

---

# Tema 2. Principios y Prácticas Ágiles





# Problemas Métodos Tradicionales

## Resumen

- ✗ No son buenos para entornos volátiles en los que los requisitos no son conocidos con exactitud (Incertidumbre y cambio).
- ✗ El cliente tiene que tomar muchas decisiones al principio (Costosas de cambiar y que no entiende).
- ✗ Los clientes piden más flexibilidad como solución a tomar decisiones pronto (Aumento de la complejidad del producto).
- ✗ El cliente toma decisiones con la información de las documentaciones técnicas.



# ¿Qué es lo importante?

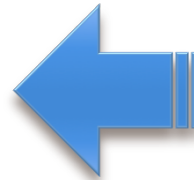


Producto



Excelencia Técnica  
O  
Satisfacción del cliente

Acabar en tiempo y  
costo  
O  
Maximizar el valor  
aportado al cliente



Proceso de desarrollo



# El cambio es inevitable

## Cambios durante el desarrollo:

- Incertidumbre (en requisitos, en tecnología, etc.).
- Modificaciones al ver las primeras soluciones.

## Cambios después de entregado el producto:

- Mantenimiento.
- Modificación y ampliaciones por cambios en el negocio.



# El cambio es inevitable

## Desarrollo Ideal:

- Los clientes saben lo que quieren y lo que necesitan.
- El equipo sabe cómo construirlo.
- Nada cambiará en el desarrollo.
- No hay restricciones (Tiempo y dinero para hacerlo).

## Desarrollo Real:

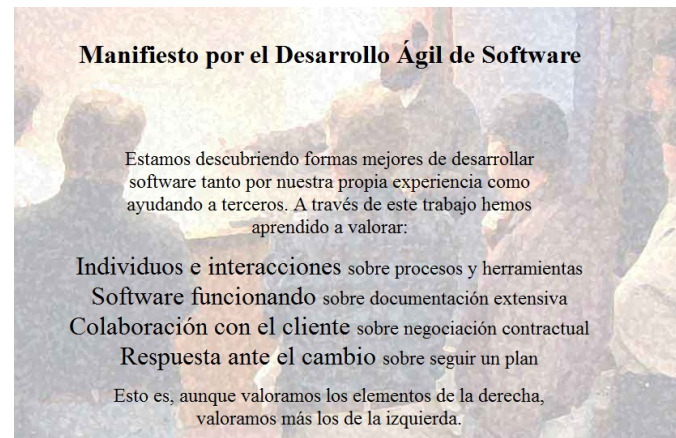
- Los clientes descubren lo que necesitan.
- Los desarrolladores descubren cómo hacerlo.
- Muchas cosas cambian en el camino.
- Siempre hay más cosas que hacer que tiempo y dinero disponible.



# El Manifiesto Ágil

¿Por qué tantos proyectos de desarrollo de software no se terminan a tiempo, cuestan más de lo presupuestado originalmente, tienen problemas de calidad serios y generan menor valor al negocio que el esperado?

Se preguntaron 17 expertos, que después elaboraron el **Manifiesto Ágil** Snowbird, Utah (USA), Feb 2001.



<http://agilemanifesto.org/iso/es/>



# Autores del Manifiesto Ágil

- × Kent Beck (XP), Mike Beedle, Arie van Bennekum (DSDM), Alistair Cockburn (Crystal), Ward Cunningham (XP), Martin Fowler (XP), James Grenning (XP), Jim Highsmith (ASD), Andrew Hunt (Pragmatic Programming), Ron Jeffries (XP), Jon Kern (FDD), Brian Marick, Robert C. Martin (XP), Steve Mellor, Ken Schwaber (Scrum), Jeff Sutherland (Scrum) y Dave Thomas (Pragmatic Programming)

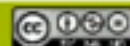


# Manifiesto Ágil

Principal **intención** del manifiesto Ágil:

- × El movimiento Ágil no es ir en contra de las metodologías (no es un “**anti-método**”).
- × El **modelado** y la **documentación** es importante pero no para generar recursos inútiles o poco usados.
- × Es importante **planificar** y **estimar** pero hay que reconocer los problemas y las necesidades de gestión en “entornos turbulentos”.







