

METODOLOGIAS AGILES

SCRUM PARA PRINCIPIANTES

Introducción a una de las
metodologías ágiles más
populares del mercado





Índice

Metodologías Ágiles

- > Ventajas de implementar metodologías ágiles sobre metodologías tradicionales
- > ¿Cuáles son las metodologías ágiles más conocidas?

Scrum

Principios

- > Transparencia
- > Inspección
- > Adaptabilidad

Eventos

- > Sprint
- > Sprint Planning
- > Daily Scrum
- > Sprint Review
- > Sprint Retrospective
- > Sprint Goal
- > Definición de Terminado

Roles

- > Product Owner
 - Listado de responsabilidades del PO
- > Scrum Master
 - Listado de responsabilidades del SM
- > Development Team
 - Listado de responsabilidades del DT

Roles externos

- > Stakeholder

Artefactos

- > Sprint Backlog
- > Incremento
- > Product Backlog

Herramientas

- > Tablero Scrum
 - Partes de un tablero
- > Historias de Usuario
 - ¿Quién define la Historia de Usuario?
 - Formato de una Historia de Usuario
 - Criterio de aceptación

Malas prácticas aplicando Scrum

- > Sprint Retrospective
- > Daily Scrum
- > Sprint Review

METODOLOGÍAS ÁGILES

Por definición, son aquellas metodologías que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del mercado y los clientes. Los pilares fundamentales de las metodologías ágiles son el trabajo colaborativo y en equipo.

> Ventajas de implementar metodologías ágiles sobre metodologías tradicionales


- Gracias a la flexibilidad y capacidad de adaptación de las mismas, son muchos los beneficios de incorporar metodologías ágiles a la gestión de las organizaciones. Aquí detallamos algunos:

- Reducción de costos.
- Rapidez en la entrega de proyectos.
- Trabajo en equipo y compromiso de todos los miembros del equipo de trabajo.
- Mayor calidad en el trabajo y en el producto final (ya sea producto o servicio).



Mientras que la metodología tradicional se maneja de manera lineal y es reticente a los cambios; la metodología ágil se compone de procesos iterativos, los cuales optimizan la previsibilidad y reducen los riesgos.

Además, permite una mejora en los tiempos de planificación, revisión y flexibilidad ante las modificaciones respecto de las especificaciones iniciales, esto lo convierte en una buena metodología para trabajar en proyectos pequeños y de corta duración, aportando valor de modo continuo; contrariamente a lo que ocurre en las metodologías tradicionales, donde el modo de trabajo puede generar que de producirse un error muy grande haya que volver a iniciar, que para cuando el producto está listo para lanzarse ya este no sea necesario o la competencia haya lanzado algo similar.




El hecho de lanzar el producto terminado hace que recién al final del proceso se retorne la inversión, por el contrario, en Agile, se logra a lo largo de la vida del proyecto desde la publicación de la primera etapa.

Un mayor involucramiento del cliente y una organización en equipo pequeños, promueve el feedback directo y frecuente, aclarando ideas, requerimientos, dificultades y evitando desviarse del objetivo, esto genera entre los involucrados un mayor compromiso y motivación. Las metodologías tradicionales son más individualistas y competitivas, donde el cliente delega la responsabilidad y los roles se dividen de modo jerárquico y especializado.

ASPECTOS	ÁGILES	TRADICIONALES
Método de trabajo	Iterado	Lineal
Flexibilidad a los cambios	Puede modificarse en el transcurso del proyecto	Hay que empezar de nuevo
Cliente	Muy involucrado	Delega la responsabilidad
Documentación	Poco documentado	Muy documentado
Equipo	Pequeños	Grandes
Tipo de proyecto	Difícil de implementar en proyectos grandes y/o complejos	Cualquier tipo de trabajo
Feedback	Constante	No hay
Retorno de inversión	A lo largo de la vida del proyecto	Al finalizar el proyecto

➤ **¿Cuáles son las metodologías ágiles más conocidas?**

Algunas de las metodologías ágiles más conocidas son : **Lean, Kanban, Scrum**, entre otras.



