1. Scrum

- Framework más común para proyectos de software y entornos colaborativos.
- Divide el trabajo en **sprints** de 2 a 4 semanas, con roles definidos: *Product Owner*, *Scrum Master* y el equipo de desarrollo.
- Incluye reuniones regulares: planificación, daily stand-ups, revisión y retrospectiva.
- Ideal cuando necesitas iteraciones claras, ajustes frecuentes y alta visibilidad del progreso.

Scrum es un **marco de trabajo ágil** para gestionar proyectos, principalmente en el desarrollo de software, pero también aplicable a otros campos. Se basa en el **empirismo** (conocimiento que surge de la experiencia) y el **pensamiento lean** (minimizar el desperdicio). El objetivo es entregar valor de forma rápida y continua en un entorno de incertidumbre. Se distingue por su enfoque en la **transparencia**, **inspección** y **adaptación**.

2. Kanban

- Basado en un tablero visual de flujo (columnas de tareas).
- Se centra en **limitar el trabajo en curso (WIP)** y **optimizar flujos continuos**, sin sprints rígidos.
- Perfecto para equipos con tareas variables o entornos que requieren entregas frecuentes y adaptabilidad.

Kanban es un **marco visual para gestionar el trabajo** que busca mejorar la eficiencia y el flujo de tareas. A diferencia de Scrum, que se basa en ciclos de tiempo fijos (Sprints), Kanban se centra en un **flujo continuo** de trabajo. Su objetivo principal es **visualizar el trabajo, limitar el trabajo en curso (WIP)** y **maximizar el flujo**.

3.Extreme Programming (XP)

- Enfocado en prácticas técnicas intensas: **pair-programming**, **Test-Driven Development (TDD)**, integración continua y propiedad colectiva del código.
- Requiere alta disciplina y colaboración cercana con el cliente.

- Recomendado para equipos pequeños con desarrolladores experimentados.

Extreme Programming (XP) es un marco de trabajo ágil que se enfoca en la **ingeniería de software de alta calidad** y en la **entrega continua de valor** para el cliente. Es una de las metodologías ágiles más disciplinadas, y sus prácticas están diseñadas para funcionar en conjunto, reforzarse unas a otras para crear un entorno de desarrollo sostenible.

4. Lean Software Development

- Adaptación de los principios de Lean Manufacturing: **eliminar desperdicios**, **mejorar procesos** y **entregar valor rápido.**

- Muy útil para organizaciones que buscan eficiencia y mejora continua con pocos recursos.

El **Desarrollo de Software Lean (Lean Software Development)** es una metodología ágil que adapta los principios del sistema de producción de **Toyota** al desarrollo de software. Su objetivo principal es **entregar valor al cliente de la manera más eficiente posible**, eliminando todo lo que no añade valor (lo que se conoce como "desperdicio" o "muda" en japonés).

A diferencia de Scrum o XP, Lean no es un conjunto de roles, artefactos o ceremonias, sino un **conjunto de principios de pensamiento** que guían la toma de decisiones y las prácticas de un equipo. Su filosofía es maximizar el valor del cliente minimizando el desperdicio.

5. Feature-Driven Development (FDD)

- Centrado en descomponer el producto en **features** pequeñas, con un flujo definido: modelado, lista de features, planificación, diseño y construcción

- Bueno para proyectos grandes con requisitos bien definidos y necesidad de visibilidad y calidad rigurosa.

Feature-Driven Development (FDD), o Desarrollo Dirigido por Funcionalidades, es un modelo de desarrollo de software **iterativo e incremental** que se centra en el concepto de "funcionalidad" (feature). A diferencia de otras metodologías ágiles que se centran en historias de usuario o tareas, FDD organiza todo el proceso de desarrollo alrededor de la creación de funcionalidades. Su principal objetivo es entregar software de forma **rápida y fiable** en proyectos a gran escala.

6. Crystal

- Familia de metodologías (*Crystal Clear*, *Crystal Yellow*, *Crystal Orange*, etc.) adaptadas según el tamaño del equipo y la criticidad del sistema.

- Enfatiza la comunicación, simplicidad, adaptabilidad y satisfacción del equipo.

