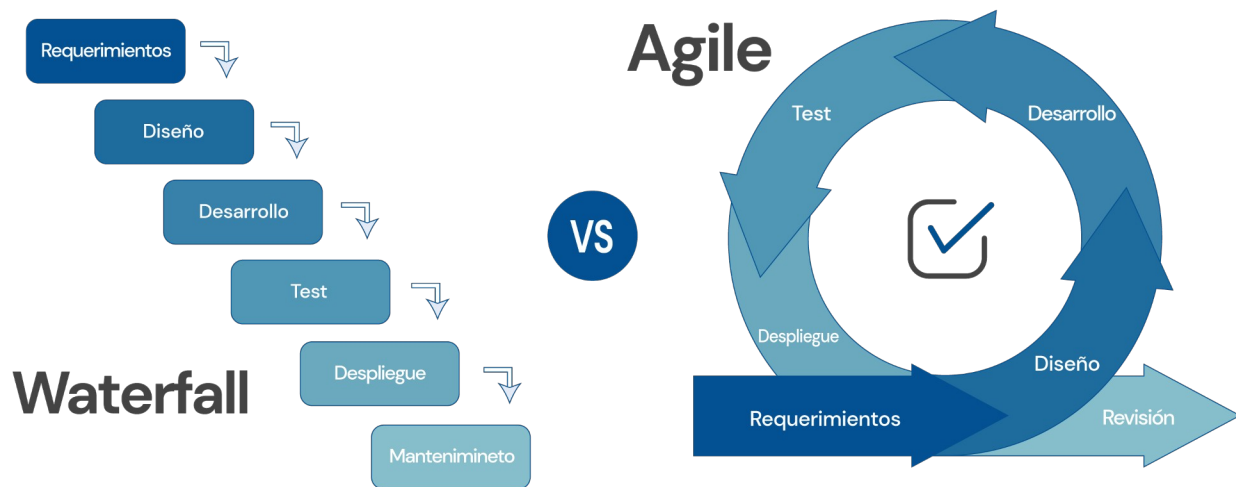


Metodología Agile



En el mundo actual de las Tecnologías de la Información (TI), los proyectos cambian rápidamente. Los requisitos evolucionan, los usuarios dan retroalimentación constante y la competencia exige entregas más rápidas.

Por eso, las metodologías tradicionales como Waterfall (en cascada) ya no son suficientes para muchos tipos de proyectos. Es aquí donde entra en juego la metodología ágil, un enfoque flexible, colaborativo e iterativo que permite a los equipos adaptarse al cambio y entregar valor más rápido.

La agilidad no es solo una metodología; es una forma de pensar.

¿Qué es la Metodología Ágil?

La metodología ágil es un enfoque para gestionar proyectos de TI que se basa en:

- La entrega incremental del producto.
- La colaboración continua con el cliente.
- La capacidad de responder al cambio más que seguir un plan rígido.

Fue formalizada en 2001 con la creación del Manifiesto Ágil para el Desarrollo de Software, por 17 desarrolladores que buscaban una alternativa al modelo lineal y rígido del desarrollo tradicional.

Los 4 Valores del Manifiesto Ágil

Valor	Explicación
Individuos e interacciones > procesos y herramientas	Las personas son más importantes que los sistemas. La comunicación directa genera mejores resultados.
Software funcionando > documentación exhaustiva	Lo importante es entregar algo que funcione, no llenar carpetas con documentos.
Colaboración con el cliente > negociación contractual	El cliente debe estar presente durante todo el proyecto, no solo al inicio y al final.
Respuesta al cambio > seguir un plan	Un buen plan puede cambiar si el mercado o el usuario lo requiere.

Uno de los principales beneficios de aplicar procesos ágiles es la posibilidad de cambiar de estrategia rápidamente, sin afectar el flujo del proyecto.

VALORES ÁGILES



Los 12 Principios Ágiles

Estos principios guían la práctica diaria de los equipos ágiles:

1. **Satisfacer al cliente** mediante la entrega temprana y continua de software valioso.
2. **Aceptar cambios incluso en etapas avanzadas** del desarrollo.
3. **Entregar software funcional frecuentemente** (semanas, no meses).
4. **Trabajar con personas motivadas** y darles el entorno y el apoyo que necesitan.
5. **Fomentar la colaboración constante** entre negocio y desarrollo.
6. **Construir proyectos alrededor de personas motivadas.**
7. **El método más eficiente de comunicación es la conversación cara a cara.**
8. **Software funcionando es la medida principal del progreso.**
9. **Mantener un ritmo de trabajo sostenible** (evitar el agotamiento).
10. **La excelencia técnica mejora la agilidad.** Un equipo puede ser más rápido, flexible y adaptable en el futuro si hoy invierte en hacer un buen trabajo técnico.
11. **La simplicidad es esencial:** No se trata de hacer más cosas, sino de hacer solo lo que realmente aporta valor. Lo mejor que puedes hacer a veces es no hacer nada.
12. **El equipo reflexiona regularmente** sobre cómo mejorar y ajusta su comportamiento.

Con regularidad, el equipo reflexiona y adapta sus formas de trabajo para favorecer la efectividad.