# Manifiesto Ágil

## Manifiesto por el Desarrollo Ágil de Software

Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar:

Individuos e interacciones sobre procesos y herramientas  
Software funcionando sobre documentación extensiva  
Colaboración con el cliente sobre negociación contractual  
Respuesta ante el cambio sobre seguir un plan

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.

<http://agilemanifesto.org/iso/es/>



# Principios del Manifiesto Ágil

1. Nuestra mayor prioridad es **satisfacer al cliente** mediante la entrega temprana y continua de software con **valor**.
2. Aceptamos que los **requisitos cambien**, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar **ventaja competitiva** al cliente.
3. Entregamos **software funcional frecuentemente**, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
4. Los **responsables de negocio** y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
5. Los proyectos se desarrollan en torno a **individuos motivados**. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.



# 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**.
7. El **software funcionando** es la medida principal de progreso.
8. Los procesos Ágiles promueven el **desarrollo sostenible**. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un **ritmo constante** de forma indefinida.
9. La atención continua a la **excelencia técnica** y al **buen diseño** mejora la Agilidad.
10. La **simplicidad**, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de **equipos auto-organizados**.
12. A intervalos regulares el equipo **reflexiona** sobre cómo ser más efectivo para a continuación **ajustar y perfeccionar** su comportamiento en consecuencia.



# ¿Qué es la Agilidad?

Según un **diccionario de lengua española**:

“Habilidad de cambiar rápida y efectivamente la dirección de un movimiento ejecutado a velocidad”.

“Conjunto de **mejores prácticas** que permite la entrega rápida de software de alta calidad y un enfoque de negocio que alinea el desarrollo con las necesidades y los objetivos de las compañías”

- ✓ La insistencia en el **cambio** es el conductor primordial hacia la agilidad.
- ✓ Un equipo ágil es un equipo rápido que responde de manera apropiada a los cambios.
- ✓ Cambios en el **software** que se va a construir, cambios entre los **miembros del equipo**, cambios por las **nuevas tecnologías**, cambios de todo tipo que puedan influir en el **producto** o en el **proyecto** que crea el producto.



