

# Unidad 3: Desarrollo Ágil

## Origen del Desarrollo Ágil

En 2001 se publicó el **Manifiesto Ágil**, firmado por 17 expertos en desarrollo de software (como Kent Beck). Este manifiesto planteó nuevos valores y principios para mejorar el proceso de desarrollo:

**Se valora más:**

- **Individuos e interacciones** sobre procesos y herramientas
- **Software funcionando** más que documentación exhaustiva
- **Colaboración con el cliente** más que negociación contractual
- **Respuesta al cambio** más que seguir un plan rígido

---

## ¿Qué es la Ingeniería de Software Ágil?

Es una forma de trabajo que combina **una filosofía + principios + prácticas livianas**, orientada a:

- Entregar software **funcional e incremental rápidamente**
- Usar **equipos pequeños, motivados y colaborativos**
- Fomentar **la comunicación directa** y continua
- Minimizar documentos y procesos innecesarios

## Diferencias clave con modelos tradicionales

Aspecto	Tradicional (Cascada)	Ágil
Foco	Planificación y documentación	Software funcionando
Entregas	Al final del proyecto	Entregas frecuentes
Cambios	Difíciles de aceptar	Bienvenidos y esperados
Cliente	Recién al final	Participa constantemente
Equipo	Dividido por roles	Multidisciplinario y unido

## Producto Final en Ágil

Lo **único que importa** es entregar un incremento de software **funcional** en el **tiempo prometido**.

Las actividades estructurales clásicas (comunicación, planificación, modelado, construcción y despliegue) **siguen existiendo**, pero se adaptan y simplifican.

---

## 12 Principios del Manifiesto Ágil

1. Entregar software funcional rápido y continuamente
  2. Aceptar cambios incluso tarde
  3. Entregar versiones funcionales cada 2-8 semanas
  4. Trabajar con el cliente **todos los días**
  5. Equipos motivados y con apoyo
  6. Comunicación cara a cara
  7. La principal medida de progreso es el software que funciona
  8. Desarrollo sostenible
  9. Atención continua a excelencia técnica
  10. Simplicidad
  11. Equipos auto-organizados generan mejor diseño
  12. Reflexión y mejora constante
- 

## Factores Humanos Clave

- Competencia técnica
  - Comunicación y colaboración constante
  - Autonomía y auto-organización
  - Confianza y respeto mutuo
  - Capacidad para resolver problemas complejos
- 

## Ejemplos de Métodos Ágiles

### 1. Programación Extrema (XP)

Principios:

- Simplicidad
- Retroalimentación rápida
- Programación en parejas
- Integración continua
- Pruebas constantes (unitarias y de aceptación)

▼ **Desventajas XP:**

- Requisitos muy cambiantes
- Poca documentación
- Riesgo si no se formaliza el diseño

---

## 2. SCRUM (Modelo Ágil por excelencia)

- Trabajo dividido en **Sprints** (iteraciones cortas de 2-4 semanas)
- Roles: Scrum Master, Product Owner, Equipo de desarrollo
- Reuniones: planificación, revisión, retrospectiva y daily stand-ups
- Backlog: lista priorizada de tareas

---

## Otros Métodos Ágiles

- **Desarrollo Adaptativo (DAS)**
- **Método Dinámico de Sistemas (MDSD)**
- **Cristal**
- **Desarrollo Impulsado por Características (DIC)**
- **Desarrollo Esbelto (Lean Software)**
- **Modelado Ágil (AM)**
- **Proceso Unificado Ágil (PUA)**

---

## Complemento desde Pressman (Cap. 3, 7ª edición)

Pressman resalta que la agilidad es **una respuesta a la rigidez de los modelos tradicionales**, especialmente útil cuando:

- Los requisitos son cambiantes

- El cliente necesita resultados rápidos
- El equipo es pequeño y experimentado

El autor también destaca que, aunque Ágil minimiza la documentación, **no significa ausencia de disciplina**, sino un enfoque más práctico y flexible.

## Resumen Final

Concepto	Descripción corta
Ágil	Desarrollo rápido, flexible, centrado en el cliente
Manifiesto	4 valores + 12 principios
XP	Programación colaborativa y testeada
SCRUM	Iteraciones, roles definidos y backlog
Otros métodos	Variaciones adaptadas a equipos y proyectos específicos