

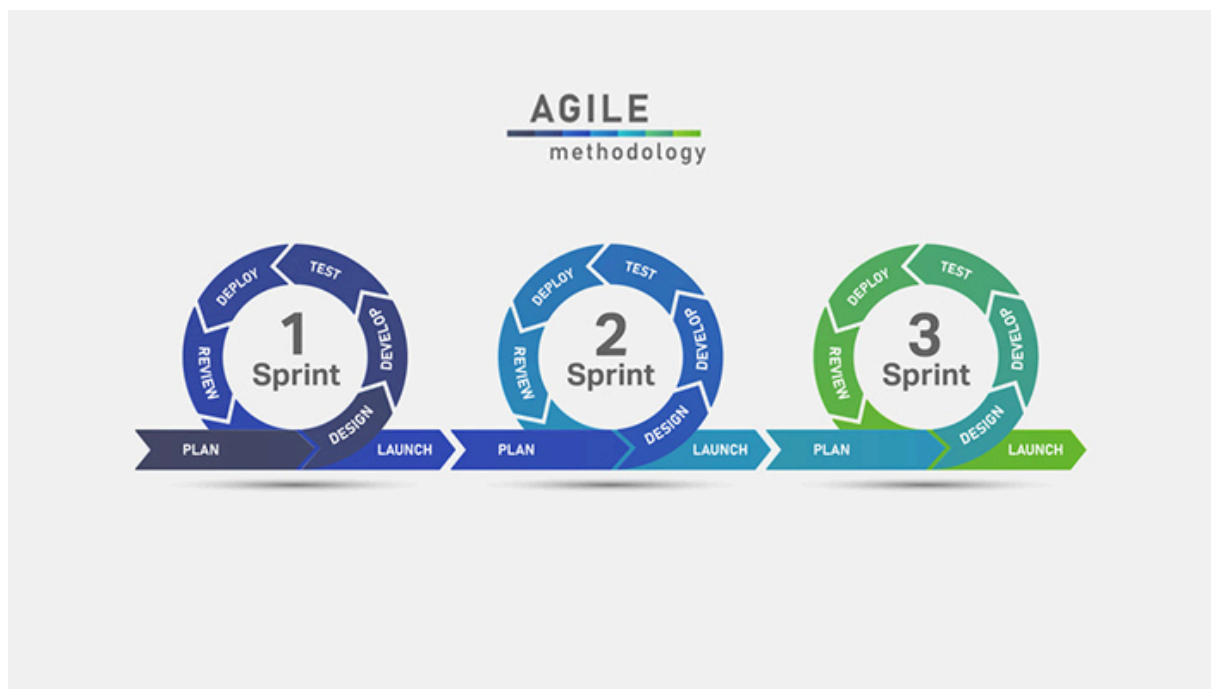
Metodologías Ágiles I

Introducción

Las **metodologías ágiles** producen software en entregas por medio de iteraciones, ciclos o sprints (tres maneras de referirnos a lo mismo). Se centran en la entrega incremental de productos, la retroalimentación constante de los usuarios y la capacidad de responder a cambios en los requisitos de manera rápida y efectiva

Cada ciclo tiene un principio y un fin marcado y las actividades dentro del ciclo son típicamente las mismas.

En cada una se avanza en la construcción del sistema que está en producción.



Es importante destacar que el desarrollo ágil es un campo en constante evolución. Veamos algunas definiciones de fuentes líderes en el ámbito de las metodologías ágiles que ofrecen perspectivas diferentes:

"Metodologías ágiles son enfoques para el desarrollo de software que promueven entregas frecuentes, colaboración constante con el cliente, equipos autoorganizados y la capacidad de responder rápidamente a cambios".. Fuente: Agile Alliance, una organización sin fines de lucro que promueve los valores y principios ágiles.

"Scrum es un marco ágil para el desarrollo, entrega y mantenimiento de productos complejos, con un énfasis en la entrega temprana de funcionalidades valiosas". Fuente: Scrum.org, una entidad que ofrece certificaciones y recursos relacionados con la metodología Scrum.

