

**Escuela Politécnica
Nacional**



Metodologías de Desarrollo de Software

GR3ISW

Autor: Fernando Huilca

Fecha 02-12-2024

Contenido

Introducción.....	3
Metodologías de desarrollo de Software	3
1. Metodología Incremental	3
2. Metodología Kanban	4
3. Metodología XP	5
Listado de empresas y metodologías.....	6
Elección de metodología para un proyecto grande	7
Referencias.....	8

Ilustración 1 Esquema de modelo incremental extraído de [5]	4
Ilustración 2 Ejemplo de metodología Kanban [11]	5
Ilustración 3 Fases de XP [13].....	6

Introducción

El desarrollo de software es un proceso que requiere y exige del ingeniero una planificación y ejecución meticulosa para satisfacer las demandas de sus clientes en un mercado dinámico y en constante evolución, con gran competencia en tiempos de entrega y exigencias en documentación.

Debido a ello se han desarrollado diversas estrategias con el objetivo de abordar las diferentes facetas del desarrollo de software, desde lo riguroso y estricto como son las metodologías tradicionales hasta lo flexible y humano presentes en las metodologías ágiles.

Según el autor: “Una metodología de desarrollo de software es similar a una receta de cocina. De la misma manera que una receta le enseña a uno cómo cocinar una comida, un método de desarrollo de software enseña cómo construir un producto de software” (p.2) [1].

Este documento analiza tres métodos recientes de trabajo de software, las características, las etapas y los resultados de estos métodos, destacando sus puntos fuertes y sus usos en diferentes situaciones. Además, se proporciona una evaluación de empresas destacadas y las técnicas que utilizan.

Finalmente, se presenta la elección de una estrategia adecuada para el desarrollo de un software grande con duración mayor a 6 meses, teniendo en cuenta elementos como la complejidad y requisitos del cliente.

Metodologías de desarrollo de Software

1. Metodología Incremental

El modelo incremental es una metodología que combina elementos del enfoque tradicional y las características iterativas propias de metodologías ágiles. Su objetivo principal es desarrollar el producto en incrementos funcionales, donde cada entrega aporta una mejora concreta y progresiva al resultado final. Esto permite una evolución constante hasta cumplir con los requisitos del cliente o usuario final [2][3].

Características Principales

- **Incrementos pequeños y acumulativos:** Cada iteración del proyecto introduce un avance funcional sobre la versión previa, garantizando una mejora constante [3].
- **Flexibilidad ante cambios:** Permite adaptarse a nuevas necesidades o requisitos surgidos durante el desarrollo del proyecto [4].
- **Entrega continua de valor:** Proporciona resultados tangibles en cada fase, lo que facilita la evaluación del progreso y reduce los riesgos [3].

El modelo incremental sigue un ciclo de vida adaptativo que puede dividirse en las siguientes fases:

1. **Recolección de Requisitos:**
En esta fase se identifican los objetivos generales y específicos del proyecto. [2].
2. **Definición de Iteraciones:**
Las tareas se agrupan en iteraciones, cada una con un objetivo claro y específico. [4].
3. **Diseño de Incrementos:**
En esta etapa se define cómo evolucionará el producto en cada iteración. [2][3].
4. **Desarrollo del Incremento:**
Los incrementos se construyen siguiendo las especificaciones previamente definidas. Aquí se realizan las tareas asignadas para cumplir con los objetivos de la iteración [2][4].
5. **Validación de Incrementos:**
Cada iteración termina con una revisión exhaustiva para verificar que los objetivos establecidos se hayan cumplido. Si hay fallos, estos se corrigen antes de avanzar [3][4].
6. **Integración de Incrementos:**
Los incrementos validados se integran para formar una versión acumulativa del producto, creando una línea de evolución coherente del proyecto [2][4].

7. Entrega del Producto Final:

Una vez que todos los incrementos han sido validados e integrados, se presenta el producto final, asegurando que cumple con los requisitos establecidos inicialmente [2].

El modelo incremental es ampliamente utilizado en sectores donde los requisitos tienden a cambiar o donde se requiere una entrega progresiva de funcionalidades. Herramientas como Scrum y Kanban suelen emplear este enfoque debido a su capacidad de gestionar proyectos mediante incrementos pequeños y controlados [3].