SCRUM

Es una metodología para gestión, mejora y mantenimiento de un sistema nuevo o existente, se concentra en cómo los miembros del equipo deberían funcionar a fin de producir un sistema flexible en un entorno que cambia constantemente.

Scrum se basa en el empirismo, donde el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo.

METODOLOGÍA SCRUM

ES DEL EQUIPO SCRUM: Dueño del Producto, Facilitador del Proyecto y Equipo de Desarrollo

EFACTOS: Listado de Tareas, Lista de Tareas del Sprint y el Incremento

NTOS: Reunión de Planificación del Sprint, Sprint, Scrum diario, Retrospectiva del Sprint, Revisión del Sprint





PRINCIPIOS

TRANSPARENCIA

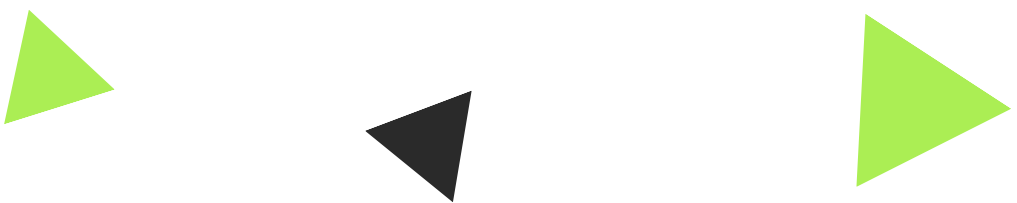
Se refiere a saber qué es lo que estamos haciendo, los aspectos significativos del proceso tienen que ser visibles para los responsables del resultado. Estos aspectos tienen que estar definidos por un estándar común, de manera que todos compartan un entendimiento de lo que se está viendo.

INSPECCIÓN

Los usuarios de Scrum deben inspeccionar frecuentemente los Artefactos de Scrum y el progreso hacia el objetivo del Sprint, para detectar rápidamente desviaciones. Estas inspecciones no deben ser tan frecuentes como para no interferir en el trabajo.

ADAPTABILIDAD

Si una inspección resulta en la detección de una desviación del objetivo que va más allá de los límites aceptables (que el producto ya no cumpla con el criterio de “Terminado”) se deben hacer ajustes cuanto antes para minimizar desviaciones mayores.



EVENTOS

SPRINT

Es el corazón del Scrum, un ciclo de un mes o menos donde se puede terminar y entregar un incremento del producto. Los Sprints contienen y consisten en la Planificación (Sprint Planning), los Scrums Diarios (Daily Scrums), el trabajo de desarrollo, la Revisión del Sprint (Sprint Review), y la Retrospectiva (Sprint Retrospective).

Cada Sprint se puede considerar un proyecto no mayor a un mes al cabo del cual se obtiene un producto entregable (incremento), un Sprint comienza inmediatamente después que el anterior termina.

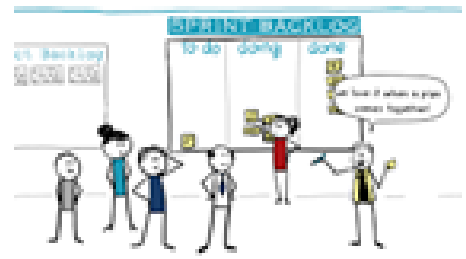


SPRINT PLANNING

Planificación del Sprint, es un evento en donde se decide qué ítems se van trabajar durante el Sprint, quién será el responsable de cada ítem y cuál es el objetivo.

La planificación debe contestar las siguientes preguntas:

- ¿Qué es factible de ser entregado en el siguiente incremento al final del Sprint?
- ¿Mediante qué acciones se logrará dicho incremento?



DAILY SCRUM

El Scrum Diario o Daily Scrum es una reunión de 15 minutos donde todos los miembros del equipo tengan una visión general del progreso y lo que falta por realizar.

Para ello, cada uno debe responder de manera breve y concisa a tres preguntas:

- **¿Qué hiciste ayer?** Para poner al tanto de los progresos
- **¿Qué vas a hacer hoy?** Para plantear los objetivos
- **¿Hay algún obstáculo?** Para informar elementos que puedan intervenir con el progreso.