"Kanban es un método ágil y evolutivo para gestionar el trabajo de conocimiento, con énfasis en la entrega continua, la mejora gradual y el respeto hacia las personas y los procesos existentes". Fuente: Kanban University, una organización que proporciona educación y certificación en el uso de Kanban..

Origen

El origen de las metodologías ágiles se remonta a la década de 1990, cuando un grupo de desarrolladores de software comenzó a buscar formas más efectivas de abordar el desarrollo de software en un entorno caracterizado por cambios constantes en los requisitos y la necesidad de una entrega rápida. Estos profesionales se reunieron y elaboraron el "Manifiesto Ágil" en el año 2001, estableciendo los principios fundamentales que guiaron el movimiento ágil. El Manifiesto Ágil fue redactado por un grupo de 17 expertos en desarrollo de software que buscaban alternativas a los enfoques de gestión de proyectos tradicionales.

Valores

Los cuatro valores del Manifiesto Ágil son:

- **Individuos e interacciones sobre procesos y herramientas:** Se enfatiza la importancia de las personas y la comunicación efectiva entre los miembros del equipo.
- **Software funcionando sobre documentación exhaustiva:** Se valora más la creación de software en funcionamiento que la generación de una gran cantidad de documentación.

- **Colaboración con el cliente sobre negociación de contratos:** Se promueve la colaboración estrecha con el cliente y la adaptación a sus necesidades cambiantes en lugar de enfocarse en contratos rígidos.
- **Responder al cambio sobre seguir un plan:** Se reconoce que los cambios en los requisitos son inevitables, y se prioriza la capacidad de adaptarse rápidamente a esos cambios.

Ventajas

Las metodologías ágiles ofrecen numerosas ventajas en el desarrollo de software, lo que ha contribuido a su creciente popularidad en la industria. Algunas de las principales ventajas de utilizar metodologías ágiles:

- **Adaptabilidad a los cambios:** Las metodologías ágiles están diseñadas para lidiar con la incertidumbre y los cambios en los requisitos del proyecto. Los equipos pueden ajustar sus planes y prioridades de manera ágil a medida que surgen nuevos conocimientos o necesidades del cliente.
- **Entrega continua de valor:** Las metodologías ágiles promueven la entrega temprana y frecuente de partes funcionales del producto. Esto permite al cliente y a los usuarios beneficiarse rápidamente del trabajo realizado y proporciona oportunidades constantes para la retroalimentación.
- **Colaboración efectiva:** Las metodologías ágiles fomentan la colaboración cercana entre los miembros del equipo y con el cliente. La comunicación constante y la interacción cara a cara ayudan a prevenir malentendidos y a alinear mejor los objetivos del proyecto.
- **Mayor satisfacción del cliente:** Al involucrar al cliente a lo largo del proyecto y adaptarse a sus necesidades en curso, las metodologías ágiles tienden a resultar en una mayor satisfacción del cliente. El cliente obtiene lo que realmente desea, y no lo que se definió al principio del proyecto.
- **Mejora de la calidad del software:** Las metodologías ágiles enfatizan prácticas como las pruebas unitarias, la revisión continua del código y la retroalimentación del cliente. Esto conduce a una mayor calidad del software y una reducción de los errores.
- **Mayor productividad del equipo:** La autoorganización y la toma de decisiones por parte del equipo permiten a los miembros del equipo asumir la responsabilidad de su trabajo y ser más productivos. La

limitación del trabajo en curso (WIP) en metodologías como Kanban también puede aumentar la eficiencia.

- **Transparencia y visibilidad:** Las metodologías ágiles suelen utilizar tableros y herramientas visuales para representar el progreso del trabajo. Esto proporciona una visibilidad clara sobre el estado del proyecto y facilita la identificación de problemas.
- **Reducción de riesgos:** La entrega temprana y frecuente permite a los equipos abordar los riesgos y los problemas a medida que surgen, en lugar de esperar hasta el final del proyecto. Esto puede reducir el impacto de los problemas imprevistos.
- **Mayor compromiso del equipo:** Las metodologías ágiles fomentan un entorno en el que los miembros del equipo tienen voz en las decisiones y se sienten más comprometidos con el éxito del proyecto.
- **Eficiencia en la planificación y el uso de recursos:** La adaptabilidad y la focalización en las prioridades permiten una planificación más eficiente y el uso de recursos de manera óptima.