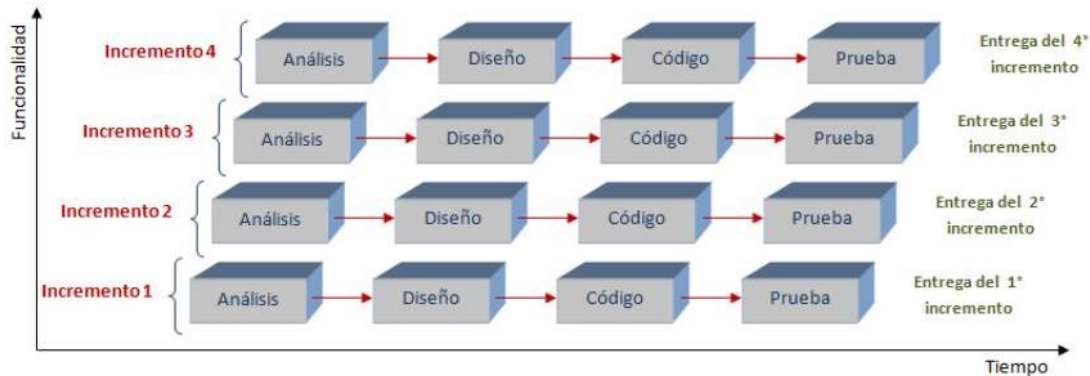


Ilustración 1 Esquema de modelo incremental extraído de [5]

2. Metodología Kanban

Kanban es una metodología ágil que se centra en la visualización del flujo de trabajo. Originalmente desarrollado como parte del sistema de producción de Toyota, se ha adaptado ampliamente al desarrollo de software y otros sectores debido a su flexibilidad y simplicidad. Su propósito principal es optimizar el flujo de trabajo, limitando el trabajo en curso (WIP, *Work In Progress*) y asegurando entregas constantes y de alta calidad [6][7].

Características Principales de Kanban

1. **Visualización del flujo de trabajo:**
Kanban utiliza un tablero dividido en columnas que representan las diferentes etapas del proceso. Las tareas se representan como tarjetas que avanzan de una columna a otra, lo que facilita la identificación de cuellos de botella y tareas pendientes [8][9].
2. **Límite de trabajo en curso (WIP):**
Establecer límites en la cantidad de tareas que pueden estar en progreso simultáneamente permite evitar la sobrecarga y mejorar la eficiencia [7].

Fases o Componentes:

Kanban no tiene fases rígidas, su implementación se puede estructurar en los siguientes pasos:

1. **Visualización del trabajo:**
Crear un tablero Kanban, físico o digital, donde se muestren las tareas en columnas como "Por hacer," "En progreso" y "Finalizado."
2. **Definición de políticas claras:**
Establecer reglas claras para mover tareas entre las columnas y criterios para priorizar el trabajo.
3. **Establecimiento de límites WIP:**
Definir cuántas tareas pueden estar en progreso en cada columna. Esto ayuda a mantener un enfoque en tareas específicas y evitar interrupciones [8].

4. **Revisión y mejora continua:**

Realizar reuniones periódicas para analizar el flujo y ajustar los límites de WIP o las políticas según sea necesario [9][10].

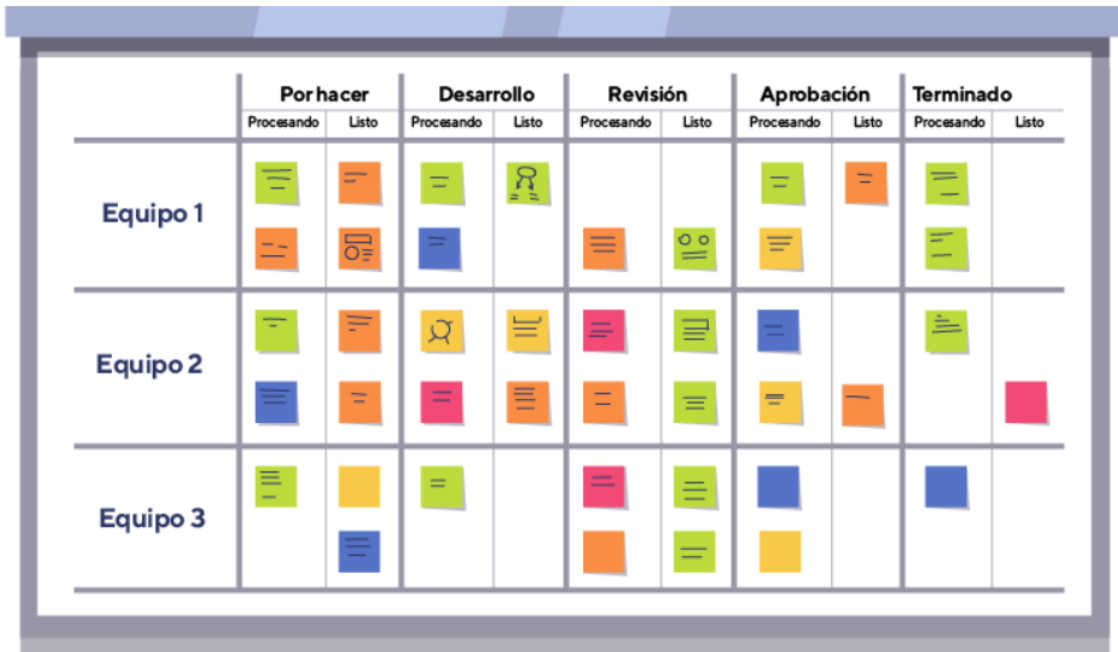


Ilustración 2 Ejemplo de metodología Kanban [11]