SPRINT REVIEW

Es un evento informal que se basa en el producto, y en el cual el Product Owner (PO) analiza el estado del Product Backlog y lo actualiza en base a los avances y dificultades que haya reportado el Development Team (DT).



SPRINT RETROSPECTIVE

Es un evento en el que se analiza el funcionamiento del equipo como tal y se planean mejoras en caso de considerarse necesarias.



SPRINT GOAL

El Sprint Goal responde a la pregunta:

“¿Para qué hacemos este Sprint?”

Es el objetivo más importante, un compromiso de las tareas a realizar que sirve de guía al DT para trabajar sobre un objetivo en común y no de manera individual



DEFINICIÓN DE TERMINADO

Es un término que debe ser entendido y acordado por todos los miembros del Equipo Scrum. Se utiliza para evaluar cuándo se ha completado el trabajo sobre el Incremento de Producto.

También se debe tener en cuenta que el propósito de cada Sprint se debe ajustar a la definición de “Terminado”.

Si hay múltiples Equipos Scrum trabajando en la entrega del sistema o producto, el DT en todos los Equipos deben definir en conjunto la definición de “Terminado”.



ROLES

PRODUCT OWNER

Product Owner (PO) es el responsable de gestionar uno de los artefactos de la metodología (Scrum) conocido como Product Backlog y de maximizar el valor del trabajo realizado por el equipo de desarrollo (Equipo Scrum).

Es el representante del cliente dentro del equipo, su voz, representa a las partes interesadas internas y es responsable de entregar el valor más alto posible al negocio.




Listado de responsabilidades del PO

- Conocer el mercado y los comportamientos de los Clientes/Usuarios Finales, con muy buena visión de Negocio.
- Ser el representante de todas las personas interesadas (Stakeholders) para conseguir una buena definición de los objetivos del producto o proyecto y de los resultados esperados.
- Estar centrado en la creación de valor y la generación del ROI.
- Actuar como interlocutor único ante el equipo, con autoridad para tomar decisiones.
- Determinar los requisitos generales y actividades iniciales del proyecto.
- Representar a los usuarios del producto.
- Buscar y asegurar los recursos financieros que requiere el proyecto para iniciarse y desarrollarse.
- Analizar la viabilidad del emprendimiento.
- Ser el responsable por la administración del Product Backlog
- Garantizar que el producto se entregue.
- Desarrollar y establecer los criterios para aceptar las historias de los usuarios.
- Aprobar o negar los productos entregables. Participar en la Retrospectiva de Sprint y del proyecto.

SCRUM MASTER

El Scrum Master (SM) es un Facilitador del Equipo Scrum. Tiene como misión principal conseguir que el Cliente/P.O., los integrantes del Equipo de desarrollo/DT, así como sus relaciones con la organización y Stakeholders, logren ser un Equipo (Scrum) de alto rendimiento.





Debe garantizar que el equipo incorpore en los diferentes procesos de tomas de decisiones los principios y valores de Agile como la teoría y prácticas de Scrum. El SM actúa como facilitador de reuniones y debe asegurar que el proceso es comprendido y seguido.

Listado de responsabilidades del SM

- Gestionar reuniones de Scrum: se encarga de organizar las reuniones y manejar que se respeten los tiempos, que sean productivas, se piense de manera conjunta, se creen sinergias y se consigan sus objetivos.
- Asegurar que esté la lista priorizada de objetivos/requisitos (la cual representa la visión y expectativas del cliente respecto a los objetivos y entregas del producto o proyecto) y esté preparada antes de la siguiente iteración.
- Resolver conflictos: propicia una buena comunicación basándose en la honestidad y el respeto.
- Promover y asegurar las buenas prácticas de las metodologías ágiles.
- Apoyar al equipo: ayuda a los miembros a solucionar eventuales dudas o problemas que surjan durante el proyecto.
- Enseñar al equipo a autogestionarse.
- Guía al equipo con preguntas para que descubran por sí mismo la posible solución.
- Conocer el grado de motivación de cada uno de los miembros del equipo.
- Quitar impedimentos que tiene el equipo en su camino para conseguir el objetivo en cada iteración.
- Proteger y aislar al equipo de interrupciones externas durante la ejecución de la iteración.