El Manifiesto Ágil dio lugar a diversas metodologías específicas, cada una con sus propias características y enfoques específicos. Es común que los equipos adapten y combinen elementos de diferentes enfoques según sus necesidades específicas. La elección de la metodología ágil adecuada depende del contexto del proyecto y de las preferencias del equipo. Algunas de las metodologías ágiles más conocidas incluyen: Scrum, Kanban, Programación Extrema, Lean, Design Sprint, entre otras.

En este apartado, nos centraremos en 2 de las metodologías más utilizadas en el rubro IT: Scrum y Kanban.

SCRUM

Scrum es uno de los marcos ágiles más populares. Se basa en ciclos de desarrollo cortos llamados "sprints", que generalmente duran de 2 a 4 semanas. Scrum enfatiza la colaboración, la transparencia y la adaptabilidad.

Scrum es una filosofía y un marco de trabajo que ayuda a las organizaciones a generar valor a través de soluciones adaptativas a problemas complejos. El foco está puesto en el **trabajo en equipo**.

Esta metodología de trabajo emplea un enfoque iterativo e Incremental para optimizar la previsibilidad y controlar el riesgo.

Las reglas de Scrum ayudan a guiar las relaciones e interacciones. No buscan proveer instrucciones detalladas ya que el scrum se basa en la inteligencia colectiva de las personas que lo utilizan.

Valores:

Los miembros del equipo se respetan, tienen el coraje de hacer lo correcto ante adversidades, son francos y asumen el compromiso de lograr sus objetivos apoyándose mutuamente con foco en su principal tarea: **el mejor progreso posible sobre el trabajo del sprint**



Equipo Scrum:

Cada equipo scrum suele ser pequeño sin jerarquía. Es una pequeña unidad multifuncional y autogestionada de profesionales.

Los equipos tienen la capacidad de decidir qué hacer, cómo y cuándo y son responsables del ciclo de vida de desarrollo de parte del producto.

Generalmente cada equipo scrum tiene menos de 10 personas ya que el equipo debe mantenerse pequeño para ser ágil y a la vez contar con la cantidad suficiente de miembros para cumplir con tareas significativas en cada sprint.

Roles:

- **Scrum master:** es un facilitador y líder del equipo, y es quien se ocupa de inculcar la metodología. Su principal responsabilidad es asegurar que el equipo siga las prácticas y valores de Scrum, eliminando obstáculos y facilitando un entorno propicio para la entrega de software de alta calidad.
- **Equipo de Desarrollo (Development Team):** Encargados de entregar el incremento del producto al final de cada sprint. Son autoorganizados, multifuncionales y colaboran estrechamente para lograr los objetivos del sprint.
- **Product Owner (Dueño del Producto):** Define y prioriza el backlog del producto. Responsable de maximizar el valor del producto y del trabajo del equipo de desarrollo..

Eventos:

Scrum define una serie de eventos, también conocidos como ceremonias, que estructuran y facilitan la planificación, la colaboración y la entrega en el marco ágil. Estos eventos están diseñados para ser breves, enfocados y regulares. Los eventos principales de Scrum son:

Planning

Este evento se lleva a cabo al inicio del sprint. Es aquí donde se define el trabajo que se realizará en el sprint. Lo que se define en la planning se define de manera grupal a nivel equipo. Es importante como miembro del equipo tener una idea de lo que hay en el backlog para optimizar el rendimiento de esta reunión estando al tanto de las posibles tareas que están en espera de desarrollo.

Daily

Es una reunión diaria que generalmente dura un máximo de 15 minutos (y suele llevarse a cabo siempre en el mismo horario). En ella, cada miembro del equipo comenta su progreso y si tiene algún roadblock (impedimento) para avanzar.

