo valor posible. Permite mejoras en el futuro si es necesario y funciona muy bien para gestionar proyectos complejos y desafiantes. Refuerza la productividad:

* Aumenta la capacidad de reacción ante cambios repentinos.
* Reduce el tiempo que se tarda en lanzar algo al mercado.
* Se obtiene mayor calidad de software.
* Mejores predicciones de tiempos.
* Reducción de riesgos.

Hay tres grandes conjuntos de elementos en esta metodología: El equipo Scrum, Los eventos Scrum y Los artefactos Scrum.

* 1. **Los roles del equipo de proyecto en Scrum**

Existen tres roles principales:

* **Product Owner:** Orientado al negocio.
* **Scrum Master:** Centrado en que el equipo sea lo más productivo posible.
* **Development Team:** Centrado en el desarrollo de funcionalidades.

Otros acrónimos importantes son Stakeholder, Expertos en la materia (SME) y Business Owner.



* + 1. ***Product Owner***

Tiene la responsabilidad de definir el trabajo y priorizar las tareas correspondientes en el proyecto. Es crucial que tenga un profundo conocimiento de los objetivos del proyecto, del cliente, del mercado y de la organización en su conjunto. Guía al equipo a lo largo de todo el proceso de desarrollo del proyecto a través del product backlog, que es un listado de funcionalidades del producto ordenadas por prioridad.

Las responsabilidades del Product Owner incluyen:

* Describir los elementos que conforman la lista del producto.
* Ordenar los elementos en la lista del producto para alcanzar los objetivos de la mejor manera posible.
* Maximizar el valor del trabajo realizado por el equipo de desarrollo.
* Garantizar que la lista del producto sea viable, transparente y clara para todos.
* Asegurar que el equipo de desarrollo comprende los elementos de la lista del producto.

El Product Owner debe hacer que managers de otras áreas no puedan pedir tareas al equipo de desarrollo sin antes pasar por su filtro. Así se evita que se soliciten tareas que no sean prioritarias o que les hagan perder el foco del sprint. También es el encargado de maximizar y optimizar el producto. Debe hacer que la visión del producto esté clara para todo el equipo Scrum y de esta forma lograr mantener el foco y entregar la mejor versión de la iteración.

Aspectos clave:

* Validar los incrementos frecuentemente con los interesados del proyecto y el cliente en particular y recibir información en tiempo real para ajustar las debidas modificaciones.
* El product owner debe estar siempre enfocado en priorizar las funcionalidades más valiosas; sin embargo, no debe dejar a un lado las funcionalidades de baja prioridad ya que estas pueden terminar desmotivando al equipo o a cierta parte de los interesados.
* El product owner tiene la última palabra en la priorización de las funcionalidades y las interdependencias entre ellas; sin embargo, este debe escuchar las recomendaciones del equipo Scrum para obtener el mejor resultado.
  + 1. ***El Scrum Master***

Garantiza que el equipo Scrum trabaje de acuerdo con los principios, prácticas y reglas de la metodología, y que estos se comprendan y adopten completamente. Guía al equipo y asegura que el dueño del producto y el equipo de desarrollo trabajen dentro del marco de Scrum.

Lo consigue gracias a:

* Eliminar los impedimentos que nos podamos encontrar durante el proyecto.
* Facilitar las reuniones de Scrum para que sean productivas y se consigan sus objetivos.
* Enseñar al equipo a autogestionarse.
* Trabajar con el product owner para asegurar que el product backlog está en buena forma y preparado para el siguiente sprint.

Es responsable de diferentes servicios y responsabilidades que varían según los otros dos roles principales en el proyecto.

Tareas que se espera del Scrum Master:

* Entender la planificación del producto y encontrar las técnicas necesarias para gestionar efectivamente la lista del producto. Asegurarse de que todos los miembros implicados conocen el marco de trabajo.
* Guiar al equipo de desarrollo para que este se autoorganice, sea multifuncional y elimine los impedimentos que pueden ocurrir en el proceso.
* Asegurar que el dueño del producto sepa cómo priorizar la lista del producto y que el equipo entienda la necesidad de contar con elementos claros y concisos para el trabajo.
* Facilitar los eventos de Scrum que sean requeridos.
* Planificar la implementación de Scrum en la organización, además de liderar y guiarla en el proceso de adopción del marco de trabajo.
* Eliminar los impedimentos que pueden ocurrir en el proceso.
  + 1. ***El Development Team***

Conjunto de profesionales encargados de crear y entregar un producto finalizado, que potencialmente se puede poner en producción, al final de cada sprint. Es el único que participa en el desarrollo del producto.

