

Metodologías Ágiles

¿Qué es una Metodología Ágil?

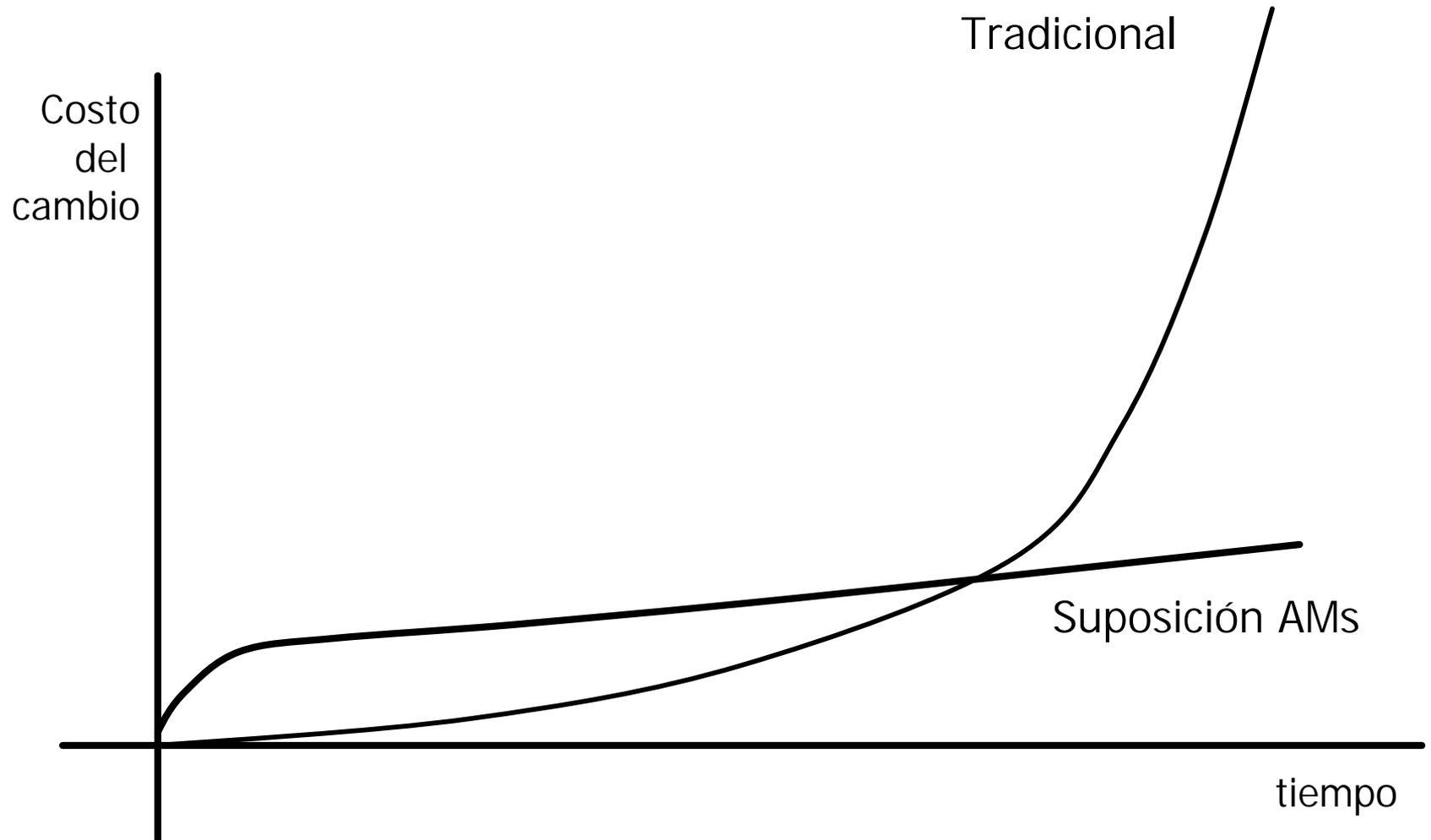
www.agilealliance.com

- Las Metodologías Ágiles (AMs) valoran:
 - Al individuo y las interacciones en el equipo de desarrollo más que a las actividades y las herramientas
 - Desarrollar software que funciona más que conseguir una buena documentación ⇒ Minimalismo respecto del modelado y la documentación del sistema
 - La colaboración con el cliente más que la negociación de un contrato
 - Responder a los cambios más que seguir estrictamente una planificación

¿Por qué surgen las Metodologías Ágiles (AMs)?