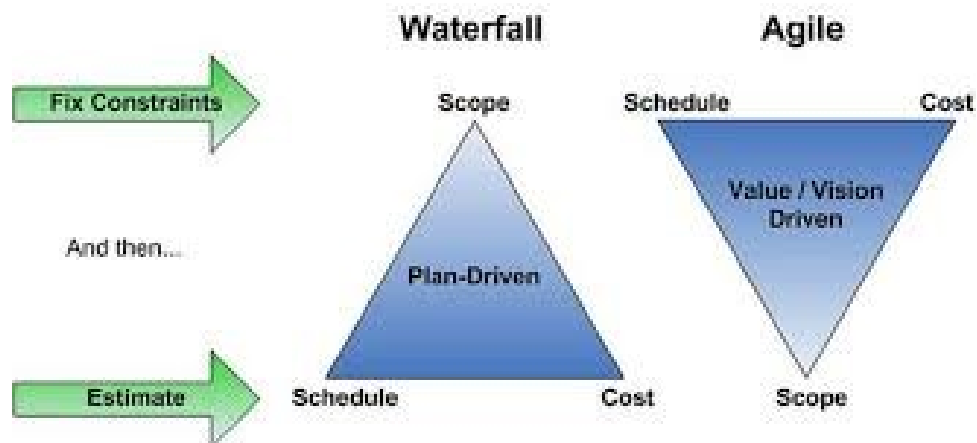
¿Cómo funciona? Ciclo de vida ágil

A diferencia del modelo Waterfall (lineal), el enfoque ágil es iterativo e incremental:

Planificar → Diseñar → Desarrollar → Probar → Evaluar → Desglose → (Repetir)

El trabajo se divide en sprints (períodos cortos de 2 a 4 semanas), al final de cada uno se entrega una versión funcional del producto y se recibe retroalimentación. Después de cada sprint, los equipos reflexionan y observan lo que ha sucedido. Evalúan qué se podría mejorar para el siguiente.

Relación Triple constraints o Iron triangle



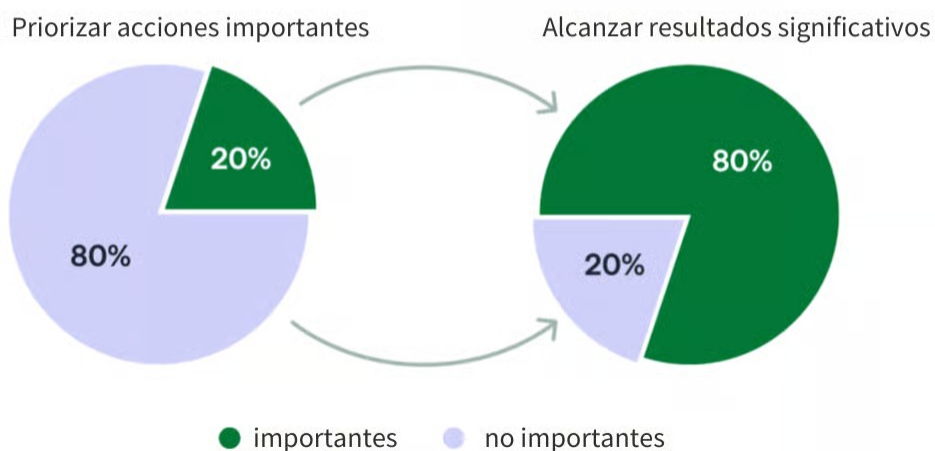
Como puedes observar, en un modelo tradicional en cascada (waterfall), el ALCANCE permanece fijo, y luego el equipo trata de determinar cuál será el costo y el tiempo necesarios para completar el trabajo (lo que hace que el cronograma y el costo sean variables).

Sin embargo, en Agilidad, se sigue un triángulo invertido, en el cual el COSTO y el TIEMPO permanecen constantes (debido a las iteraciones limitadas en el tiempo, como los sprints). Luego, el equipo de proyecto busca estimar cuánto ALCANCE puede asumir en una sola iteración o sprint, lo que convierte al ALCANCE en variable.

Principio de Pareto

El Principio de Pareto

El 80 % de los resultados son generados por el 20 % de los esfuerzos



Ley de Pareto, también conocida como el principio del 80/20, es una herramienta analítica que afirma:

El 80% de los resultados provienen del 20% de las causas.

Este concepto se aplica a una amplia variedad de situaciones, desde negocios hasta productividad personal, y es especialmente útil para priorizar tareas, recursos y esfuerzos en entornos complejos como los proyectos de TI.

Origen del principio

El principio fue formulado por el economista italiano Vilfredo Pareto en 1896, quien observó que: El 20% de los italianos poseían el 80% de la riqueza.

Años después, en 1954, el ingeniero Joseph Juran extendió esta idea al mundo de la calidad y la gestión, proponiendo que:

Una minoría de causas (el 20%) genera la mayoría de los efectos (el 80%).

Esto dio origen al Diagrama de Pareto, una herramienta clave en análisis de calidad y toma de decisiones.