Es estructurado por la organización para coordinar y gestionar su propio trabajo, optimizando así la eficiencia y efectividad de su trabajo en conjunto. Es responsable de llevar a cabo el trabajo real del proyecto, llevándolo más allá de la planificación propuesta y convirtiéndolo en algo tangible. Cada miembro del equipo tiene habilidades específicas que, junto con las de otros miembros, se combinan para abordar cualquier actividad que se presente. El equipo trabaja de manera colaborativa, actúa colectivamente y se esfuerza por descubrir cómo lograr sus objetivos.

El tamaño óptimo del equipo de desarrollo es aquel que es lo suficientemente pequeño para mantener la agilidad, pero lo suficientemente grande para completar una cantidad significativa de trabajo. Si tiene menos de tres miembros puede haber limitaciones en cuando a las habilidades necesarias para un sprint, pudiendo retrasar la entrega y pudiendo reducir la interacción y producir ganancias de productividad limitadas. Si tiene más de nueve miembros, puede requerir demasiada coordinación y generar demasiada complejidad como para ser gestionados adecuadamente mediante un proceso empírico.

El tamaño del equipo de desarrollo debe estar entre tres y nueve personas, sin tomar en consideración al product owner ni al Scrum master. Si se da el caso de requerir que el equipo sea mayor a nueve personas, se debe reconsiderar la estrategia para dividir en más equipos.

Características:

* **Son auto-organizados:** Completamente autónomo. Ni el Scrum master indica al equipo cómo desarrollar cada tarea. El equipo es responsable de llevar a cabo su trabajo y de cómo lo realizan.
* **Todos los miembros se consideran en la misma posición:** Todos los miembros son desarrolladores y no se les asigna otra función o título, independientemente del trabajo que realicen. No se permiten subequipos.
* **Son multifuncionales:** El equipo dispone de las habilidades adecuadas sin depender de agentes externos para proporcionar el incremento de producto esperado.
* **La responsabilidad se comparte:** Si bien cada miembro del equipo de desarrollo puede tener habilidades especializadas y desenvolverse mejor en algunas áreas, la responsabilidad final recae en el equipo en su conjunto.
  + 1. ***Roles auxiliares en Scrum***

Son dos:

* **Stakeholders:** Son las personas que hacen posible el proyecto y para quienes este producirá el beneficio acordado que justifica su desarrollo. Su participación directa solo es necesaria durante las revisiones del sprint. Son los clientes, proveedores y vendedores.
* **Administradores:** También se les dice managers, y son los responsables de establecer el entorno para el desarrollo del proyecto.
  + 1. ***La dinámica de un equipo Scrum***

La dinámica del equipo Scrum se puede equiparar con la dinámica que se mantiene en un equipo de rugby, más específicamente con una jugada llamada «melé». Si un miembro del equipo se viene abajo, se cae toda la melé; en consecuencia, los jugadores deben estar bien coordinados, apoyarse en sus compañeros para empujar al mismo tiempo y, con ello, avanzar a la misma velocidad.

Esto mismo sucede con Scrum, cada uno de los roles que mencionamos anteriormente forma parte de la melé, y pone su trabajo al servicio del resto del equipo, que sería el proyecto, para obtener el mismo objetivo, que en el rugby es el balón, pero en la gestión de proyectos son los beneficios que se obtienen.



1. **LOS EVENTOS DE SCRUM**
   1. **Sprint**

Base del scrum. Se lleva a cabo en periodos de tiempo cortos (un mes como máximo y un promedio de 2-3 semanas). Es el momento de crear un incremento de producto, utilizable y potencialmente liberable.



El sprint se utiliza para lograr algo; tiene una definición de lo que se va a desarrollar, un diseño y un plan flexible que guiarán su construcción, el trabajo y el producto final. Al alcanzar los objetivos, un nuevo sprint comenzará.

