DIPLOMADO

FORMACIÓN PROFESIONAL EN PROYECTOS

- **DOBLE CERTIFICACIÓN INTERNACIONAL**





Humans PRO

MARIO ERASO

Docente – Instructor certificado
PMC® - Social Project Management® -
SCRUM Master Professional



Metodologías Ágiles (SCRUM)

BIENVENIDOS



Origen y evolución:

Comprender el surgimiento de los modelos ágiles como respuesta a las limitaciones del enfoque tradicional predictivo (modelo en cascada), contextualizando la evolución histórica de la gestión de proyectos y el desarrollo de software.

ORIGEN Y EVOLUCIÓN: EL ENFOQUE TRADICIONAL PREDICTIVO:

- Definición: planificación detallada desde el inicio del proyecto, con alcance, cronograma y costos fijos.
- Etapas típicas: Inicio → Planeación → Ejecución → Seguimiento y control → Cierre.
- Ejemplo: PMBOK y su aplicación en industrias como la construcción o la manufactura.

ORIGEN Y EVOLUCIÓN: EL MODELO EN CASCADA (WATERFALL)

Introducción por Winston Royce (1970).

- Etapas: Requisitos → Diseño → Implementación → Verificación → Mantenimiento.
- Características:

1. Lineal y secuencial.
2. Poca tolerancia al cambio.
3. Fuerte análisis de documentación.
4. Entrega al final del ciclo.

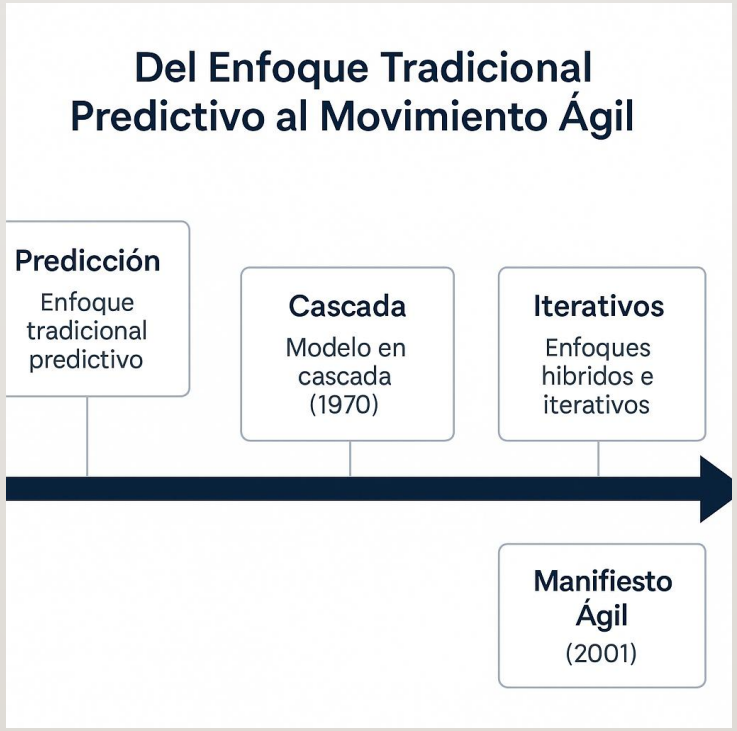
ORIGEN Y EVOLUCIÓN: BÚSQUEDA DE ALTERNATIVAS ITERATIVAS

- Enfoques híbridos: Spiral, RUP (Rational Unified Process)
- Introducción al pensamiento empírico: aprender haciendo, feedback temprano, entregas parciales.

ORIGEN Y EVOLUCIÓN: ORIGEN DEL MANIFIESTO ÁGIL (2001)

- Reunión de 17 desarrolladores en Snowbird, Utah.
- Reacción a la rigidez del modelo tradicional.
- Nacimiento del paradigma ágil basado en colaboración, adaptación y entrega de valor.

ORIGEN Y EVOLUCIÓN: LÍNEA DE TIEMPO



¿POR QUÉ ÁGIL?

- **Propósito:**
  Fomentar la argumentación crítica y el análisis comparado entre el enfoque ágil y el tradicional (predictivo) en la gestión de proyectos, reflexionando sobre su aplicabilidad en diversos contextos, tanto del sector público como privado.

LOS PRINCIPIOS DEL MANIFIESTO ÁGIL

1. Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.

➤ Explicación: se prioriza entregar versiones funcionales desde el inicio, no al final del proyecto.

➤ Ejemplo: en lugar de entregar una plataforma completa en 6 meses, se entrega un módulo útil en 2 semanas (como el portal de registro de usuarios).

LOS PRINCIPIOS DEL MANIFIESTO ÁGIL

2. Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.

➤ Explicación: los cambios no son un problema, sino una oportunidad para mejorar el producto.

➤ Ejemplo: el cliente pide agregar integración con WhatsApp a mitad del proyecto; se ajusta el backlog y se prioriza para el siguiente sprint.

LOS PRINCIPIOS DEL MANIFIESTO ÁGIL

3. Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia por los períodos de tiempo más cortos.