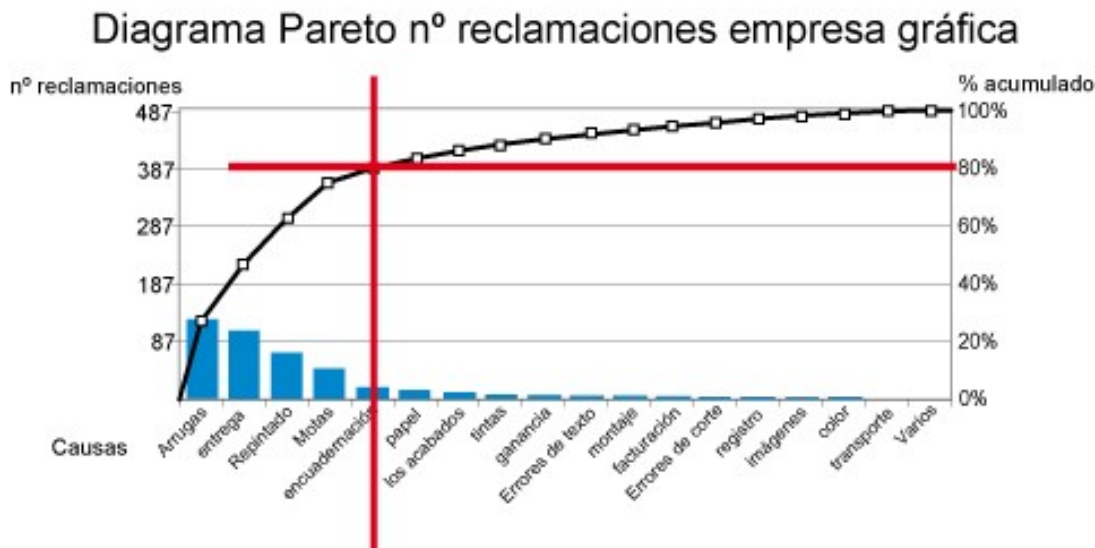


Diagrama de Pareto elaborado a partir de los problemas detectados en los productos y que han sido causa de reclamación. Una vez ordenados se observa que el 27,8% de los problemas han causado el 80% de las reclamaciones o el 22,3% causan en torno al 78% de los problemas. La empresa al dar solución a esos 4 o 5 problemas “vitales” resuelve aproximadamente el 80% de las reclamaciones, lo cual no está nada mal.

¿Cómo funciona la regla del 80/20?

No se trata de una fórmula matemática exacta, sino de una guía mental para identificar lo que realmente importa.

En lugar de...	Usa el principio para...
Trabajar en todo por igual	Identificar el 20% de tareas que generan el 80% del impacto.
Gestionar todos los clientes por igual	Enfocarse en el 20% que genera el 80% de los ingresos.
Corregir todos los errores	Atacar el 20% de problemas que causan el 80% de los retrasos.

Aplicaciones clave de la Ley de Pareto

1. Clientes y ventas :

- El 20% de los clientes generan el 80% de las ventas.
- Acción: Prioriza la atención, servicio y retención de estos clientes clave.

2. Productos y servicios:

- El 20% de los productos representan el 80% de los ingresos.
- Acción: Invierte más en marketing, desarrollo y mejora de esos productos estrella.

3. Marketing digital

- El 20% de las campañas generan el 80% de los leads.
- Acción: Analiza qué canales funcionan y redirige el presupuesto hacia ellos.

4. Ventas y equipos comerciales

- El 20% de los vendedores realizan el 80% de las ventas.
- Acción: Estudia sus técnicas y compártelas con el equipo para mejorar el desempeño colectivo.

5. Gestión del tiempo

- El 20% de tus actividades producen el 80% de tu productividad.
- Acción: Identifica esas tareas clave y protégelas del ruido diario.

La Ley de Pareto en Gestión de Proyectos de TI

En proyectos tecnológicos, este principio puede transformar la forma en que se planifica y ejecuta el trabajo.

Ejemplos prácticos:

Situación	Aplicación del 80/20
Requisitos del cliente	El 20% de los requisitos cubren el 80% de las necesidades reales. Priorízalos.
Errores en software	El 20% de los módulos contienen el 80% de los bugs. Enfoca pruebas

	allí.
Riesgos del proyecto	El 20% de los riesgos tienen el 80% de probabilidad de impactar. Mitígalo primero.
Tareas del cronograma	El 20% de las actividades son críticas (camino crítico). Vigílaslas con atención.

¿Y si no es exactamente 80/20?

¡Perfecto! El número no es rígido. Puede ser:

- 70/30
- 90/10
- 65/35

Lo importante no es la proporción exacta, sino el concepto: una pequeña parte causa la mayor parte del resultado.

No busques el 20% exacto. Busca el 'poco que importa mucho

Beneficios de aplicar la Ley de Pareto

Beneficio	Explicación
Mejor toma de decisiones	Te ayuda a enfocarte en lo que realmente genera valor.
Uso eficiente de recursos	Evita desperdiciar tiempo, dinero o personas en actividades de bajo impacto.
Mayor productividad	Al eliminar el ruido, puedes avanzar más rápido.
Reducción de riesgos	Al atacar las causas principales, previenes muchos problemas secundarios.