* No es correcto realizar modificaciones que pongan en peligro el objetivo que se ha de conseguir.
* No se pueden disminuir los objetivos de calidad.
* El alcance puede ser renegociado con el dueño del producto y el equipo de desarrollo a medida que se descubren nuevas condiciones, pero el tiempo máximo del sprint no es negociable.
  + 1. ***La cancelación de un sprint***

Puede ser cancelado antes de que el bloque de tiempo pautado llegue a su fin, pero solo el dueño del producto tiene la autoridad para cancelarlo. Lo más común es que se cancele si este queda obsoleto antes de completarse, si la compañía cambia de dirección y enfoque, o si cambian las condiciones del mercado y tecnología.

Aspectos más relevantes de la cancelación:

* Los elementos terminados (done) pueden ser puestos a disposición del cliente y el resto de ellos volverá a la pila del producto.
* Los elementos que no se han podido completar deben volver a la pila del producto, donde se evaluará su prioridad o se descartarán de no ser requeridos para los objetivos del producto.
* Si un sprint es cancelado, se debe poner a disposición del cliente aquello que se encuentre terminado (done) e iniciar inmediatamente el nuevo sprint. Las cancelaciones de sprint consumen recursos, ya que todos deben reagruparse en otra reunión de planificación para empezar otro proceso. Además, generan malestar en el equipo scrum en general.
  1. **Sprint planning**

Su objetivo es definir lo que se va a hacer en el sprint y cómo se va a hacer. Dura 8 horas como máximo para sprints de un mes; si la duración del sprint es menor, la reunión de planificación también se reduce.

Un sprint planning meeting para un sprint de duración de dos semanas puede ser realizado incluso entre 60 y 45 minutos si las funcionalidades están bien definidas y son priorizadas para que puedan ser estimadas fácilmente por el equipo scrum.

Participan todos los miembros del equipo scrum, pero es el scrum master quien se asegura de que el evento tenga lugar y los asistentes entiendan su propósito, así como de que la reunión no exceda el tiempo de duración. En esta reunión se responden dos preguntas: ¿Qué puede entregarse en el incremento resultante del sprint que comienza? ¿Cómo se conseguirá hacer el trabajo necesario para entregar el incremento?

* + 1. ***Origen del contenido que realizar en el sprint***

**\* Product Backlog**

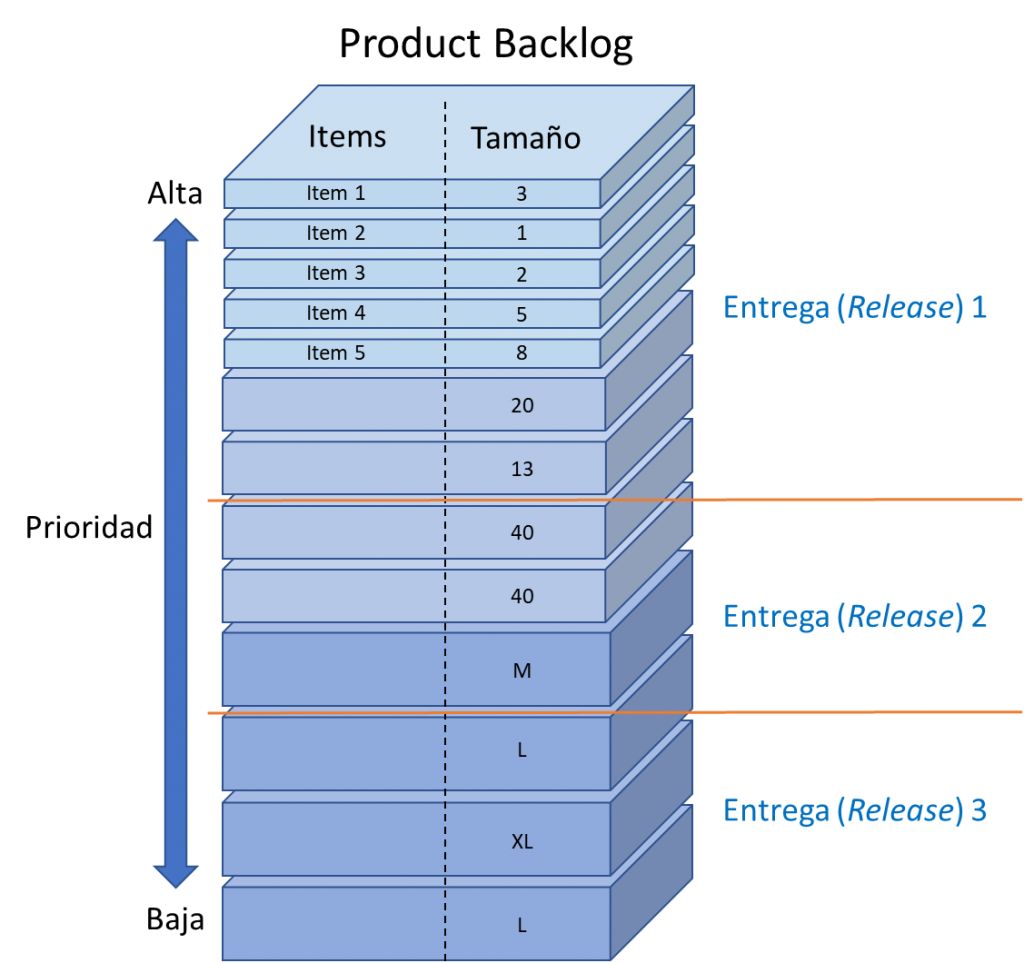
Listado de funcionalidades priorizadas, que contienen pequeñas descripciones de todas las funcionalidades deseadas en el producto. La persona que prioriza y trabaja en el product backlog es el producto owner, aunque un equipo scrum y el product owner son los encargados de identificar funcionalidades que pueden ser parte del producto. Luego, el product owner es el encargado de priorizarlas acorde con lo que los clientes están demandando del producto.