# ¿Qué necesitamos?

## **AGILIDAD**

**PERSONAS**

**Trabajo en equipo**

**Priorizar por valor**

**Aprendizaje**

**Auto-Organización**

**Flexibilidad**

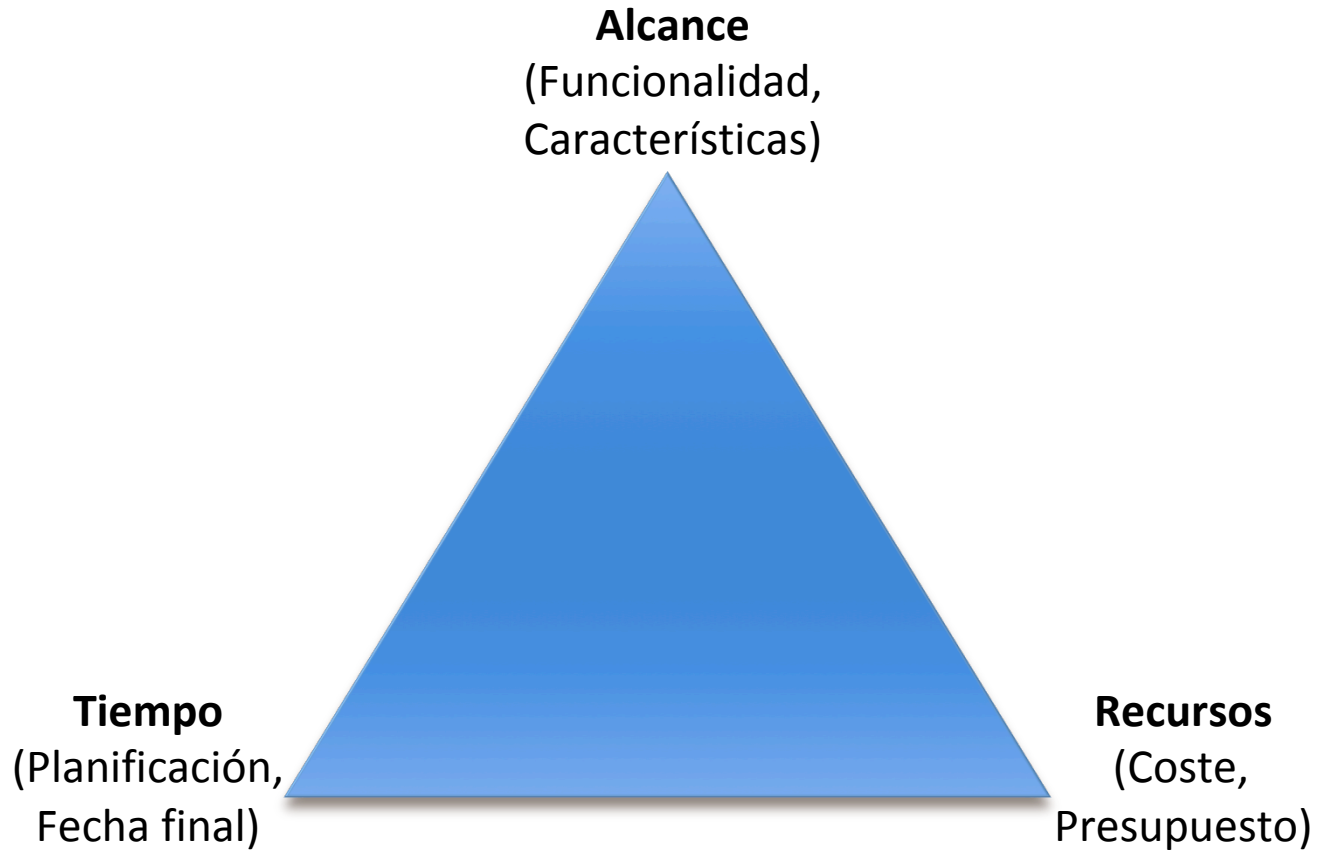