Errores comunes al usar el principio de Pareto

Error	Solución
Creer que el 20% es "todo"	El otro 80% también tiene valor, pero con menor prioridad.
Aplicarlo sin datos	Usa registros, encuestas o métricas para identificar el 20%.
Ignorar cambios en el tiempo	Revisa periódicamente: hoy el 20% puede ser diferente mañana.

La Ley de Pareto no es solo una curiosidad estadística, sino una herramienta estratégica poderosa para líderes, gerentes y administradores de proyectos.

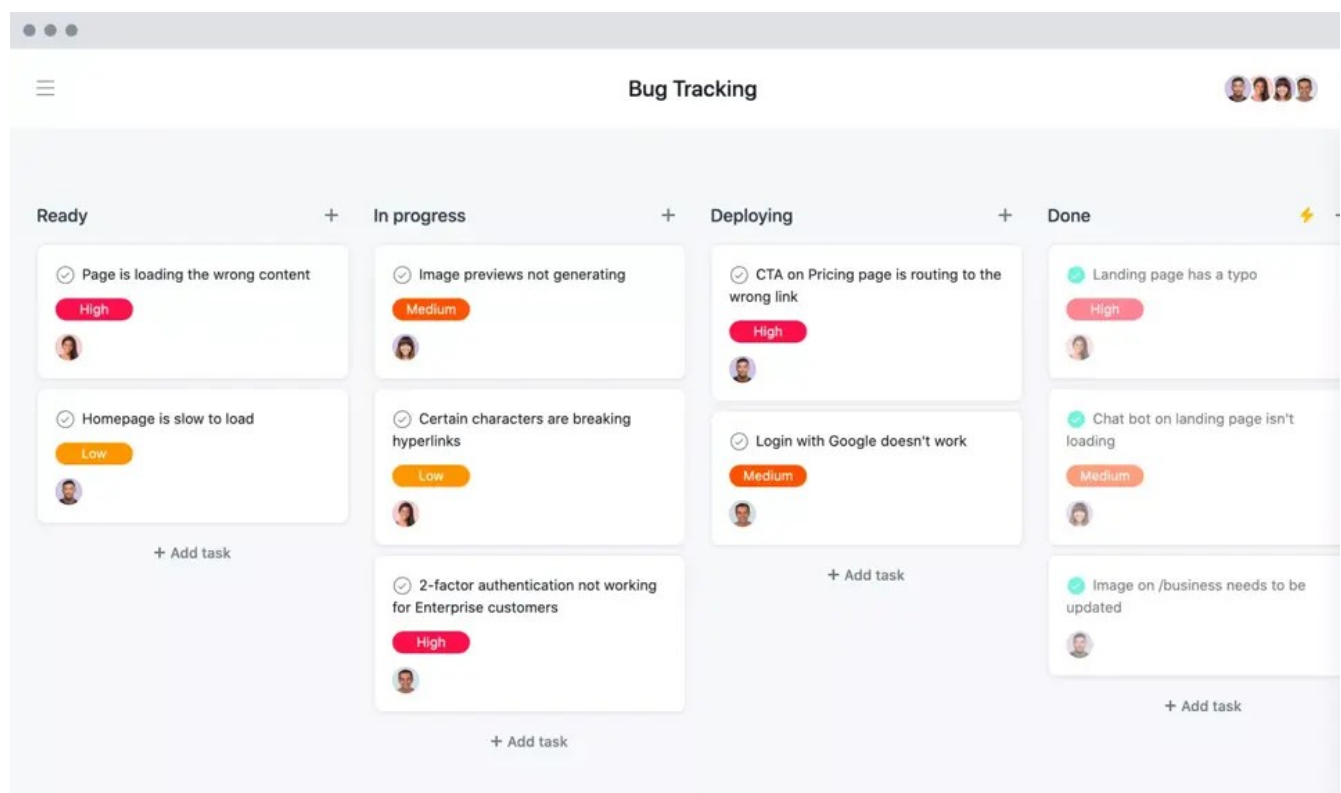
Haz menos, pero mejor. Concéntrate en el 20% que mueve el 80%.

Dominar este principio te permitirá tomar decisiones más inteligentes, optimizar recursos y entregar proyectos con mayor impacto y menor esfuerzo.

Principales marcos de trabajo ágiles

Nombre	Descripción
Scrum	Usa sprints, roles definidos (Product Owner, Scrum Master, equipo) y reuniones diarias. Ideal para equipos pequeños y medianos.
Kanban	Se enfoca en visualizar el flujo de trabajo (tablero: To Do, Doing, Done). No tiene sprints fijos. Perfecto para mantenimiento o soporte entre otros.
Hybrid (Híbrido)	Combina lo mejor de Ágil y Waterfall. Útil cuando hay partes estables (ej: normas legales) y otras cambiantes.

Kanban



Kanban es una subsección de la metodología ágil que se desarrolla dentro de la mentalidad ágil más amplia. Esta filosofía se centra en la planificación adaptativa, la entrega temprana y la mejora continua, aspectos que el marco Kanban puede respaldar.

Implementada por primera vez en los años 80, en el sector automotriz, *Kanban es una herramienta orientada a la gestión de proyectos especialmente complejos o en los que se presenten cuellos de*

Fase 1: planificación del sprint. Estos suelen durar dos semanas, aunque los equipos pueden realizar sprints más rápidos o breves. Durante esta etapa, el Scrum master y el equipo analizan el trabajo pendiente y determinan qué tareas realizar durante el Sprint.