Una vez que el equipo scrum y el product owner han identificado funcionalidades y el product owner las ha priorizado, ya es casi seguro que tendremos más funcionalidades de las que van a caber en el primer sprint, por lo que el equipo scrum ya puede empezar a trabajar.

Elementos:

* **Funcionalidades:** Son propiamente las historias de usuario.
* **Bugs:** Tanto los reportados por el cliente como los encontrados por el equipo de calidad.
* **Épicas:** Las épicas son historias de usuario muy grandes que no pueden finalizarse en un sprint, por lo que necesitan que se descompongan en otras historias de usuario con un tamaño mucho más adecuado para ser gestionadas con los principios y técnicas ágiles: estimación y seguimiento cercano.
* **Trabajo de investigación o adquisición de conocimientos:** Muchas veces, para realizar una historia de usuario hay que tener un profundo conocimiento de a qué nos enfrentemos, y scrum aconseja valorar el trabajo que vas a realizar en materia de estudio/investigación.

Las funcionalidades de máxima prioridad (las que están arriba del todo en el backlog y entrarán en el siguiente sprint) tienen que estar estimadas, deben tener una valoración sobre el peso o el tamaño que tiene esa historia de usuario.



**\* Sprint Backlog**

Lista de tareas identificadas por el equipo para que se completen durante el sprint. Durante la reunión del sprint planning, el equipo selecciona un número de elementos del backlog e identifica las tareas necesarias para completar cada historia de usuario. La mayoría de los equipos suelen estimar también el número de horas que cada tarea requerirá a un miembro del equipo para completarla.

Es fundamental que el equipo seleccione los elementos y estime un buen sprint backlog, ya que sus miembros son los que se comprometen a completar las tareas y demostrar esas funcionalidades al product owner al final del sprint.

CONSIDERAR ESCRIBIR QUÉ SON LAS HISTORIAS DE USUARIO Y LOS PUNTOS DE HISTORIA (COMPLEJIDAD Y PRIORIDAD)

* + 1. ***¿Qué puede entregarse en este sprint?***

En esta sección de la planificación se deben definir y acordar aquellos elementos del product backlog que dentro de un sprint se pueden completar o llegar a su fase done.

Durante el sprint los miembros del equipo deben actualizar el sprint backlog con la nueva información disponible una vez al día. El scrum master puede verificar con estos datos el progreso que el equipo lleva de cada tarea de esa historia de usuario. Se suelen demorar hasta cinco sprints en lograr que sus planificaciones se acerquen a lo que se entrega al final de este. Los primeros sprints serán de reconocimiento, para afinar detalles y llegar al rendimiento óptimo del equipo.