Crystal es un conjunto de metodologías de desarrollo de software, no una sola. Fue creado por Alistair Cockburn, uno de los autores del Manifiesto Ágil, quien reconoció que no existe una única "receta" que funcione para todos los proyectos. En su lugar, Crystal se enfoca en la idea de que los proyectos son como cristales: cada uno tiene sus propias características y requiere un enfoque único.

7. DSDM (Dynamic Systems Development Method)

- Marco ágil con fuerte enfoque en **gobernanza**, roles claros y priorización de requisitos con la técnica **MoSCoW** (Must, Should, Could, Won't).

- Coste, calidad y tiempo se fijan al inicio; es una opción estructurada para organizaciones grandes que requieren previsibilidad.

DSDM es una metodología ágil de gestión de proyectos que se enfoca en entregar proyectos a tiempo y dentro del presupuesto, manteniendo la calidad y la colaboración. Fue creada en el Reino Unido en la década de 1990 para proporcionar un marco estructurado que abordara la falta de control y gobernanza en las primeras metodologías de desarrollo rápido de aplicaciones (RAD).

DSDM se basa en la idea de que **el 80% del valor de un proyecto se puede entregar en el 20% del tiempo**. En lugar de intentar entregar todo, se enfoca en las funcionalidades más importantes y de mayor valor para el negocio. A diferencia de otras metodologías ágiles que pueden ser muy ligeras, DSDM ofrece un marco completo, con roles, procesos y directrices muy bien definidos.

8. Adaptive Software Development (ASD)

- Enfatiza adaptación continua: **especulación, aprendizaje y colaboración** durante todo el ciclo de desarrollo.

- Permite iteraciones no lineales y dinámicas, ideal en entornos altamente cambiantes.

Adaptive Software Development (ASD), o Desarrollo de Software Adaptativo, es una metodología ágil que se enfoca en la **adaptación constante** y la **experimentación** en entornos de alta incertidumbre. Surgió como una evolución del Desarrollo de Aplicaciones

Rápido (RAD) y fue concebida por Jim Highsmith para proyectos donde los requisitos son volátiles y el camino a seguir no está claro.

9. Scrumban

- Híbrido de Scrum y Kanban pensado para facilitar la transición entre ambos.
- Combina iteraciones breves con tablero visual y límites WIP; no impone roles nuevos: el equipo mantiene los que ya tiene.
- Útil cuando se busca flexibilidad sin descartar estructura.

Scrumban es un marco de trabajo híbrido que combina la estructura de **Scrum** con el sistema visual y la gestión del flujo de **Kanban**. No es una metodología formal, sino un enfoque pragmático que los equipos adoptan para evolucionar su forma de trabajar, a menudo cuando sienten que Scrum es demasiado rígido o que Kanban no ofrece suficiente estructura.

10.SAFe (Scaled Agile Framework)

- Marco para escalar prácticas ágiles a nivel empresarial, integrando Lean, Agile y pensamiento sistémico.
- Soporta coordinación entre múltiples equipos, alineación estratégica y gobernanza formal (última versión 6.0, marzo de 2023).
- Para organizaciones grandes que necesitan consistencia y escalabilidad.

SAFe es un marco de trabajo que permite a las grandes organizaciones implementar principios y prácticas ágiles y Lean a escala empresarial. A diferencia de metodologías como Scrum, que se centran en equipos pequeños, SAFe proporciona una estructura y una guía para coordinar a cientos, e incluso miles, de personas en múltiples equipos ágiles.

11. Disciplined Agile (DA / DAD)

- Toolkit contextualizado que combina prácticas Agile y Lean, ayudando a elegir el workflow más adecuado al entorno particular de la organización.

Disciplined Agile (DA) es un **"kit de herramientas" de procesos** que ayuda a las organizaciones a encontrar la mejor forma de trabajar (**Way of Working** o **WoW**) para sus proyectos específicos. A diferencia de metodologías como Scrum o XP, que ofrecen un conjunto fijo de reglas y prácticas, DA reconoce que cada equipo, proyecto y organización es único. Por lo tanto, no hay una única "receta" que funcione para todos.