Review

En esta ceremonia se presenta el trabajo del sprint a los interesados (stakeholders). Se revisa lo realizado y se comenta sobre próximos pasos.

Retrospective

Esta es una reunión interna del equipo en la que se analiza la necesidad de ajuste para mejorar la calidad y efectividad del trabajo del sprint. Se inspecciona el proceso de trabajo en relación (tareas, personas, comunicación, herramientas) y se identifican limitantes. Se analizan posibles correcciones a problemas no resueltos.



💡 ¿Quieres saber más sobre scrum?

En <https://scrumguides.org/download.html> puedes encontrar la guía oficial de scrum (actualizada en 2020) escrita por sus creadores (Jeff Sutherland y Ken Schwaber) y traducciones a diferentes idiomas hechas por miembros de la comunidad.

KANBAN

El método Kanban forma parte de la conocida estrategia japonesa de trabajo llamada Kaizen. parte de la premisa de que las cosas, siempre se pueden hacer mejor. Es decir, su propósito radica en que de no modificarse los hábitos se perderá competitividad en el mercado.

Origen:

Kanban en Japonés significa letrero o señal. Nació a finales de los años '40 como sistema en ámbitos fabriles con el objetivo de controlar el proceso de producción usando un modelo JIT (sistema de arrastre basado en la demanda). Cuando un producto es vendido, la señal se mueve hasta el principio de la línea de fabricación para que se produzca uno nuevo. El propósito es minimizar los desperdicios.

Kanban es un sistema de gestión de proceso visual que indica qué producir, cuándo y en qué cantidad.

Enfoque:

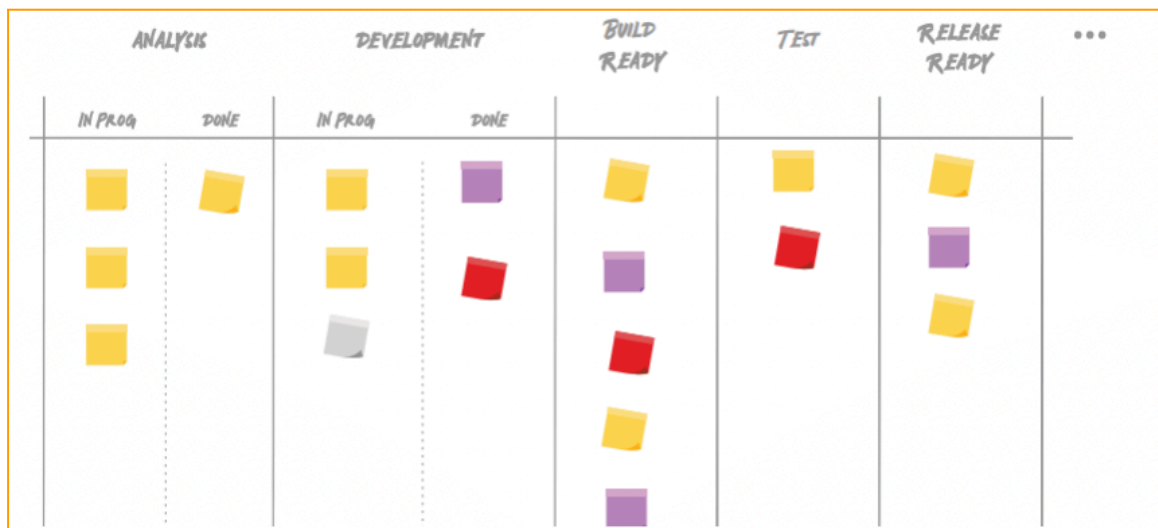
El método Kanban, formulado por David J. Anderson es un enfoque para el cambio evolutivo incremental para las organizaciones de desarrollo. O dicho de otra manera: Una aproximación a la mejora de procesos organizacionales de manera incremental y evolutiva.

Limita el trabajo en curso (aka WIP) o sistema de extracción limitada de trabajo en progreso, para exponer los problemas de funcionamiento del sistema o proceso estimulando la colaboración para la mejora continua del sistema.

El proceso de cambio de sistemas hacia la mejora debe ser gradual y evolutivo.

