

METODOLOGÍA ÁGIL SCRUM

Scrum es un marco de trabajo ágil que se utiliza en el desarrollo de software. Se basa en principios iterativos e incrementales para gestionar el trabajo de manera colaborativa.

En Scrum, se utilizan las metáforas de "gallinas" y "cerdos" para describir diferentes niveles de compromiso con el proyecto:

1. Gallinas: Las "gallinas" son **personas que están involucradas en el proyecto pero no son parte del Equipo de Desarrollo**. Esto incluye al Product Owner, al Scrum Master, a los stakeholders y a cualquier otra persona interesada en el proyecto. Las gallinas tienen interés en el éxito del proyecto pero no están directamente involucradas en la realización del trabajo.

2. Cerdos: Los "cerdos" son miembros del Equipo de Desarrollo que están completamente comprometidos con el proyecto. Esta metáfora proviene de la historia del desayuno: mientras que las gallinas solo proporcionan los huevos (compromiso), los cerdos proporcionan el jamón (compromiso total). Los cerdos son responsables de realizar el trabajo durante el Sprint y de entregar el incremento del producto.

Por lo tanto, las gallinas tienen un interés indirecto en el proyecto y pueden aportar ideas y feedback, pero no están directamente involucradas en la ejecución del trabajo. Los cerdos, por otro lado, son los que llevan a cabo el trabajo y están completamente comprometidos con el éxito del proyecto.

1. Roles: Scrum define tres roles principales: el **Product Owner**, que representa los intereses del cliente y prioriza el trabajo en el **Backlog** del **Producto**; el **Scrum Master**, que facilita el proceso y elimina los obstáculos que puedan surgir; y el **Equipo de Desarrollo**, que es autoorganizado y multifuncional.

- **Scrum Master:** responsable de garantizar que el equipo Scrum comprenda y adopte los principios y prácticas de Scrum. Facilita el proceso Scrum y elimina los obstáculos que impiden el progreso del equipo.
- **Product Owner:** responsable de maximizar el valor del producto y del trabajo del equipo de desarrollo. Define las características del producto, prioriza el backlog del producto y asegura que el equipo trabaje en las áreas de mayor valor.
- **Development Team:** grupo de profesionales que llevan a cabo el trabajo necesario para entregar un incremento de producto potencialmente entregable al final de cada Sprint. Son autoorganizados y multidisciplinarios.

2. Eventos/Reuniones: Scrum tiene varios eventos predefinidos para fomentar la colaboración y la transparencia en el equipo. Estos incluyen la **Planificación del Sprint**, la **Revisión del Sprint**, la **Retrospectiva del Sprint** y las **reuniones diarias de Scrum**.

- **Sprint Planning:** Durante esta reunión, el equipo Scrum planifica el trabajo a realizar durante el próximo Sprint. Discuten y seleccionan los elementos del backlog del producto que trabajarán y establecen un objetivo para el Sprint.

- **Daily Scrum:** Esta es una reunión diaria de 15 minutos en la que el equipo Scrum se reúne para sincronizarse y planificar el trabajo del día. Cada miembro del equipo responde a tres preguntas: ¿Qué he hecho desde la última reunión diaria?, ¿Qué voy a hacer hasta la próxima reunión diaria?, ¿Hay algún impedimento que me impida avanzar en mi trabajo?
- **Sprint Review:** Al final de cada Sprint, el equipo Scrum realiza una revisión del Sprint para demostrar el trabajo completado al Product Owner y a los stakeholders. Discuten lo que se completó durante el Sprint y obtienen retroalimentación sobre el trabajo realizado.
- **Sprint Retrospective:** Después de la revisión del Sprint, el equipo Scrum realiza una retrospectiva del Sprint para analizar qué salió bien, qué se puede mejorar y qué acciones se pueden tomar para el próximo Sprint. Es una oportunidad para la mejora continua del equipo.

3. Elementos: elementos tangibles que se utilizan para gestionar el trabajo. El más importante es el **Backlog del Producto**, que contiene todas las funcionalidades, mejoras y correcciones que se planea desarrollar. También está el **Backlog del Sprint**, que contiene las tareas seleccionadas para el próximo Sprint.

- **Product Backlog (Diana):** Es una lista ordenada y priorizada de todas las funcionalidades, requisitos, mejoras y correcciones pendientes que se desean para el producto. Es responsabilidad del Product Owner mantener y gestionar el Product Backlog.
- **Sprint Backlog:** Es una lista de elementos seleccionados del Product Backlog que el equipo se compromete a completar durante un Sprint específico. El Sprint Backlog se crea durante la reunión de Sprint Planning y se actualiza durante el Sprint.
- **Incremento (Increment):** Es el resultado tangible y potencialmente entregable del trabajo completado durante un Sprint. El Incremento es una versión incrementada y mejorada del producto que se puede presentar y demostrar en la revisión del Sprint.
- **Definition of Done (DoD):** Es un conjunto de criterios que definen cuándo un elemento del Product Backlog está completo y "hecho". Estos criterios se establecen por el equipo Scrum y suelen incluir aspectos de calidad, pruebas, documentación y otros requisitos específicos.

4. Sprints: Un Sprint es un período de tiempo fijo durante el cual se desarrolla un conjunto de funcionalidades o mejoras. Los Sprints suelen tener una duración de dos a cuatro semanas, y al final de cada Sprint se entrega un incremento potencialmente entregable del producto.

Las entregas de el proyecto será siguiendo el proceso de "entrega continua" y utilizando metodología Scrum, es decir, a finalizar cada Sprint se debe entregar una parte del proyecto.

La entrega continua es un enfoque en el desarrollo de software que implica la automatización de los procesos de construcción, pruebas y despliegue, con el objetivo de permitir la entrega rápida y frecuente de software de alta calidad. En el contexto de Scrum, la entrega continua se integra en el ciclo de desarrollo ágil para garantizar que el software desarrollado esté listo para ser entregado al cliente al final de cada iteración o sprint.

1. Automatización de la construcción y las pruebas: Utilizando herramientas como Jenkins, Travis CI o GitLab CI, los equipos pueden automatizar la compilación del código, las pruebas unitarias, las pruebas de integración y otras pruebas de calidad.

2. Integración continua: Los cambios de código se integran automáticamente en un repositorio compartido varias veces al día. Esto garantiza que los problemas de integración se detecten y solucionen rápidamente.

3. Despliegue continuo: Los cambios de código que pasan todas las pruebas automáticamente se despliegan en un entorno de producción o de pruebas, lo que permite obtener retroalimentación rápida sobre el software desarrollado.