Fase 2: reuniones diarias de actualización. Durante el transcurso del Scrum (también conocido como el “ciclo” de Scrum), los equipos suelen reunirse durante 15 minutos todos los días (daily) para hablar sobre el progreso y asegurarse de que la cantidad de trabajo asignado sea la adecuada.

Fase 3: análisis retrospectivo del sprint. Cuando finaliza el proceso, el Scrum master organiza una reunión retrospectiva del sprint para evaluar qué trabajo se realizó, agregar cualquier tarea incompleta a la lista de trabajo pendiente y prepararse para el siguiente sprint.

Los equipos que implementan Scrum tienen reglas, rituales y responsabilidades claramente establecidos. Además, si combinas las reuniones diarias de Scrum con la planificación y la revisión de sprints (o reuniones “retrospectivas”), tu equipo podrá analizar y mejorar los procesos actuales de forma continua.

Este modelo se basa en una lista de trabajo pendiente y se inicia con una reunión de planificación de sprints, por eso, ofrece una estructura sencilla e integrada para que los líderes del equipo o los encargados de productos administren y respalden el trabajo más importante de su equipo. Durante el proceso, el equipo tiene una cantidad preestablecida y limitada de trabajo y tiempo para cada sprint. Este nivel de priorización integrado se combina con responsabilidades claramente definidas, lo que garantiza que todos sepan de qué son responsables en todo momento.

Un Sprint **Backlog** (trabajo pendiente o pila de tareas) es una lista de tareas y el plan de trabajo que un equipo de desarrollo se compromete a realizar para completar el trabajo definido en un período de tiempo específico, llamado sprint. Es un subconjunto del Product Backlog (la lista general de todos los elementos del producto) y se crea durante la Planificación del Sprint, donde el equipo selecciona los elementos de mayor prioridad del Product Backlog hasta alcanzar su capacidad de trabajo.

<https://asana.com/es/resources/sprint-backlog>

Los primeros pasos en el desarrollo de un proyecto, se recomienda seleccionar el enfoque más apropiado según el tipo de proyecto.

Diferencias clave: Ágil vs. Waterfall

Característica	Ágil	Waterfall
Estructura	Iterativa (se repite) e incremental	Secuencial (una fase tras otra)
Ideal para	Proyectos donde se espera que los requisitos y las prioridades cambien durante su desarrollo	Proyectos con requisitos estables, bien definidos y un alcance predecible desde el principio.
Cambios	Bienvenidos en cualquier momento	Difíciles y costosos después del diseño
Entrega	Versión funcional cada 2-4 semanas	Una sola entrega al final
Cliente	Participa activamente durante todo el proyecto	Suele intervenir solo al inicio y fin
Documentación	Mínima necesaria	Completa antes de comenzar
Riesgos	Se detectan y corrigen rápido	Pueden descubrirse muy tarde

Ejemplo:

- Un banco que implementa un sistema de pagos bajo normas estrictas puede usar Waterfall.
- Una startup que lanza una app móvil con nuevas funciones cada mes necesita Ágil.

Roles clave en un equipo ágil

Aunque el documento original no menciona estos títulos directamente, sus funciones están implícitas en la gestión de proyectos.

Rol	Función principal
Product Owner	Define qué se desarrolla y en qué orden. Representa al cliente.
Scrum Master	Facilita el proceso, elimina obstáculos y asegura que se sigan las reglas de Scrum.
Equipo de Desarrollo	Diseña, programa y prueba el software. Autoorganizado y multifuncional.
Project Manager	En algunos casos, coordina recursos, riesgos y comunicaciones entre áreas

El Project Manager actúa como administrador, facilitador y líder, lo cual encaja perfectamente con estos roles.

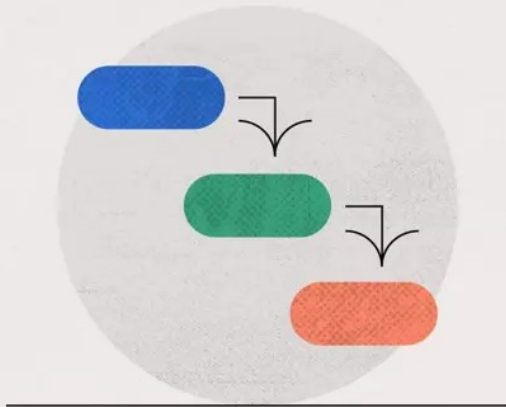
Beneficios de usar Ágil en TI

Beneficio	Explicación
Mayor satisfacción del cliente	Recibe funcionalidades útiles desde el principio.
Adaptabilidad al cambio	Los nuevos requerimientos se integran fácilmente.
Detección temprana de errores	Las pruebas ocurren en cada sprint.
Mejor comunicación	Reuniones diarias y revisiones constantes.
Reducción del riesgo de fracaso total	Si algo falla, se sabe pronto y con poco impacto.
Ritmo de trabajo sostenible	Evita el agotamiento del equipo.
Excelencia técnica continua	El código se mejora en cada iteración.

