




Metodologías ágiles de construcción de software

Ingeniería del Software



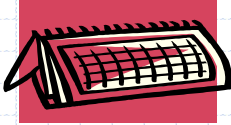
Curso 2025/2026
Universidad San Pablo-CEU
Escuela Politécnica Superior
Campus de Montepríncipe

1




¿Proyecto exitoso?

- » **Jefe de Proyecto:** *¿por qué va tan retrasado este proyecto?*
- » **Programador:** *es que el cliente ha vuelto a cambiar las especificaciones y al modificar el código se cometieron errores en partes que ya estaban terminadas.*
- » **Jefe de Proyecto:** *pero, ¿no sabes lo que nos esta costando esto?*



29-ago.-25 Ingeniería del Software página 2

2



¿Por qué fracasan los proyectos de software?

- » **Retrasos y desviaciones en la planificación**
 - La fecha de entrega siempre está 6 meses en el futuro
 - Pueden llegar a provocar la cancelación del proyecto
- » **Costes de mantenimiento muy elevados**
 - Tras un par de años en operación y algunos cambios, aparecen cada vez más errores
- » **Alta tasa de defectos**
 - El sistema tiene tantos defectos que ni siquiera se usa
- » **Requisitos mal comprendidos**
 - El software no responde las preguntas apropiadas
- » **Cambios de negocio no reflejados en el software**
 - El software responde preguntas erróneas o fuera de fecha
- » **Falsa riqueza de características**
 - Gran cantidad de ellas (casi) no se usan
- » **Rotación de personal**
 - ¿Dónde se ha ido los programadores buenos?

29-ago.-25 Ingeniería del Software página 3

3



Los proyectos de desarrollo


- » En 1990 IBM realiza un estudio a nivel mundial:

Equipos exitosos	Equipos con problemas
<ul style="list-style-type: none">» No habían seguido métodos formales» Habían estimulado la comunicación y las pruebas	<ul style="list-style-type: none">» No entendían sus fracasos... si habían cumplido con los métodos formales
- » La experiencia se repitió por toda la década, por todo el mundo y con todas las herramientas:

Conclusión
Menos énfasis en la documentación exhaustiva y más en versiones que corran y puedan ser probadas

29-ago.-25 Ingeniería del Software página 4

4




¿Cuál es la diferencia?

- » Esquema **tradicional**
 - Mucha documentación: notaciones de modelado, herramientas, diagramas
 - Burocráticas y rígidas: rigurosa definición de actividades, artefactos y roles
 - Efectivo en proyectos de gran envergadura
- » Desarrollos **actuales**
 - El contexto es muy cambiante
 - Se exige reducir el tiempo de desarrollo pero manteniendo una alta calidad de producto.
 - En la práctica, se prescinde del "buen hacer" de la ingeniería del software para ajustarse a estas restricciones
- » **¿El fin de las metodologías?**

29-ago.-25 Ingeniería del Software página 5

5



Las Metodologías Ágiles

- » Las **Metodologías Ágiles (AMs)**
 - Se basan en una serie de reglas y principios que, usados conjuntamente, proporcionan una metodología de desarrollo de software en la que se priorizan las tareas que dan **resultados directos** y que reducen la burocracia existente, tanto como sea posible (pero no más)
 - › Orientadas a la **gente** más que orientadas al proceso
 - › Orientadas al **código** en lugar de a la documentación
 - › Orientados hacia la **entrega de código que funciona**
 - › **Comunicación** intensiva
 - Son **adaptables** en lugar de predictivas
 - Llenan el vacío metodológico
 - Orientadas a proyectos **pequeños**
 - No renuncian a las prácticas esenciales para asegurar la **calidad del producto**

29-ago.-25 Ingeniería del Software página 6

6




Origen de las Metodologías Ágiles

- » Origen
 - En marzo de 2001, 17 expertos de la industria del software críticos de las metodologías formales, convocados por Kent Beck (que acababa de definir una nueva metodología denominada *Extreme Programming*), se reunieron en Salt Lake City (Utah, EEUU) para discutir sobre los modelos de desarrollo de software
 - En la reunión se acuñó el término “**Métodos Ágiles**” para definir a aquellos que estaban surgiendo como alternativa a las metodologías formales (CMM-SW, PMI, SPICE), a las que consideraban excesivamente “pesadas” y rígidas por su carácter normativo y fuerte dependencia de planificaciones detalladas, previas al desarrollo
- » Se creó **The Agile Alliance** (www.agilealliance.org)