DEVELOPMENT TEAM

El Equipo de Desarrollo (DT) es un grupo de entre 5 y 9 personas encargadas de la producción del proyecto y enfocadas a producir resultados.

Todos los desarrolladores se encuentran en la misma jerarquía, tienen distintas habilidades, son autosuficientes, responsables de sus decisiones y se enfocan en un área distinta, sin embargo, los roles no son fijos, ya que los miembros pueden cambiar de rol de un Sprint a otro.

Cuanto más chico es el grupo más tareas debería realizar cada uno, mientras que en grupos demasiado grandes se dificulta la comunicación.

Listado de responsabilidades del DT

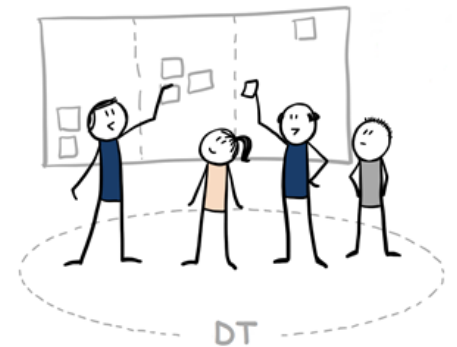
Desarrollar el producto y crear un incremento terminado a partir de los ítems del Product Backlog seleccionados.

Participar de todos los eventos:

Sprint Planning: Estimar y seleccionar ítems del Product Backlog para el Sprint.

Sprint Review: Hacer un demo del incremento, intercambiar opiniones e información con el Product Owner, contestar preguntas que puedan surgir y discutir qué cosas fueron bien y cuáles no y que se hizo para corregirlas.

Sprint Retrospective: Intercambiar opiniones con los demás miembros del Development Team.



ROLES EXTERNOS

STAKEHOLDER

Hace referencia a aquellas personas que tienen un interés, directo o indirecto, por el producto. Los Stakeholders no tienen un rol formal y no se involucran frecuentemente en el Scrum, sin embargo deben ser tomados en cuenta.

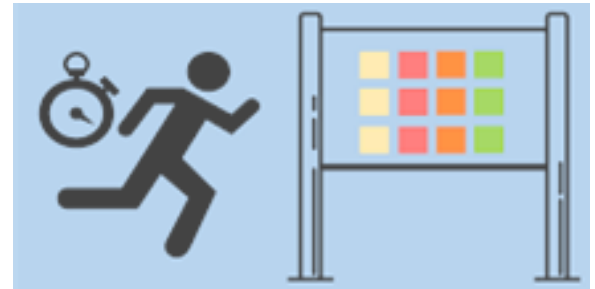
El Stakeholder participa al inicio del ciclo ya que es el que se comunica con el PO.



ARTEFACTOS

SPRINT BACKLOG

El Sprint Backlog es una sublista de objetivos que surge del Product Backlog donde se incluye los objetivos para cada Sprint, aportando un soporte visual para ver el estado de las tareas pendientes, en proceso y terminadas.



INCREMENTO

Es la suma de todos los ítems del Product Backlog completados durante un Sprint y el aporte de valor de los incrementos de los Sprints anteriores.

Al finalizar cada Sprint, habrá un nuevo incremento de funcionalidad que potencialmente se pueda poner en producción y que se ajuste a la definición de “Terminado”.



PRODUCT BACKLOG

El Product Backlog es un listado de todas las tareas que se pretenden hacer durante el desarrollo del proyecto.

Todas las tareas deben listarse en el Product Backlog, para que estén visibles ante todo el equipo y se pueda tener una visión panorámica de todo lo que se espera realizar. El PO es el encargado de realizar la lista y ordenarla.



HERRAMIENTAS

TABLERO SCRUM

Un Tablero Scrum es un método visual para organizar la gestión de un proyecto. Cuenta con columnas que indican el estado de cada tarea y sirve para:

- Dividir el proyecto en fases bien definidas.
- Conocer la efectividad del equipo.
- Monitorear el desarrollo de las
- tareas, de acuerdo con los objetos de cada iteración.
- Aplicar cambios para obtener los resultados esperados.
- Dinamizar la actividad.