El proceso, desde la definición de la tarea hasta su entrega se muestra para ser vista por los participantes. ¿De qué manera? Utilizando tableros Kanban o Kanban boards, como el que se muestra en la imagen.

Las tarjetas no funcionan como señales para realizar más trabajo, sino que representan los elementos de trabajo en sí mismo.



Algunas herramientas que permiten implementar el método Kanban son: Kanban Tool, JIRA, Trello, Note Board, Targetprocess.

Se debe tener en cuenta 3 puntos principales :

- **Comenzar con la actividad actual:** El método Kanban no exige que el proceso cambie drásticamente, sino que se basa en la evolución del proceso. Es decir poco a poco. Para eso, se continúa trabajando del modo en el que se hacía respetando procesos, roles, responsabilidades y cargos.
- **Acuerdo del cambio continuo, incremental y evolutivo:** Debe existir cierto acuerdo en la organización (o en el equipo) justificando trabajar en pequeñas mejoras incrementales.
- **Liderazgo en cada nivel:** Es importante que todos fomenten la mejora continua para alcanzar el rendimiento óptimo. En palabras de David J. Anderson: "La información sobre dónde las mejoras son posibles puede aparecer en cualquier nivel dentro de una organización y, con mayor frecuencia, a nivel del contribuyente individual... Debe existir una cultura de

seguridad, para actuar sin temor a represalias, siempre que los cambios se puedan defender con base en una explicación lógica utilizando modelos y datos.”

La organización debe fomentar y estar abierta al feedback y propuestas de todos los miembros de la organización.

Principios y Prácticas Fundamentales :

1. Visualizar el workflow (flujo de trabajo)

Visualizar el flujo de trabajo y hacerlo visible es la base para comprender cómo avanza el trabajo. Sin comprenderlo es difícil realizar los cambios adecuados. Una manera de hacerlo visible es representar con tarjetas o columnas cada actividad que genera nueva información >>> **análisis, diseño, testing.**

Al poner de manifiesto y representar la secuencia de actividades que llevamos a cabo para completar nuestro trabajo estamos dando visibilidad a nuestro flujo de trabajo. Las columnas representan los diferentes estados o pasos en el flujo de trabajo.

2. Limitar work in progress

Limitar el trabajo en curso implica la implementación de un sistema de “extracción o ingreso”.

La clave es que el trabajo en progreso en cada estado del flujo debe ser limitado. Solamente ingresamos nueva actividad cuando hay capacidad para hacerlo (por ejemplo, si se liberó lugar en el flujo de trabajo al completarse otra tarea).

3. Manejar y monitorear el flujo de trabajo

El flujo de actividades a través de cada estado del flujo debería ser monitoreado. Esto se realiza a partir de reportes para medir la velocidad con la que avanzan las tareas a través de los distintos estados del flujo de trabajo.

Se identifican así los cuellos de botella. Reconocer las dificultades y visualizar los problemas, para facilitar la mejora continua.

El método Kanban recomienda conseguir un flujo rápido y fluido. Un flujo rápido y fluido significa que el sistema está creando valor rápidamente, lo que minimiza el riesgo y evita el costo (de oportunidad) de la demora, y también agrega predictibilidad al trabajo y los tiempos de realización.

4. Hacer explícitos los procesos y políticas.

Es clave que los procesos y políticas (o la total ausencia de los mismos) estén claros y al conocimiento de todos. Para modificar o crear un proceso hay que reflexionar si existe ya alguno definido para esta tarea. De ser así ¿cual sería?

Esa información es el punto de partida para la revisión y mejora (o creación) de los procesos necesarios.

5. Usar circuitos de retroalimentación (Feedback Continuo)

El propósito de la retroalimentación es ajustar los procesos y políticas y mejorarlos. Para eso, se comparan los resultados esperados con los reales y luego se decide qué modificar.

6. Mejora Colaborativa y Evolutiva

Cuando los equipos trabajan en conjunto es probable que puedan construir una visión compartida de un problema y proponer acciones de mejora acordadas por consenso. La mejora continua es un esfuerzo colaborativo. Los equipos y las organizaciones deben evolucionar gradualmente, ajustando sus prácticas según las necesidades y aprendiendo de la retroalimentación.