**Excelencia técnica**

**Comunicación cara a cara**



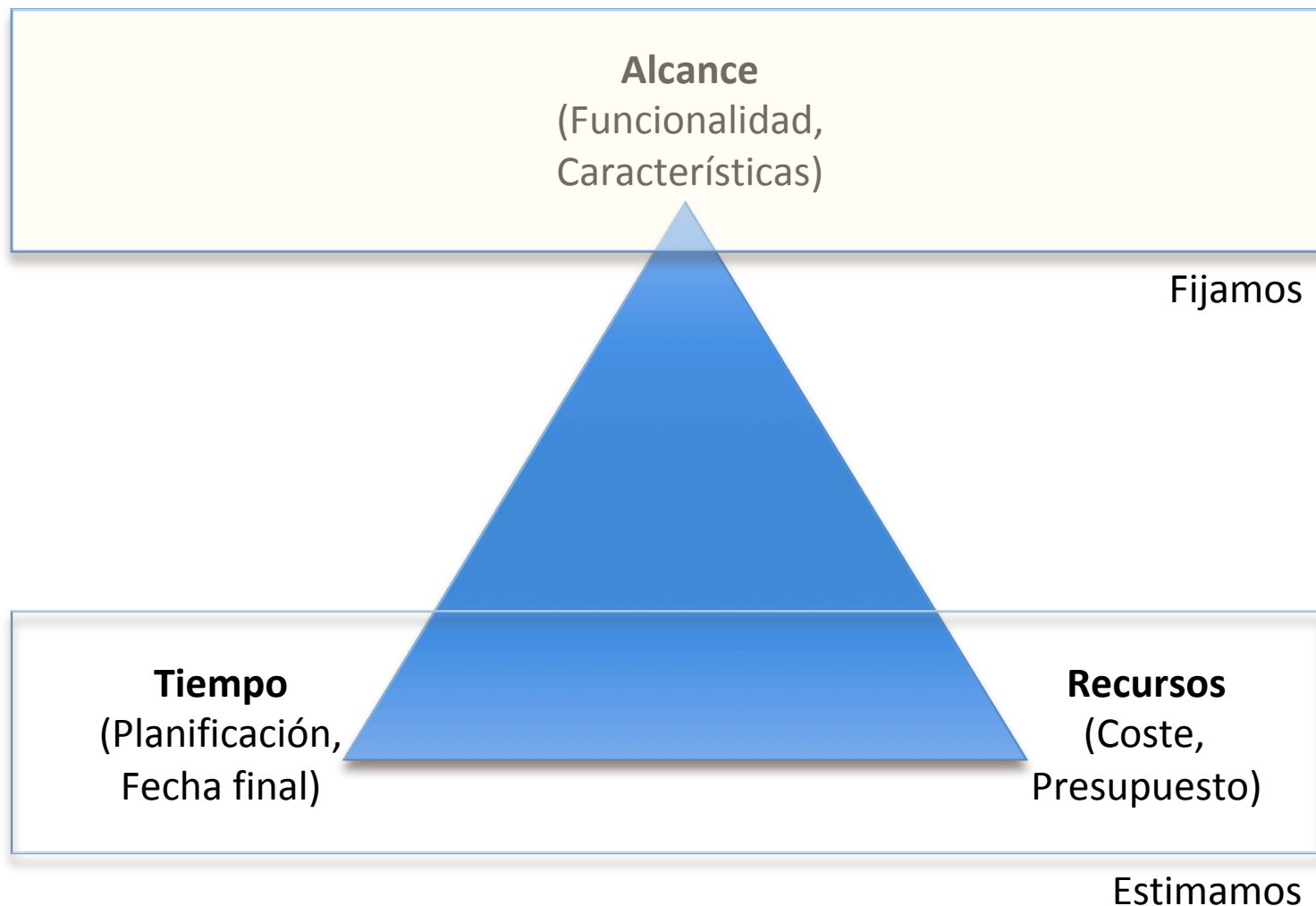


# Gestión de un proyecto





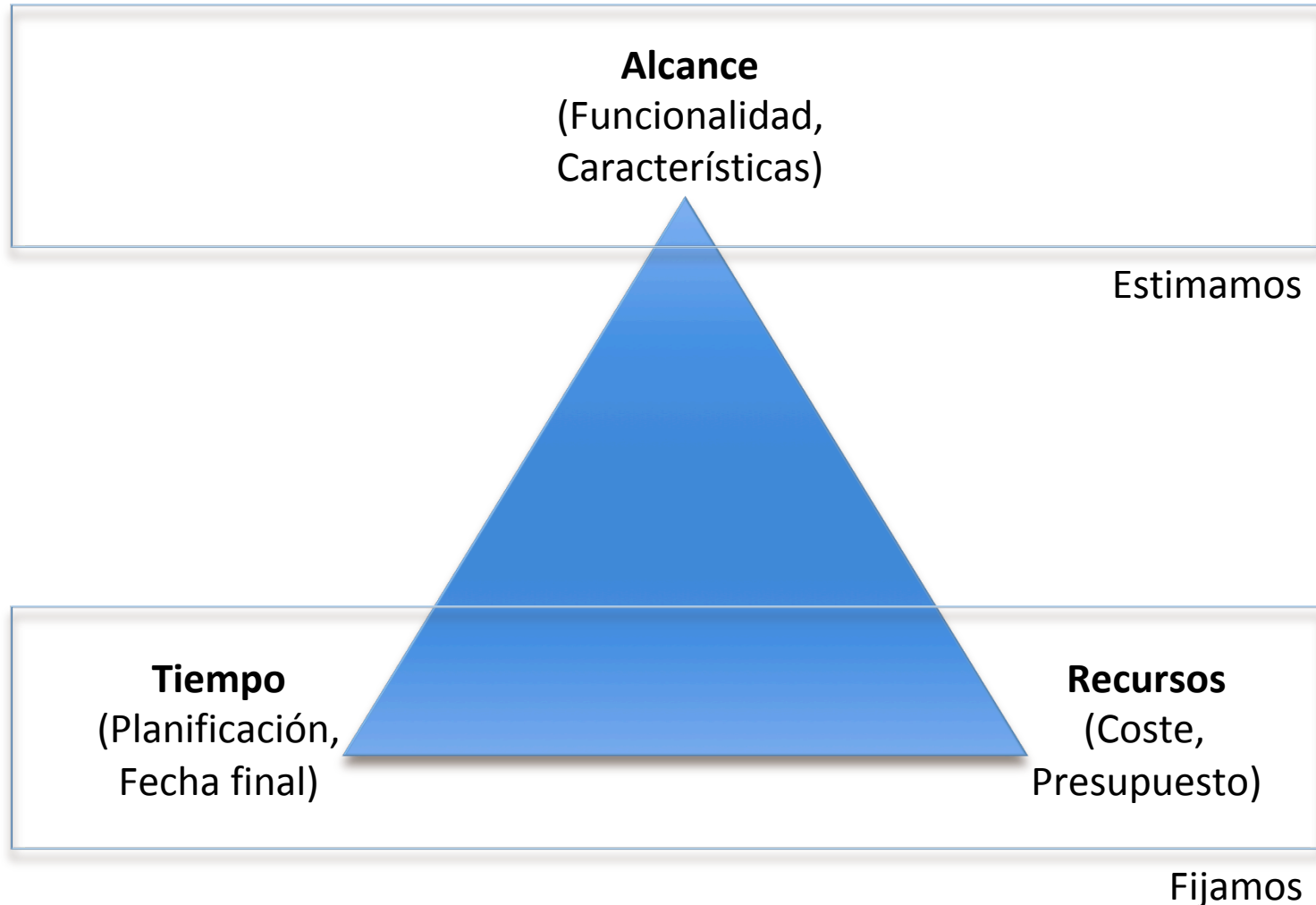
# Dirigido por “un plan” (predictivo)