Retos comunes del enfoque ágil

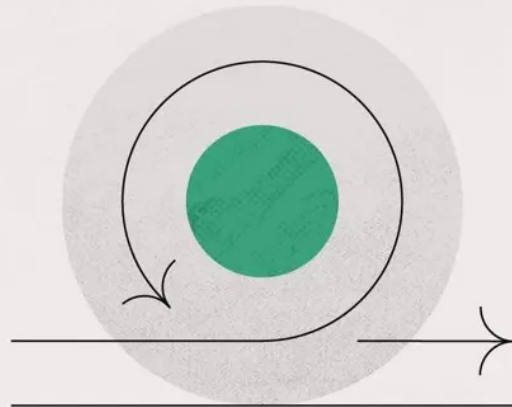
Reto	Solución recomendada
Falta de experiencia del equipo	Capacitación y coaching continuo.
Clientes que no participan	Asignar un Product Owner comprometido.
Resistencia al cambio en la organización	Comunicar beneficios y mostrar resultados rápidos.
Mal entendimiento del "trabajo terminado"	Definir bien los criterios de aceptación.

Waterfall vs. Kanban vs. Agile vs. Scrum



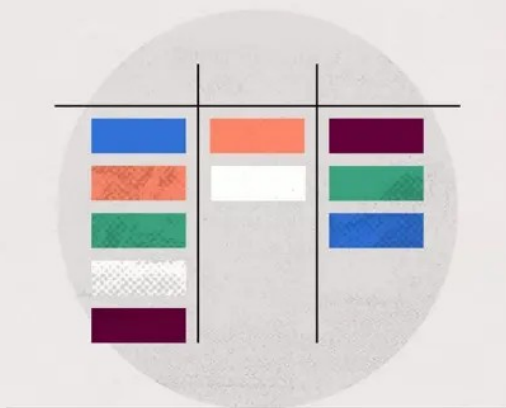
Waterfall

El modelo de cascada o Waterfall divide cada proyecto en diferentes fases y **estas se completan en orden secuencial**.



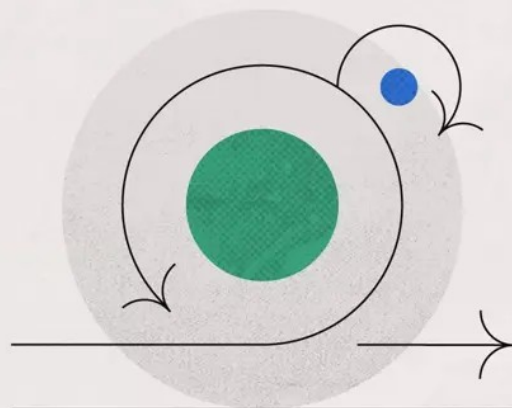
Agile

La metodología ágil o Agile es una **metodología iterativa** que divide los proyectos en varias fases dinámicas conocidas como “sprints”.



Kanban

Kanban es una estrategia visual de **gestión de proyectos** que muestra el trabajo en una vista tipo tablero.



Scrum

Scrum es un **marco ágil** que ayuda a los equipos a colaborar y realizar un trabajo de alto impacto.

Ejemplo Tablero Agil

	Backlog	To DO	Doing	Done
Paula				
Sergio				
Valentina				

El backlog no entra en el sprint.

<https://www.youtube.com/watch?v=6ZBIE0XJU1M>

Diferencia entre los tableros de Kanban y Scrum

Aunque ambos son herramientas visuales usadas en entornos ágiles para gestionar el trabajo, los tableros de Kanban y los tableros de Scrum tienen propósitos, estructuras y usos diferentes.

Aquí las principales diferencias:

Característica	Kanban	Scrum	Scrumban
Enfoque	Gestión continua del flujo de trabajo.	Entregas iterativas en sprints (ciclos fijos de 2-4 semanas).	Donde existe la necesidad de mucha flexibilidad y estructura para mantenerlo todo junto.
Estructura	Columnas fijas (ej: Backlog, To Do, Doing, Done). Las tarjetas se mueven según avanza el trabajo.	También usa columnas, pero está alineado con el sprint actual. Se reinicia o archiva al final de cada sprint.	Similar Kanban
Tiempo	No tiene límites de tiempo. El trabajo fluye constantemente.	Trabaja por sprints con duración definida (por ejemplo, 2 semanas).	Iteraciones de 1-4 semanas
Roles	No define roles específicos.	Requiere roles claros: Scrum Master, Product Owner y Equipo de Desarrollo.	No define roles específicos.
Reuniones	No requiere reuniones formales.	Incluye reuniones obligatorias: Daily Stand-up, Sprint Planning, Review y Retrospective.	No requiere reuniones formales.