Aspectos relevantes:

* Para seleccionar las funcionalidades que se van a desarrollar se deben considerar el tamaño de los elementos y su prioridad. Se debe entender la cantidad de días laborables que existen dentro del sprint en conjunto al rendimiento del equipo en los incrementos pasados.
* La planificación es uno de los eventos fundamentales del sprint y debe hacerse tomando en consideración los datos del rendimiento real.
  + 1. ***¿Cómo se puede entregar el trabajo seleccionado?***

Se responde una vez se ha realizado la primera parte de la planificación; el equipo de desarrollo establece cómo desarrollará los elementos que han sido seleccionados para el sprint. Para esto se tienen en consideración los siguientes aspectos:

* El equipo debe descomponer en unidades pequeñas las funcionalidades; el objetivo de esto es que se puedan manejar actividades con duración de un día o menos y, así, monitorear el avance de sprint.
* La labor de cómo desarrollar el sprint y todas las funcionalidades seleccionadas corresponde al equipo de desarrollo; este, de forma autoorganizada, elabora el camino que se ha de seguir.
* El product owner es el responsable de transmitir al equipo de desarrollo lo que se espera de cada uno de los elementos que desarrollarán en el sprint. De esta forma, el equipo puede entender cuánto se debe cubrir y hasta qué punto son capaces de desarrollarlo en solo incremento.
* Si el equipo de desarrollo detecta, al desglosar las actividades, que no puede cubrir todos los elementos seleccionados, es un buen momento para eliminar las funcionalidades que puedan esperar y elaborar una planificación más real.
  + 1. ***Objetivo del sprint o sprint goal***

Meta establecida que puede ser alcanzada mediante la implementación de la lista de producto. Creado durante la reunión de planificación del sprint, y ofrece al equipo de desarrollo cierta flexibilidad con respecto a la funcionalidad implementada en el proceso de creación.

A medida que el equipo de desarrollo trabaja, se mantiene el objetivo del sprint en mente, y con el fin de satisfacerlo se implementan la funcionalidad y la tecnología.

Aspectos del sprint goal:

* Se debe tener un objetivo claramente definido que esté sobre todas las tareas y que englobe las funcionalidades que se van a desarrollar durante el sprint.
* El objetivo debe ser el centro del trabajo y foco de todo el equipo.
* Si en determinado caso, durante el desarrollo del sprint, el equipo de desarrollo determina que no es capaz de alcanzar el objetivo, lo correcto es renegociar el sprint backlog con el product owner y alcanzar una solución beneficiosa.

Se utiliza para informar a los stakeholders sobre el progreso del sprint, sobre lo que está trabajando el equipo, ya que muchas veces los clientes quieren saber o incluso necesitan escuchar en detalle lo que el equipo está trabajando.

* 1. **Daily meeting**

El daily scrum es una reunión con un bloque de tiempo de quince minutos para que el equipo de desarrollo sincronice sus actividades y cree un plan para las siguientes 24 horas, partiendo de una inspección del trabajo realizado desde el último scrum diario y haciendo una proyección acerca del trabajo que podría completarse antes del siguiente. Se realiza a la misma hora y en el mismo lugar todos los días para reducir la complejidad. Cada miembro del equipo de desarrollo da respuesta a las siguientes preguntas:

* ¿Qué hice ayer que ayudó al equipo de desarrollo a lograr el objetivo del sprint?
* ¿Qué haré hoy para ayudar al equipo de desarrollo a lograr el objetivo del sprint?
* ¿Veo algún impedimento que evite que el equipo de desarrollo o yo logremos el objetivo del sprint?

El equipo de desarrollo utiliza el scrum diario para medir y evaluar el progreso hacia el objetivo del sprint y su tendencia a la finalización de la lista de tareas. Esto optimiza las posibilidades de cumplir con los objetivos. El Scrum Master garantiza que se realice la reunión, pero el equipo es responsable de dirigirla. A veces, hay situaciones en las que los equipos eligen hacer dos daily meetings. Esto tiene un impacto sumamente positivo, ya que ayuda a los miembros del equipo a mantenerse en sincronía.

El scrum diario mejora la comunicación, identifica y suprime impedimentos, promueve la toma de decisiones rápidas, y mejora el nivel de conocimiento del equipo. Es fundamental para aumentar la productividad de los miembros del equipo colaborando entre ellos. Cada miembro puede ver las tareas que realizan otros para poder hacer las adaptaciones necesarias y presentar un trabajo de calidad al cliente.