# Dirigido por “el valor”





# Contrato con el cliente

- × Acuerdo que establecemos con el cliente sobre el servicio que vamos a proporcionar (ganar-ganar).
- × Un buen contrato aumenta las probabilidades de éxito.
- × Los riesgos deberían ser compartidos.
- × “Colaboración con el cliente frente a negociación contractual”.



# Contrato con el cliente

- × Fijar todo.
- × Fijar algunas cosas + Colaboración.
- × Fijamos cosas de forma progresiva.
- × Contratamos para la visión + Otro contrato.
- × Cualquier funcionalidad puede cambiarse por otra de igual peso (...yo quito tu eliges).
- × El cliente puede cancelar el proyecto si considera que no necesita más (pagando el 20% de lo cancelado).



# Contrato con el cliente

El dijo...	Ella dijo...
El sistema no funciona	El cliente cambió de idea
No podemos usarlo	Los usuarios no están formados
Nos deberían haber avisado	No nos hubieran escuchado
El sistema está lleno de bugs	Los datos estaban corruptos
Mal asesoramiento	Malas decisiones
Desarrolladores no cualificados	Interlocutores inadecuados
Mal proceso de desarrollo	Cliente no involucrado
El sistema no funcionó en campo	El cliente no adaptó sus procesos
Entregaron tarde	Añadieron cambios



