**1. Metodología Cascada (Waterfall)**

La **metodología Cascada** es un enfoque tradicional y secuencial para la gestión de proyectos, especialmente en el desarrollo de software. En este enfoque, el proyecto se divide en fases claras y ordenadas que deben completarse una tras otra. Estas fases generalmente incluyen:

* **Requisitos**: Identificación de las necesidades del cliente.
* **Diseño**: Creación de la arquitectura y los planes del sistema.
* **Desarrollo**: Implementación del código o sistema según el diseño.
* **Pruebas**: Verificación y validación del producto.
* **Implementación**: Despliegue del sistema en el entorno de producción.
* **Mantenimiento**: Soporte y actualizaciones tras el lanzamiento.

La principal característica de Cascada es que no se regresa a fases anteriores, lo que implica que cada fase debe completarse antes de pasar a la siguiente. Es adecuado para proyectos con requisitos bien definidos y sin cambios frecuentes.

**2. Metodología Ágil (Agile)**

La **metodología Ágil** es un enfoque de desarrollo de software que enfatiza la flexibilidad, la colaboración y la entrega incremental. A diferencia de la metodología Cascada, Ágil permite que los requisitos y soluciones evolucionen a través de la colaboración constante entre equipos multifuncionales y con los clientes.

Algunas de las características principales de Ágil son:

* **Iteraciones cortas** (llamadas *sprints* o *ciclos*), donde se entregan incrementos funcionales del producto.
* **Adaptabilidad**: Cambios en los requisitos son bienvenidos incluso en fases avanzadas del desarrollo.
* **Enfoque en la colaboración** con el cliente y la retroalimentación continua.

Existen diferentes marcos de trabajo y enfoques dentro de Ágil, como **Scrum** y **Kanban**, que ayudan a gestionar el flujo de trabajo de manera eficiente.

**3. Scrum**

**Scrum** es una de las implementaciones más populares de la metodología Ágil. Se centra en entregar incrementos de software en ciclos cortos, llamados *sprints* (generalmente de 2 a 4 semanas).

Las características clave de Scrum son:

* **Roles definidos**:
  + **Product Owner**: Responsable de la visión del producto y priorización del trabajo.
  + **Scrum Master**: Facilita el proceso Scrum y ayuda a resolver obstáculos.
  + **Equipo de Desarrollo**: Ejecuta el trabajo y entrega el producto funcional.
* **Eventos**: Scrum tiene reuniones regulares llamadas *ceremonias*, como las *Daily Standups* (reuniones diarias) y las *Sprint Reviews* (revisiones del sprint).
* **Artefactos**: Incluye el *Product Backlog* (lista de tareas pendientes), *Sprint Backlog* (tareas específicas para el sprint), y el *Incremento* (el producto terminado al final de cada sprint).

Scrum es muy útil en entornos donde los requisitos son cambiantes y la colaboración constante con el cliente es crucial.

**4. Kanban**

**Kanban** es un enfoque visual y flexible para la gestión de proyectos, particularmente adecuado para la mejora continua y la optimización del flujo de trabajo. Aunque también puede utilizarse en el contexto ágil, Kanban se distingue por no usar ciclos fijos (como los *sprints* en Scrum), sino que se enfoca en la gestión continua de tareas.

Las características clave de Kanban son:

* **Tableros Visuales**: Utiliza un tablero (físico o digital) donde las tareas se representan mediante tarjetas movibles entre columnas que reflejan el estado del trabajo (Ej. "Por hacer", "En progreso", "Hecho").
* **Limitar el trabajo en curso** (*WIP - Work In Progress*): Limita la cantidad de tareas que pueden estar en cada columna del tablero al mismo tiempo, lo que ayuda a identificar cuellos de botella y mejorar la eficiencia.
* **Enfoque en el flujo continuo**: Las tareas se entregan de manera continua a medida que se completan, en lugar de trabajar por ciclos.

Kanban es especialmente útil en entornos donde los proyectos requieren flexibilidad y los equipos se enfrentan a flujos de trabajo impredecibles.