➤ Explicación: se trabaja en ciclos cortos (sprints) donde cada entrega es funcional y usable.

➤ Ejemplo: cada sprint de 2 semanas produce una app usable con nuevas funciones.

LOS PRINCIPIOS DEL MANIFIESTO ÁGIL

4. Los responsables del negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.

➤ Explicación: se rompe la barrera entre áreas técnicas y de negocio.

➤ Ejemplo: el Product Owner (de negocio) participa a diario en reuniones con los desarrolladores para alinear prioridades.

LOS PRINCIPIOS DEL MANIFIESTO ÁGIL

5. Construimos proyectos en torno a individuos motivados. Les damos el entorno y el apoyo que necesitan y confiamos en que hagan bien su trabajo.

➤ Explicación: la motivación y autonomía del equipo son clave; se evita la microgestión.

➤ Ejemplo: los miembros del equipo proponen soluciones y gestionan sus tareas sin órdenes jerárquicas.

LOS PRINCIPIOS DEL MANIFIESTO ÁGIL

6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.

➤ Explicación: se favorece la comunicación directa (o videollamadas en entornos virtuales).

➤ Ejemplo: si surge un problema, se discute de inmediato en una daily meeting, no por correos largos.

LOS PRINCIPIOS DEL MANIFIESTO ÁGIL

7. El software funcionando es la medida principal de progreso.

➤ Explicación: no se mide por tareas completadas o documentos, sino por entregas que funcionan.

➤ Ejemplo: una pantalla que se diseñó pero no se conecta a la base de datos aún no cuenta como progreso.

LOS PRINCIPIOS DEL MANIFIESTO ÁGIL

8. Los procesos ágiles promueven el desarrollo sostenible. Los patrocinadores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de manera indefinida.

➤ Explicación: se busca un ritmo equilibrado, evitando el agotamiento (burnout).

➤ Ejemplo: los equipos trabajan 40 horas semanales constantes, no con picos de 70 horas antes de cada entrega.

LOS PRINCIPIOS DEL MANIFIESTO ÁGIL

9. La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.

- Explicación: se invierte tiempo en calidad técnica para facilitar futuros cambios.
- Ejemplo: se hace refactorización de código regularmente y se aplican pruebas automáticas.

LOS PRINCIPIOS DEL MANIFIESTO ÁGIL

10. La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.

- Explicación: se elimina lo innecesario, se enfoca solo en lo valioso.
- Ejemplo: si una función no aporta valor claro al usuario, se descarta aunque esté planeada.

LOS PRINCIPIOS DEL MANIFIESTO ÁGIL

11. Las mejores arquitecturas, requisitos y diseños emergen de equipos autoorganizados.

- Explicación: los equipos toman decisiones técnicas y de diseño sin depender de jefes externos.
- Ejemplo: el equipo decide cuál framework utilizar y cómo organizar los microservicios.

LOS PRINCIPIOS DEL MANIFIESTO ÁGIL

12. A intervalos regulares, el equipo reflexiona sobre cómo ser más efectivo y ajusta su comportamiento en consecuencia.

➤ Explicación: se realizan retrospectivas frecuentes para mejorar continuamente.

➤ Ejemplo: después de cada sprint, el equipo identifica qué funcionó, qué no, y aplica mejoras inmediatas.

LOS PRINCIPIOS DEL MANIFIESTO ÁGIL: ACTIVIDAD

- Discusión en grupos pequeños (salas en Zoom):

“¿Cuál principio les parece más difícil de aplicar y por qué?”

COMPARATIVA DE 5 METODOLOGÍAS ÁGILES

- SCRUM
- KANBAN
- Extreme Programming (XP)
- Lean Software Development
- Feature-Driven Development (FDD)

Metodologías Ágiles: SCRUM

Descripción:

Scrum es una metodología ágil basada en un enfoque iterativo e incremental para la gestión de proyectos. Se estructura en ciclos cortos llamados “sprints” (generalmente de 2 a 4 semanas), en los cuales se entregan incrementos funcionales del producto.

Roles principales:

- Product Owner: define y prioriza el trabajo.
- Scrum Master: facilita el proceso, elimina impedimentos.
- Development Team: equipo multidisciplinario que ejecuta el trabajo.

Eventos:

- Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective.

METODOLOGÍAS ÁGILES: SCRUM

Ejemplo:

Una empresa está desarrollando una plataforma para gestionar servicios públicos. En cada sprint, el equipo entrega funcionalidades completas, como el registro de usuarios, el módulo de pagos o la gestión de reclamos. Cada dos semanas hacen una revisión con el cliente para ajustar prioridades.

Fortalezas:

- Adaptabilidad a cambios.
- Transparencia y mejora continua.
- Foco en entregables funcionales.

Debilidades:

- Requiere disciplina y compromiso del equipo.
- Puede ser difícil para organizaciones con estructuras jerárquicas.

METODOLOGÍAS ÁGILES: KANBAN

Descripción:

Kanban es un método visual para gestionar el flujo de trabajo. Se basa en un tablero con columnas (Ej.: Por hacer, En progreso, Hecho) y tarjetas que representan tareas. Se trabaja en flujo continuo y se limitan las tareas simultáneas (WIP: Work In Progress).