Cambios durante el ciclo	Se pueden agregar tareas en cualquier momento	Una vez iniciado el sprint, el contenido no debe cambiar.	Se pueden agregar tareas en cualquier momento
Tablero	Continuo a lo largo del proyecto	Se reinicia después de cada iteración	Continuo a lo largo del proyecto

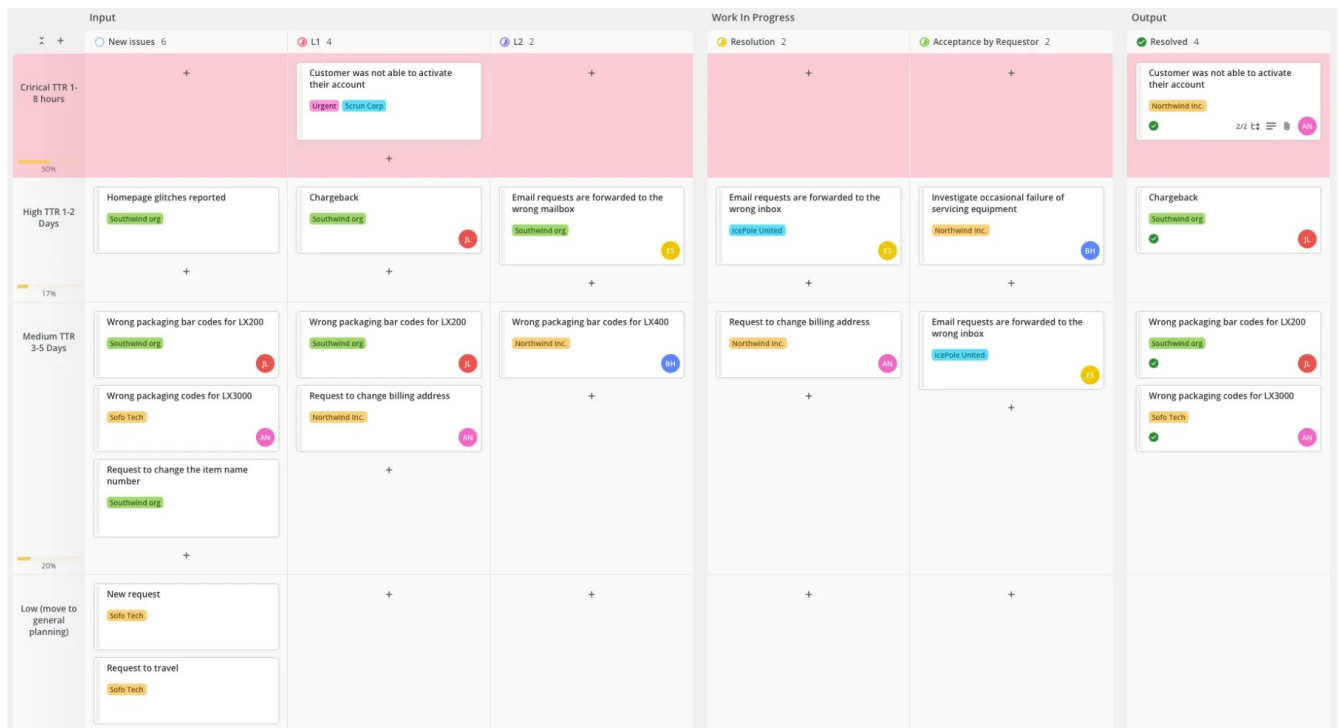
Tablero Kanban:

El tablero de Kanban usualmente está compuesto de columnas que visualizan los pasos del proceso laboral del equipo.

- La sección de Tareas pendientes mantiene todas las ideas y tareas priorizadas para el equipo. Para indicar cuáles tareas son las más importantes, los equipos usan las columnas de prioridad como – Prioridad 1 y Prioridad 2.
- La sección de En progreso está compuesta de tantas columnas como existan pasos en el proceso del equipo. Por ejemplo, diseño, fabricación, prueba y garantía de calidad.
- La sección de Completado mantiene todos los elementos terminados. Algunos equipos podrían desglosar más esta sección para representar las etapas de culminación como 90-95% completado y 100% completado.

Los miembros del equipo de Kanban son responsables de sacar ellos mismos las tareas de las tareas pendientes priorizadas, esto significa que una vez que tengan la capacidad echarán un vistazo a las tareas pendientes y decidirán cuál tarea completar ahora.

Para limitar cuánto trabajo se realiza al mismo tiempo, se usan los límites WIP (Work in Progress - Trabajo en Curso). Usualmente corresponden el número de miembros de equipo y solo permiten 1-2 tareas por miembro de equipo a la vez.