Historias	Por Hacer	En Proceso	Terminado
Historia A		Tarea	Tarea
Historia B	Tarea	Tarea	Tarea
Historia C		Tarea	Tarea

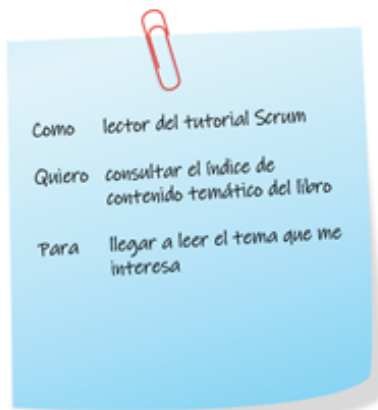


Partes de un tablero:

- **Historias:** Aquí se proporciona información sobre cómo debe ser el comportamiento del requerimiento que se está trabajando, y su prioridad según el PO. Su función es proporcionar información directa del cliente en caso de existir algún cambio.
- **Por Hacer:** cuando una tarea es asignada a un miembro del equipo pasa a esta columna, así como también el SM y
- PO son notificados respecto a que está haciendo cada miembro y cuánto tiempo lleva trabajando en dicha tarea.
- **En Proceso:** cuando una tarea se está realizando, se mueve a esta columna.
- **Terminado:** cuando todas las tareas están completas, y se hicieron las revisiones correspondientes para ser enviadas.

Existen otras Apps para organizar las tareas en el equipo: **Asana, Trello, Basecamp, Kanbanflow, Tribescale**, entre otras.

HISTORIAS DE USUARIO




La Historia de Usuario es algo funcional que aporta valor al usuario y es un elemento básico para aplicar metodologías Ágiles y especialmente para poder aplicar Scrum.

Es una representación de los requisitos de una funcionalidad que el cliente solicita para cumplir con los objetivos del negocio. Es una breve descripción (escrita generalmente en un párrafo) de una característica expresada desde la perspectiva de la persona (usuario o cliente) que desea la nueva capacidad del producto.

Es una técnica muy utilizada en metodologías ágiles para la especificación de los requisitos. Permite una forma rápida de administrar los requisitos de usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos, asegurando responder rápidamente a los requisitos cambiantes.

¿Quién define la Historia de Usuario?

Sabiendo que en Agile se trabaja de forma colaborativa con todo el Equipo Scrum, cualquiera podría o debería escribir historias de usuario. Pero es el PO el responsable de definir las Historias de Usuario, teniendo en cuenta, que es el responsable último de mantener ordenado y priorizado el Producto Backlog.





Formato de una Historia de Usuario

Para redactar la Historia de Usuario se debe tener en cuenta describir el Rol (de forma explícita), la funcionalidad (indicando el beneficio para el área del negocio) y el resultado esperado (en una frase corta). Debe estar acompañada (al reverso) de los criterios de aceptación, hasta un máximo de 4 por historia, redactado también en una frase que indique el contexto, el evento y el comportamiento esperado ante ese evento.

Responde a las siguientes preguntas:

- ¿Quién se beneficia?
- ¿Qué se quiere?
- ¿Cuál es el beneficio?

El resultado de la frase redactada tiene la siguiente sintaxis:
Como [rol] quiero [algo] para poder [beneficio].

Una Historia de Usuario sigue el siguiente formato:

Como <quién> Quiero <qué> Para <objetivo>.

Definir quién

Utilizará la funcionalidad a desarrollar.

Es muy valioso tener alguna idea de las características que tienen las personas que utilizarán el producto, y detallar su necesidad y problemas actuales para dar lugar al entendimiento sobre sus expectativas reales.

Especificar qué producto quiere el usuario.

Las Historias de Usuario deben describir qué se espera como salida de la implementación, y cómo se ve beneficiado el usuario final. Se expresa en lenguaje natural y sencillo, para poder conversar directamente con ellos sobre el tema.

Hay que tener en consideración que los usuarios finales pueden desconocer el lenguaje técnico, por lo que deben evitarse las palabras difíciles. Al usuario, en general, no le interesa qué tecnología se utiliza para la resolución del problema; espera que sea simple, intuitiva y fácil de usar.

Para qué utilizará el producto.