29-ago.-25 Ingeniería del Software página 7

7



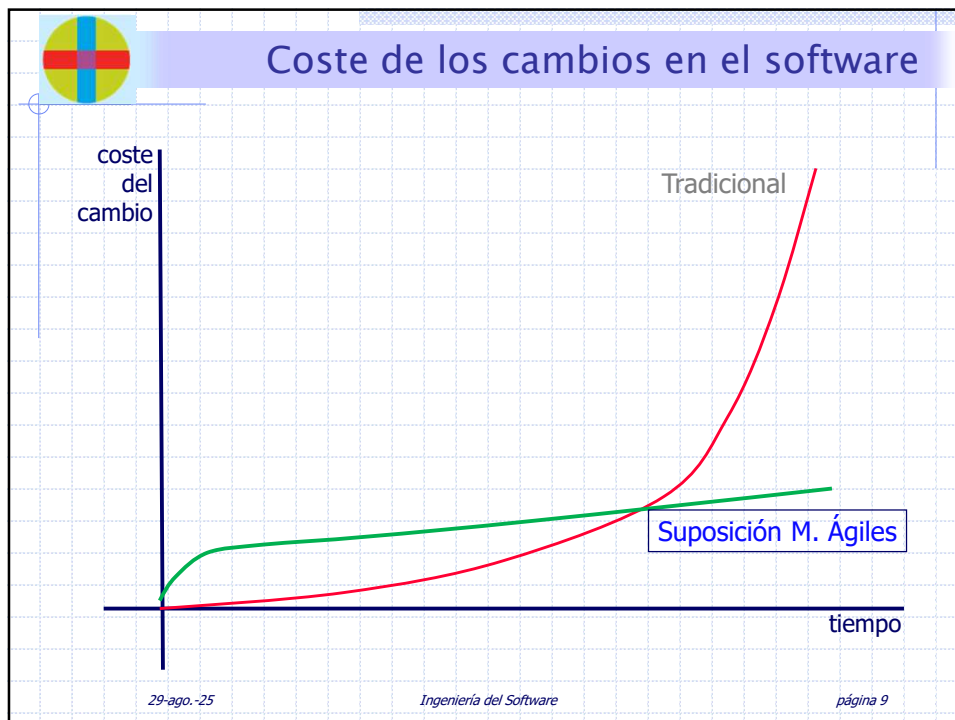
¿Qué valora una Metodología Ágil?

- » Las Metodologías Ágiles valoran:
 - Al **individuo** y las **interacciones** en el equipo de desarrollo más que a las actividades, procesos y herramientas
 - **Desarrollar software que funciona** más que conseguir documentación extensiva ⇒ minimalismo respecto del modelado y la documentación del sistema
 - La **colaboración** entre el cliente y el equipo de desarrollo, más que la negociación de un contrato
 - **Responder a los cambios** más que seguir estrictamente una planificación

www.agilealliance.org

29-ago.-25 Ingeniería del Software página 8

8



9

Manifiesto de las AMs (I)

» Principios:

1. La prioridad esencial es **satisfacer al cliente** mediante tempranas y continuas **entregas de software** que le reporte un **valor**
2. Dar la **bienvenida a los cambios**. Incluso en la parte final del desarrollo. Las AMs capturan los cambios para que el cliente tenga una ventaja competitiva
3. **Entregar frecuentemente** software **que funcione**, desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas
4. La gente del negocio y los desarrolladores deben **trabajar juntos** diariamente a lo largo del proyecto
5. Construir proyecto en torno a **individuos motivados**. Darles el entorno y el apoyo que necesitan y **confiar en ellos** para conseguir el trabajo
6. El **diálogo cara a cara** es el método más eficaz para comunicar información en un equipo de desarrollo

29-ago.-25 Ingeniería del Software página 10

10



Manifiesto de las AMs (II)

» Principios:

7. El **software que funciona** es la medida principal de **progreso**

8. Los procesos ágiles promueven un **desarrollo perdurable**. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener un ritmo constante

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

10. La **simplicidad** (el arte de maximizar la cantidad de trabajo no realizado) es esencial

11. Las mejores arquitecturas, requisitos y diseños surgen de los **equipos autogestionados**


12. A intervalos regulares, el equipo **reflexiona** sobre cómo ser más eficaz, y según esto **ajusta** su comportamiento

<http://agilemanifesto.org/principles.html>

29-ago.-25

Ingeniería del Software

página 11



Metodologías ágiles y tradicionales

Metodologías Ágiles

- Basadas en la práctica
- Preparadas para cambios
- Impuestas internamente
- No existe un contrato tradicional o al menos es bastante flexible
- El cliente es parte del equipo de desarrollo (in-situ)
- Grupos pequeños (menos de 10 integrantes) y trabajando en el mismo sitio
- Menos énfasis en la arquitectura
- Pocos artefactos
- Pocos roles

Metodologías Tradicionales

- Basadas en estándares
- Resistencia al cambio
- Impuestas externamente
- Contrato prefijado
- El cliente interactúa con el equipo de desarrollo mediante reuniones
- Grupos grandes y distribuidos
- La arquitectura es esencial
- Más artefactos
- Más roles

29-ago.-25

Ingeniería del Software

página 12



Principales Metodologías Ágiles


- **Extreme Programming** (Kent Beck)
www.extremeprogramming.org
- **SCRUM** (Ken Schwaber, Jeff Sutherland)
www.scrum.org
- **Crystal Methodologies** (Alistair Cockburn)
alistair.cockburn.us/Crystal+methodologies
heartofagile.com
- **DSDM** (Dynamic Systems Development Method)
<https://www.agilebusiness.org>
- **Adaptive Software Development** (Jim Highsmith)
www.adaptivesd.com
- **Lean Programming** (Mary Poppendieck)
www.poppendieck.com
- **Feature-Driven Development** (Peter Coad, Jeff De Luca)
www.nebulon.com/fdd

29-ago.-25 Ingeniería del Software página 13


13



14



¿Preguntas?



29-ago.-25

Ingeniería del Software

página 15