Principios clave:

- Visualización del trabajo.
- Límites de trabajo en curso.
- Mejora continua del flujo.

METODOLOGÍAS ÁGILES: KANBAN

Ejemplo:

Un equipo de mantenimiento TI gestiona incidentes y solicitudes internas. Usan un tablero Kanban donde cada ticket pasa por las columnas. Limitan a 3 tareas simultáneas por persona para evitar sobrecarga. No hay sprints, solo flujo continuo.

Fortalezas:

- Muy visual y fácil de implementar.
- Ideal para entornos de soporte, mantenimiento o flujo continuo.

Debilidades:

- Menos estructura para equipos nuevos.
- No obliga a hacer retrospectivas o reuniones formales.

METODOLOGÍAS ÁGILES: XP (EXTREME PROGRAMMING)

Descripción:

XP se enfoca en el desarrollo de software con alta calidad técnica y capacidad de respuesta rápida al cambio. Promueve prácticas de ingeniería rigurosas y trabajo colaborativo.

Prácticas clave:

- Desarrollo dirigido por pruebas (TDD).
- Programación en pareja.
- Integración continua.
- Pequeñas versiones frecuentes.

METODOLOGÍAS ÁGILES: XP (EXTREME PROGRAMMING)

Ejemplo:

Una fintech desarrolla su nueva API de préstamos. El equipo crea primero las pruebas automatizadas (TDD) antes de codificar, hacen programación en pareja (dos desarrolladores por tarea), y suben cambios constantemente al repositorio para evitar errores tardíos.

Fortalezas:

- Calidad de código excepcional.
- Detecta errores desde etapas tempranas.

Debilidades:

- Alta exigencia técnica.
- No apta para todos los equipos ni todos los productos.

METODOLOGÍAS ÁGILES: LEAN SOFTWARE DEVELOPMENT

Descripción:

Lean proviene del sistema de producción de Toyota y se adapta al desarrollo de software. Su objetivo es maximizar el valor entregado al cliente reduciendo el desperdicio (tiempo, trabajo innecesario, errores, esperas).

Principios clave:

- Eliminar desperdicio.
- Entrega rápida.
- Calidad integrada.
- Respeto por las personas.
- Mejora continua.

METODOLOGÍAS ÁGILES: LEAN SOFTWARE DEVELOPMENT

Ejemplo:

Una startup lanza una nueva app de reservas. En vez de desarrollar todas las funciones desde el inicio, lanza una versión básica (MVP) en 3 semanas, mide uso real y elimina funciones que nadie usa.

Fortalezas:

- Mentalidad de eficiencia y valor.
- Alineada con innovación y aprendizaje continuo.

Debilidades:

- Puede ser difícil identificar "desperdicios" sin experiencia.
- Pocas guías formales sobre roles o ceremonias.

METODOLOGÍAS ÁGILES: FDD (FEATURE-DRIVEN DEVELOPMENT)

Descripción:

FDD es una metodología ágil enfocada en construir el sistema a partir de funcionalidades (features) significativas para el cliente. Sigue cinco pasos y se apoya en una planificación más detallada que otras metodologías ágiles.

Pasos principales:

1. Modelado general del sistema.
2. Listado de funcionalidades.
3. Planificación por funcionalidad.
4. Diseño por funcionalidad.
5. Construcción por funcionalidad.

METODOLOGÍAS ÁGILES: FDD (FEATURE-DRIVEN DEVELOPMENT)

Ejemplo:

Un banco digital organiza su desarrollo en términos de funcionalidades: "crear cuenta", "transferir fondos", "consultar saldo". Cada funcionalidad se diseña y se desarrolla por equipos pequeños y especializados, siguiendo un modelo previamente establecido.

Fortalezas:

- Enfoque centrado en valor entregado.
- Útil para proyectos grandes y estructurados.

Debilidades:

- Menos flexible ante cambios que Scrum o Kanban.
- Requiere roles y estructura más definidos desde el inicio.

COMPARATIVA DE 5 METODOLOGÍAS ÁGILES

Metodología	Ciclo	Enfoque	Fortalezas	Debilidades
Scrum	Iterativo	Gestión de equipos	Clara estructura, alta adaptabilidad	Requiere madurez del equipo
Kanban	Flujo continuo	Visualización y control	Simple, flexible	Difícil medir velocidad
XP	Iterativo	Calidad técnica	Alta calidad de código	Alta exigencia técnica
Lean	Flujo / Adaptativo	Eficiencia	Elimina desperdicios	Difícil de aplicar en grandes equipos
FDD	Iterativo	Funcionalidades	Orientado a resultados	Poco usado actualmente


BCKLG
TO DO
DOING
DONE
SCRUMBAN
ALINEACIÓN
MOTIVACIÓN
RENDIMIENTO
ZADECON

DIPLOMADO

FORMACIÓN PROFESIONAL EN PROYECTOS

- **DOBLE CERTIFICACIÓN INTERNACIONAL**





Humans PRO