Es importante definir el contexto donde surge la Historia de Usuario que se está generando. Esto ayudará a comprender el valor agregado que se dará, establecerá el objetivo de creación y desarrollo y brindará la posibilidad de poder contar con otras alternativas de solución.

Criterio de aceptación.

Es aquí donde se especifica qué salidas se obtendrán cuando se finalice el proceso de ejecución de la funcionalidad, y servirá para verificar que está terminada la funcionalidad. Estará relacionada con las pruebas que se realizarán para verificar el cumplimiento de la expectativa de diseño, usabilidad, rendimiento, y la satisfacción del usuario.



MALAS PRÁCTICAS APLICANDO SCRUM

SPRINT RETROSPECTIVE


- No realizar las reuniones “retrospectivas” de forma periódica por el equipo, que llevan a mejorar el marco de trabajo.
- Buscar culpables: esto genera que no sea un espacio seguro, afecta directamente a la transparencia que vamos a tener en futuras retrospectivas. Si esto sucede nadie va a querer transparentar los problemas.
- Realizar catarsis sin acciones: es un problema común. Si no definimos accionables nos quedamos solo en el problema y lo que hicimos mal, en estos casos la retrospectiva no tiene sentido.
- Realizar accionables no medibles: se definen accionables no medibles como por ejemplo: “mejorar la comunicación con el área de ventas” eso podríamos cambiarlo por “coordinemos al menos 2 reuniones con las personas clave del área de ventas en el próximo sprint”.
- Tener demasiados accionables: está demostrado que querer tomar muchos accionables (más de 3 o 4) provoca que se haga nada de lo definido o se haga a medias.
- No realizar seguimiento de los accionables: muchos equipos definen accionables pero luego no se siguen o se revisan recién en la siguiente retrospectiva. Una buena práctica es sumar esos accionables al Product Backlog o tenerlos en el lugar visible para revisarlos en la Daily Scrum.



DAILY SCRUM

- Pasar más del tiempo permitido en la reunión (no más de 15 minutos).
- Realizar la reunión en diferentes horarios y/o lugares.
- Tomar decisiones gerenciales u operativas.
- Discutir y resolver problemas técnicos y/o operativos del Sprint.
- Considerar una reunión de control sobre el DT.
- Prestar más atención al objetivo del ticket que a al objetivo del Sprint, esto puede disminuir considerablemente el trabajo colaborativo.



- 
- Tener muchos tickets en proceso, puede atentar a que no se dé o den por terminados al finalizar el Sprint.
 - Tener temas sin conexión alguna. Esto puede ser un síntoma de que se está trabajando con funcionalidades muy distintas y no como un equipo de trabajo colaborativo, lo que puede conllevar a que no le encuentren sentido al evento y pierdan el interés de hacerlo.
 - Tener mucho ruido en la conversación, esto puede ocasionar que se pierda el hilo conductor entre los miembros y se pierda tiempo.
 - Atentar a que no se incentive la auto organización del DT y gestión del evento cuando el SM asigna los tickets al DT. Esto puede poner en
 - peligro la eficacia en el proceso ya que tampoco se promueve la responsabilidad compartida para cumplir con el objetivo del Sprint.
 - Al finalizar un ticket, tomar otro sin respetar las prioridades del Sprint Backlog. No considerar si puede colaborar con otro miembro del DT, o no tener en cuenta aquellos que agregan mayor valor al cumplimiento del objetivo del Sprint.
 - Ser un DT dependiente del SM y que no exista una sincronización en el equipo. Esto puede generar que peligre la realización del Daily Scrum por falta del SM.

SPRINT REVIEW

- Improvisar en la entrega del incremento, producto no funcional, historias de usuario inconclusas.
 - No invitar al PO o a los Stakeholders.
 - Ser un PO egocéntrico o que no escuche el feedback de los Stakeholders.
 - El incremento sea presentado por el SM.
 - Ser un DT muy técnico a la hora de hablar sobre el Sprint a los Stakeholder
 - No tener participación, siempre son los mismos DT quienes participan.
 - Realizar una presentación del incremento por más de 4 horas.
 - Discutir sobre el proyecto
 - Falta de compromiso o pasividad por parte de los Stakeholders
 - Tener discontinuidad y cambios frecuentes de Stakeholders
- 