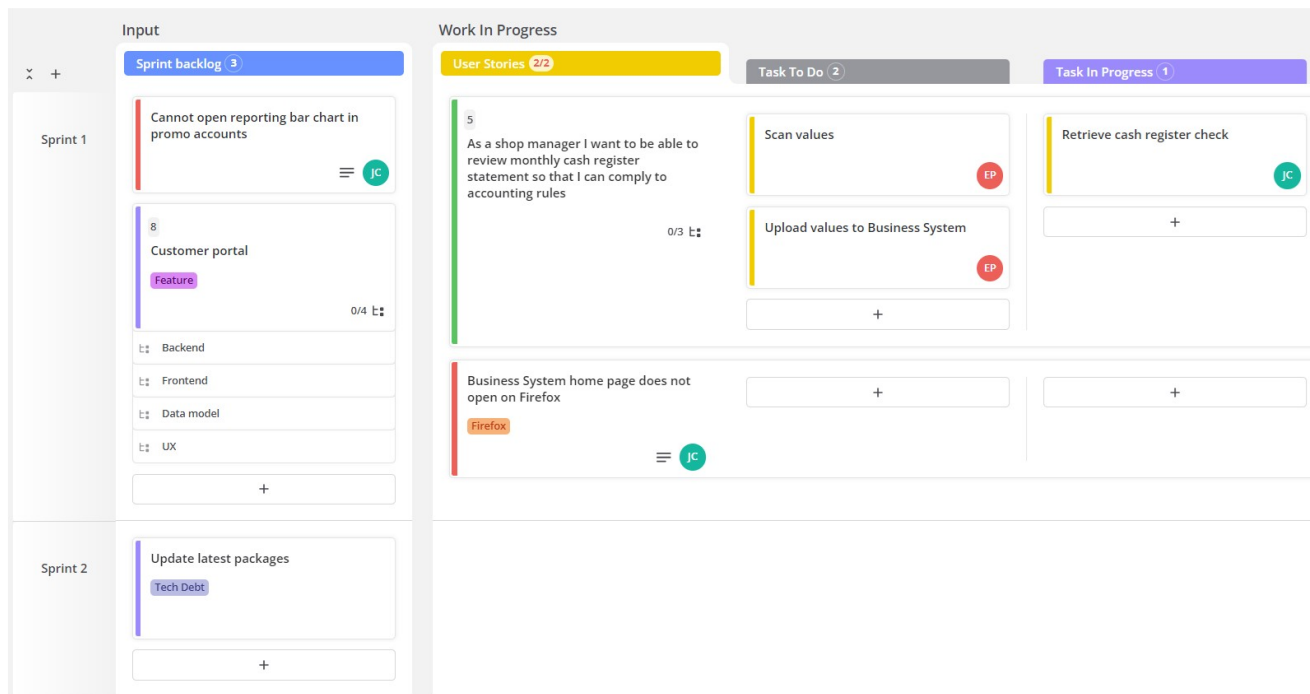


Tablero Scrum:

El tablero de Scrum usualmente está compuesto de las secciones de Tareas pendientes Sprint, En progreso y Completado.

- Tareas pendientes Sprint mantiene las tareas que el equipo está comprometido en completar en Sprint actual.
- La sección de En progreso ayuda a monitorear la ejecución de los elementos comprometidos
- La sección de Listo mantiene los elementos laborales durante el Sprint.

Una vez que Sprint termine, el tablero se despeja y se añaden nuevos elementos laborales para el siguiente Sprint. Si algo no se completa durante el Sprint, se puede mover a la siguiente iteración.



Comparativo:

<https://teamhood.com/wp-content/uploads/2024/07/Hoja-de-referencia-de-Kanban-y-Scrum.pdf>

<https://teamhood.com/es/kanban-4/kanban-scrum-y-scrumban-como-se-diferencian/>

¿Cuándo usar cada uno?

- Usa **Kanban** si:
 - Necesitas flexibilidad total para añadir tareas en cualquier momento.
 - Quieres visualizar el flujo de trabajo y eliminar cuellos de botella.
- Usa **Scrum** si:
 - Necesitas entregas frecuentes y feedback constante del cliente.
 - Tu equipo puede comprometerse con objetivos por sprint.
- Usa **Scrumban** si:
 - En Scrumban, los equipos trabajan en iteraciones definidas similares a Scrum.
 - Tienen una fecha de comienzo y de final establecidas. Las tareas se priorizan y se planifican según esta capacidad.

- Aunque la duración de la iteración se basa en Scrum, el tablero de Scrumban se parece al tablero de Kanban, incluyendo las tareas pendientes continuas y las columnas de prioridad

Ejemplo práctico

- Un equipo de soporte usa Kanban: reciben tickets todo el día y los van moviendo en un tablero (Nuevo → En proceso → Resuelto).
- Un equipo de desarrollo de software usa Scrum: planifican un sprint de 2 semanas, llenan su tablero con tareas acordadas y no cambian el alcance hasta terminarlo.

Herramientas comunes

Ambos tableros se pueden implementar con herramientas como:

- Asana
- Trello
- Jira
- Miro

Estas permiten crear columnas, asignar tarjetas y visualizar el progreso en tiempo real.

- Kanban es ideal para gestionar el flujo continuo de trabajo.
- Scrum es mejor para proyectos con ciclos de entrega definidos y equipos autoorganizados.

Ambos usan tableros visuales, pero Scrum agrega estructura, roles y ritmo, mientras que Kanban se enfoca en la eficiencia del flujo.

<https://miro.com/blog/scrum-kanban-boards-differences/>

La metodología ágil es hoy una de las formas más eficaces de gestionar proyectos de TI, especialmente cuando los requisitos son inciertos o cambiantes.

No es una moda, sino una respuesta madura a la complejidad del desarrollo tecnológico moderno. Como futuros ingenieros de sistemas, dominar este enfoque será clave para liderar proyectos exitosos.

No existe la mejor metodología, sino la más adecuada para tu proyecto.

Fuentes:

<https://www.youtube.com/watch?v=v3fLx7VHxGM>

<https://asana.com/es/resources/agile-methodology>

<https://asana.com/es/resources/pareto-principle-80-20-rule>

<https://asana.com/es/resources/waterfall-agile-kanban-scrum>

<https://miro.com/blog/scrum-kanban-boards-differences/>