2012 Proyectalis Gestión de Proyectos S.L.

<http://www.proyectosagiles.org/como-cocinar-tu-contrato-agil>

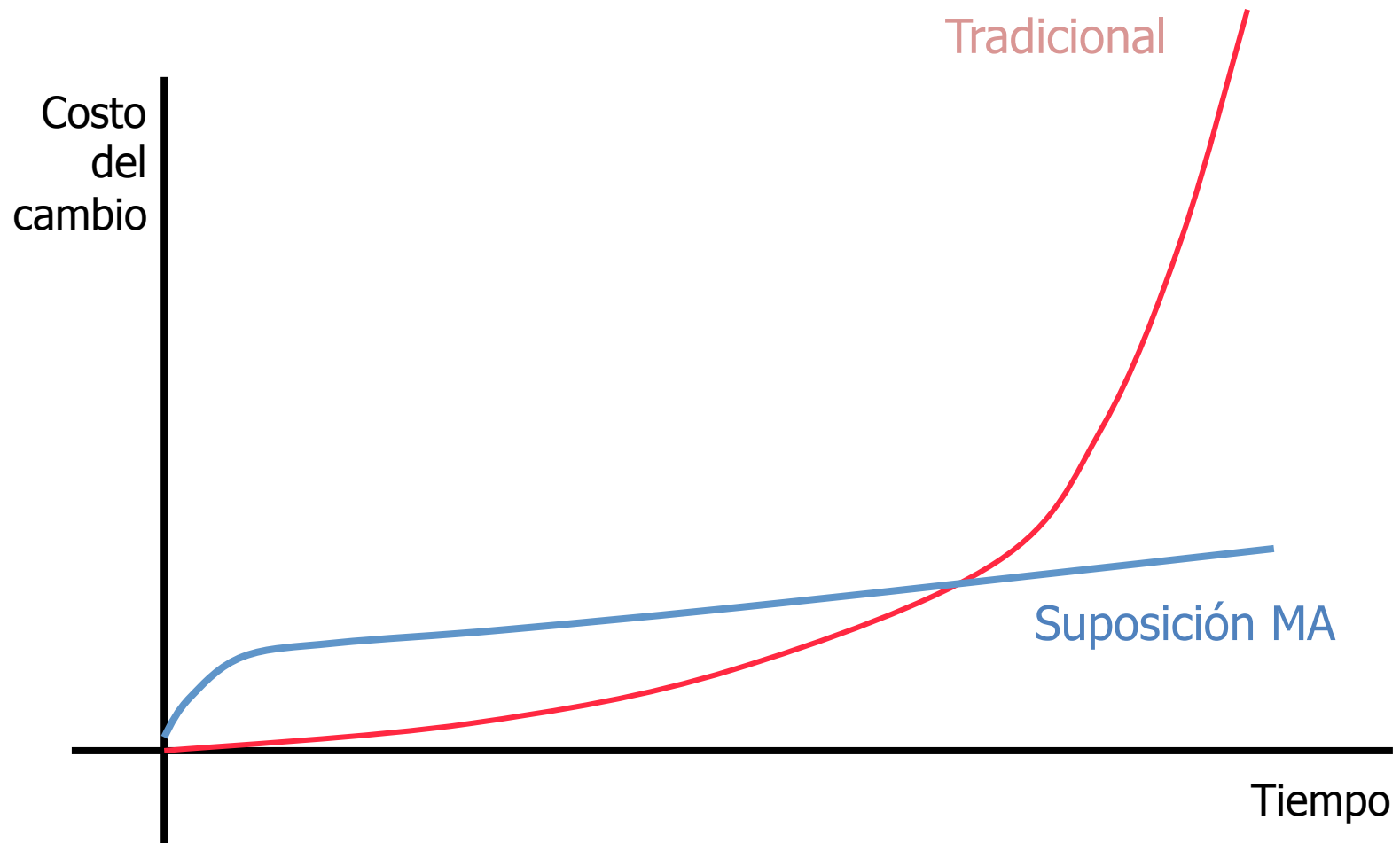


# Seguimiento del proyecto

- × Control y seguimiento del proyecto basado en objetivos.
- × Los requisitos se desarrollaran priorizados por el valor aportado al cliente.
- × El control y seguimiento del proyecto se basará en los requisitos completados.



# Importancia del “Coste del cambio”





# Valor real de los productos

- ✗ Dedicamos mucho tiempo a desarrollar objetivos que aportan poco valor.
- ✗ ¿Quién conoce el valor real de una funcionalidad?

De las funcionalidades desarrolladas:

- 7% se usan siempre.
- 13% se usan a menudo.
- 19% se usan pocas veces.
- 45% no se usan **nunca**.



# Equipos Ágiles

- × **Competencia**. Habilidades específicas sobre el software y el proceso a seguir.
- × **Enfoque común**. Una sola meta. “Entregar al cliente un incremento de valor del producto software”.
- × **Colaboración**. Comunicación y Trabajo en equipo.
- × **Toma de decisiones**. Autonomía y Autoridad.
- × **Resolución de problemas confusos**. Adaptación.
- × **Confianza y respeto**.
- × **Auto organización**.





# ¿Cuándo aplicar agilidad?

- × Cuando tenemos un equipo pequeño y localizado en el mismo sitio.
- × Cuando los requisitos son dinámicos.
- × No tenemos un sistema crítico.
- × Los clientes tienen disponibilidad para el proyecto y quieren colaborar.
- × La empresa está abierta a nuevas formas de desarrollo de software.