💡 ¿Quieres saber más sobre KANBAN?

Te dejamos varios recursos que seguro te resultará de utilidad:

- <https://dja.com/kanban-board-examples/>
- <https://dja.com/revisiting-emerging-roles-in-kanban/>
- <https://dja.com/kanban-evergreen-should-we-include-waiting-or-blocked-items-in-wip-limits/>
- <https://dja.com/revisando-los-principios-y-practicas-generales-del-metodo-kanban/>
- <https://dja.com/applying-kanban-to-sales/>

- <https://dja.com/individual-kanban-faq/>
- <https://dja.com/applicability-of-the-kanban-method-to-the-legal-sector/>

Información de la industria:



SCRUM y **KANBAN** no son excluyentes entre sí.

Los equipos de trabajo pueden usar scrum para su operativa y Kanban para monitorear y verificar que los procesos fluyan con la agilidad necesaria o para hacer ajustes globales, de ser necesario.

Extra Credits: 12 Principios del Manifiesto Ágil

Los **12 Principios del Manifiesto Ágil** complementan los valores establecidos en el manifiesto y proporcionan directrices adicionales para el desarrollo de software ágil.

A continuación, exploraremos los 12 principios del Manifiesto Ágil.

- **Satisfacer al cliente a través de la entrega temprana y continua de software valioso.** Este principio enfatiza la importancia de la satisfacción del cliente como objetivo central. La entrega temprana y frecuente de software permite al cliente obtener valor rápidamente.
- **Aceptar cambios en los requisitos, incluso en etapas tardías del desarrollo.** Reconoce que los requisitos pueden cambiar y que los procesos ágiles están diseñados para adaptarse a estos cambios, lo que brinda ventaja al cliente.
- **Entregar software funcional frecuentemente, con preferencia a intervalos cortos.** Fomenta la entrega continua de partes funcionales del software en plazos cortos, lo que permite obtener retroalimentación constante y acelerar la entrega de valor.
- **Colaborar de manera constante con los clientes y los interesados a lo largo del proyecto.** Destaca la importancia de la colaboración continua entre el equipo de desarrollo y los interesados (clientes, usuarios, etc.) para garantizar una comprensión común de los objetivos y requisitos.

- **Construir proyectos en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en que harán el trabajo.** Enfatiza la importancia de empoderar y motivar a los miembros del equipo, lo que conduce a una mayor responsabilidad y productividad.
- **El método más eficiente y efectivo de comunicar información dentro de un equipo de desarrollo es la conversación cara a cara.** La comunicación directa y cara a cara se considera el método más efectivo para compartir información y resolver problemas.
- **El software funcional es la principal medida de progreso.** Se valora más la creación de software en funcionamiento que la generación de documentación extensa. El progreso se mide por la funcionalidad entregada.
- **Los procesos ágiles promueven un entorno sostenible. Los patrocinadores, desarrolladores y usuarios deben ser capaces de mantener un ritmo constante de forma indefinida.** Busca evitar la fatiga y el agotamiento del equipo al mantener un ritmo de trabajo sostenible a lo largo del proyecto.
- **La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.** La calidad del código y la arquitectura son fundamentales para la agilidad, ya que facilitan la adaptación a cambios futuros.
- **La simplicidad, o el arte de maximizar la cantidad de trabajo no hecho, es esencial.** Se prioriza la simplicidad y se busca evitar la **sobre-ingeniería**, enfocándose en las tareas realmente necesarias.
- **Las mejores arquitecturas, requisitos y diseños emergen de equipos autoorganizados.** Se enfatiza la autoorganización del equipo y la toma de decisiones colaborativa como fuente de soluciones efectivas.
- **A intervalos regulares, el equipo reflexiona sobre cómo ser más efectivo, y ajusta su comportamiento en consecuencia.** La mejora continua es fundamental; el equipo debe reflexionar y ajustar sus prácticas y procesos de manera constante.