3. Metodología XP

Es una metodología ágil que pone énfasis en la colaboración, la adaptabilidad y la entrega constante de valor al cliente. Se centra en la mejora continua del software mediante iteraciones cortas y prácticas disciplinadas que aseguran la calidad técnica del producto. XP es especialmente adecuada para proyectos con requisitos cambiantes o ambiguos y un alto nivel de incertidumbre técnica [12].

Características:

1. **Historias de Usuario:**

Los requisitos se expresan en historias de usuario, breves descripciones funcionales que el cliente define y prioriza. Estas son la base para planificar cada iteración [12]

2. **Pruebas Constantes:**

Las pruebas unitarias y funcionales son clave en XP. Se desarrollan antes del código y se ejecutan de forma continua, lo que asegura la calidad del software y facilita la detección temprana de errores [12].

3. **Programación en Parejas:**

Todo el código es escrito por dos desarrolladores trabajando juntos. Esto mejora la calidad del código, fomenta el aprendizaje compartido y acelera la resolución de problemas [12].

4. **Refactorización:**

El código se mejora constantemente para mantenerlo limpio, legible y adaptable a cambios futuros [12].

5. **Iteraciones Cortas y Entregas Frecuentes:**

El trabajo se organiza en ciclos de desarrollo de pocas semanas, garantizando entregas funcionales y ajustadas a las prioridades del cliente [12].

Ciclo de vida fases:

1. **Exploración:**
Se identifican historias de usuario iniciales y se evalúan las herramientas y tecnologías necesarias para el proyecto.
2. **Planificación de la Entrega:**
Se establecen prioridades y se realiza una estimación del esfuerzo para cada historia.
3. **Iteraciones:**
Se desarrollan incrementos del software mediante ciclos cortos, integrando las funcionalidades prioritarias.
4. **Producción:**
El sistema se prueba y se optimiza antes de ser implementado en el entorno del cliente.
5. **Mantenimiento:**
Se asegura la operación continua del software mientras se desarrollan mejoras y nuevas funcionalidades.
6. **Cierre del Proyecto:**
Se completa el desarrollo cuando el cliente no requiere más cambios o el producto alcanza su máxima funcionalidad [12].

Extreme Programming

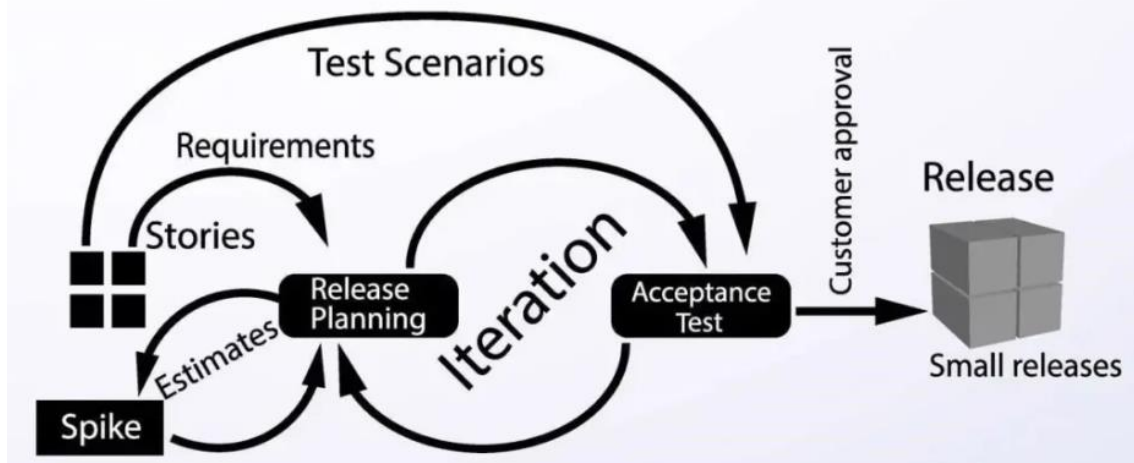


Ilustración 3 Fases de XP [13]