- Dificultad para implantar metodologías tradicionales. Sofisticadas herramientas CASE y notaciones (UML)
- Una solución a medida para un segmento importante de proyectos de desarrollo de software
- Pugna entre comunidades/gurús
- “Aceptar el cambio” ...
- Gestión del Conocimiento

Costo de los Cambios en SW



Manifiesto de las AMs

agilemanifesto.org

Principios:

1. La prioridad principal es satisfacer al cliente mediante tempranas y continuas entregas de software que le reporte un valor
2. Dar la bienvenida a los cambios. Los 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 una entrega y la siguiente

... Manifiesto de las AMs

4. La gente del negocio y los desarrolladores deben trabajar juntos 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 eficiente y efectivo para comunicar información dentro de un equipo de desarrollo
7. El software que funciona es la medida principal de progreso

... Manifiesto de las AMs

8. Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante
9. La atención continua a la calidad técnica y al buen diseño mejora la agilidad
10. La simplicidad es esencial
11. Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos
12. En intervalos regulares, el equipo reflexiona respecto de cómo llegar a ser más efectivo, y según esto ajusta su comportamiento

Comparación

Metodología Ágil	Metodología No Ágil
Pocos Artefactos	Más Artefactos
Pocos Roles	Más Roles
No existe un contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
Cliente es parte del equipo de desarrollo (además in-situ)	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (< 10 integrantes) y trabajando en el mismo sitio	Grupos grandes
Menos énfasis en la arquitectura	La arquitectura es esencial

Limitaciones

- Proporcionan una ayuda limitada en equipos de trabajo dispersos físicamente
- Proporcionan una ayuda limitada en equipos de trabajo grandes
- Consideran una ayuda limitada al tratamiento de subcontratos
- No privilegian la reutilización de componentes
- Proporcionan una ayuda limitada para desarrollar software de seguridad crítica
- Proporcionan ayuda limitada para desarrollar software grande y complejo
- Dificultad en la utilización de herramientas que apoyen el desarrollo

Tipos de Proyectos

Tradicionales

- Grandes
- Con requerimientos estables
- Aplicaciones críticas
- Grandes equipos de desarrollo
- Equipo de desarrollo distribuídos geográficamente

Agiles

- Ambientes dinámicos, con equipos de trabajo pequeños y produciendo aplicaciones no críticas
- Requerimientos desconocidos o inestables, garantizando un menor riesgo ante la posibilidad de cambio en los requerimientos

Principales AMs

- Crystal Methodologies, Alistair Cockburn, www.crystallmethodologies.org
- SCRUM, Ken Schwaber & Jeff Sutherland, www.controlchaos.com
- DSDM (Dynamic Systems Development Method), www.dsdm.org
- Lean Programming, Mary Poppendieck, www.poppendieck.com
- FDD (Feature-Driven Development), Peter Coad & Jeff De Luca, www.nebulon.com/fdd, www.coad.com/peter/#fdd
- Extreme Programming, Kent Beck www.extremeprogramming.org, www.xprogramming.com
- Adaptative Software Development, Jim Highsmith www.adaptivesd.com

¿Qué resultado proveen las Metodologías Ágiles?

- Hay pocos datos concretos del índice de éxito de proyectos
- Está teniendo un gran auge
 - Aumento en el número de proyectos
 - ¿Por qué?
 - Tiene el apoyo de muchos gurús en ingeniería de sw
 - Es un proceso para gente que odia los procesos
 - Tiene sentido
 - ¿Política? ... Pugna entre comunidades

¿Cuándo utilizar una Metodología Ágil?

- ¿Existe ya un proceso? Si
- ¿Reacciona bien a los cambios? Si
- ¿Está el equipo contento con él? Si

⇒ Mejor esperar

- Se están recogiendo datos
- En un futuro se podrán hacer comparaciones sobre lo que es más conveniente

... ¿Cuándo utilizar una Metodología Ágil?

- ¿Existe ya un proceso? No
 - o existe pero no reacciona bien a los cambios
 - o existe pero el equipo no está contento con él

⇒ Una Metodología Ágil puede ser una buena forma de empezar

- Fácil de financiar
- A los programadores les gusta
- A los clientes les gusta el mayor control

Conclusiones

- Las Metodologías Ágiles surgen como respuesta a problemas reales
- Las metodologías tradicionales no son totalmente adecuadas para todos los desarrollos software, principalmente por su falta de flexibilidad
- Los métodos ágiles y los tradicionales no son competidores directos. Cada uno de ellos tiene su propio segmento de aplicación o terreno
- Algunos aspectos del desarrollo de software se beneficiarán del enfoque agilista mientras otros obtendrán beneficios de un enfoque tradicional-predictivo menos ágil
- Lo importante es saber determinar el tipo de proceso y herramientas que mejor sirvan a cada proyecto y organización