Listado de empresas y metodologías

Empresa de Software	Metodología aplicada	Resumen
Microsoft	Scaled Agile Framework - SAFe	Microsoft utiliza SAFe para escalar la agilidad en sus equipos globales. Esta metodología les permite coordinar múltiples equipos en proyectos complejos.
Intel	Agile y Scrum	Intel emplea Scrum para proyectos de desarrollo de software, combinándolo con otros enfoques ágiles. Esto le permite adaptarse rápidamente a los cambios
Cisco	DevOps	Cisco adopta DevOps para fomentar la colaboración entre desarrollo y operaciones, lo que acelera el lanzamiento de actualizaciones y mejora la calidad del software

WhatsApp	Kanban	WhatsApp utiliza Kanban para administrar su flujo de trabajo, priorizando tareas clave y asegurando lanzamientos frecuentes con alta calidad
Oracle	Waterfall y Agile	La empresa utiliza Waterfall para desarrollos grandes y estructurados, como bases de datos y software empresarial. En proyectos modernos, emplea Agile para fomentar la flexibilidad y adaptarse a los requisitos cambiantes.

Elección de metodología para un proyecto grande

A continuación, presento un pequeño resumen de donde y cuando aplicar las metodologías que se estudiaron en este documento:

Metodología	Duración	Contexto Ideal
XP	3 a 9 meses	Requisitos cambiantes y equipos pequeños
Incremental	6 meses a 2 años	Módulos definidos y entregas parciales
Kanban	Flexible (Cualquier tiempo)	Trabajo continuo, proyectos sin plazos rígidos.

Si estuviera liderando la construcción de un software grande con una duración de **más de 6 meses**, consideraría varias variables clave, como la complejidad del proyecto, los requisitos cambiantes, la colaboración con el cliente y los equipos involucrados. Basado en estas características, la metodología más adecuada sería **el Modelo Incremental** por su capacidad de gestionar proyectos de gran escala.

Porque me permitiría una flexibilidad en el desarrollo del trabajo dividiéndolos en módulos y dándome la posibilidad de hacer entregables de manera progresiva haciendo feedback del cliente, así también puedo reducir riesgos antes y durante el desarrollo tomando en cuenta que tendré que trabajar con un equipo grande por más de 6 meses.

Referencias

- [1] J. H. Canós, P. Letelier, y M. C. Penadés, "Metodologías ágiles en el desarrollo de software," Universidad Politécnica de Valencia, Valencia, España, pp. 1–8, 2003.
- [2] A. Pérez, "Características y fases del modelo incremental," *OBS Business School*, 2024. [Online]. Available: <https://www.obsbusiness.school/blog/caracteristicas-y-fases-del-modelo-incremental>. [Accessed: Dec. 1, 2024].
- [3] R. S. Pressman, *Ingeniería de Software: Un Enfoque Práctico*, 7th ed. Madrid, España: McGraw-Hill, 2010.
- [3] K. Beck et al., "Manifiesto for Agile Software Development," *Agile Alliance*, 2001. [Online]. Available: <https://agilemanifesto.org/>. [Accessed: Dec. 1, 2024].
- [5] Mwebs, "¿Qué es el desarrollo incremental?," *Mwebs*, 2023. [Online]. Available: <https://mwebs.com.uy/blog/qué-es-el-desarrollo-incremental/23>. [Accessed: Dec. 1, 2024].
- [6] D. J. Anderson, *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press, 2010.
- [7] K. Beck et al., "Manifiesto for Agile Software Development," *Agile Alliance*, 2001. [Online]. Available: <https://agilemanifesto.org/>. [Accessed: Dec. 1, 2024].
- [8] J. H. Canós, P. Letelier, y M. C. Penadés, "Metodologías ágiles en el desarrollo de software," Universidad Politécnica de Valencia, Valencia, España, pp. 1–8, 2003.
- [9] Mwebs, "¿Qué es el desarrollo incremental?," *Mwebs*, 2023. [Online]. Available: <https://mwebs.com.uy/blog/qué-es-el-desarrollo-incremental/23>. [Accessed: Dec. 1, 2024].
- [10] H. Kniberg y M. Skarin, *Kanban and Scrum - Making the Most of Both*. C4Media, 2010.
- [11] Initium Soft, "Metodología Kanban: Cómo mejorar la productividad en equipos de trabajo," *Initium Soft*, 2024. [Online]. Available: https://www.initiumsoft.com/blog_initium/metodologia-kanban/. [Accessed: Dec. 1, 2024].
- [12] P. Letelier y M. C. Penadés, "Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)," *Técnica Administrativa*, vol. 5, no. 26, pp. 1–25, 2006. [Online]. Available: <https://www.cyta.com.ar/ta/article.php?id=050201>.
- [13] Nimblework, "Programación Extrema (XP): Una metodología ágil para el desarrollo de software," *Nimblework*, 2024. [Online]. Available: <https://www.nimblework.com/es/agile/programacion-extrema-xp/>. [Accessed: Dec. 1, 2024].