

Clase 1

Metodologías ágiles



Concepto

Desarrollo ágil de software es un **conjunto de métodos** en el que las necesidades y soluciones **evolucionan** a través de la colaboración entre **equipos multifunción** y **auto-organizados**.



Marco

Promueve la **planificación adaptativa**, **desarrollo evolutivo**, **entrega temprana** y **mejora continua**, promoviendo una respuesta **rápida** y **flexible** a los cambios.

Es un marco conceptual que se centra en la distribución de software de trabajo con la mínima cantidad de trabajo.

Manifiesto

El **Manifiesto Ágil** introdujo el término en 2001. Desde entonces, el **Movimiento Ágil**, con todos sus valores, principios, métodos, prácticas, herramientas, defensores y profesionales, filosofías y culturas, ha cambiado considerablemente el panorama del desarrollo comercial e ingeniería moderna de software en la era de Internet.

1. Historia

Los métodos de desarrollo de **software incremental** se remontan a **1957** en Los Ángeles, bajo la dirección de **Bernard Dimsdale** en el Service Bureau Corporation de **IBM**.



1. Historia

En **1974**, un artículo de **E. Edmonds** introdujo un proceso de desarrollo de software **adaptativo**.

Al mismo tiempo y de forma independiente los mismos métodos fueron desarrollados e implementados por el Centro de Desarrollo de Sistemas de la Compañía Telefónica de Nueva York bajo la dirección de **Dan Gielan**.

1. Historia

Por su parte, **Tom Gilb** comenzó a publicar los conceptos de **Gestión de Proyectos Evolutivos** (EVO por las siglas en inglés de Evolutionary Project Management).

A fines de la década de **1970** Gielan disertó ampliamente en todo Estados Unidos sobre esta metodología, sus prácticas y sus beneficios.



1. Historia

Los llamados **métodos livianos** de desarrollo **ágil** de software evolucionaron a mediados de la década de **1990** como una reacción contra de los **métodos pesados** de **cascada**, criticados por ser fuertemente regulados y reglamentados.



1. Historia

Primeras implementaciones de métodos ágiles:

- 1994: **RUP** (Rational Unified Process)
- 1995: **Scrum**
DSDM (Dynamic Systems Dev. Method)
- 1996: **Crystal Clear**
XP (Extreme Programming),
- 1997: **ASD** (Adaptive Software Development)
FDD (Feature Driven Development)

1. Historia

Estas y otras implementaciones ahora se conocen colectivamente como "**metodologías ágiles**", después del **Manifiesto Ágil** publicado en **2001**.



2. Manifiesto Ágil

En 2001, 17 desarrolladores de software se reunieron en un hotel de Utah, Estados Unidos, para discutir los métodos ágiles de desarrollo.



2. Manifiesto Ágil

Como resultado publicaron el "**Manifiesto para el desarrollo ágil de software**".



Algunos de los autores formaron la **Alianza Ágil**, una organización sin fines de lucro que promueve el desarrollo de software de acuerdo a los valores y principios del manifiesto.

2. Manifiesto Ágil

Doce principios:

1. Satisfacción del cliente mediante la rápida entrega de software útil.
2. Los requisitos cambiantes son bienvenidos, incluso si llegan tarde al desarrollo.
3. Entregar con frecuencia software que funcione (semanas y no meses).



2. Manifiesto Ágil

Doce principios:

4. Cooperación diaria y cercana entre las personas del negocio y los desarrolladores.
5. Construcción de proyectos en torno a individuos motivados en quienes se debe confiar.
6. La conversación cara a cara es la mejor forma de comunicación.

2. Manifiesto Ágil

Doce principios:

- 7. El software funcionando es la principal medida de avance.
- 8. Desarrollo sostenido, capaz de mantener un ritmo constante.
- 9. Atención continua a la excelencia técnica y el buen diseño.



2. Manifiesto Ágil

Doce principios:

10. La simplicidad *-el arte de maximizar la cantidad de trabajo que no se hace-* es esencial.
11. Equipos auto-organizados.
12. Frecuente adaptación a circunstancias de cambio.

2. Manifiesto Ágil

La utilización de todas las buenas prácticas enumeradas en el manifiesto ágil no implica ser ágil, pero incumplir alguna de ellas significa no ser ágil.



2. Manifiesto Ágil

Se tienen en cuenta estos puntos, dándole más importancia a la primera parte que a la segunda:

1. Se valora a los **individuos** y las **interacciones** sobre los procesos y las herramientas.
2. Se valoran las **aplicaciones que funcionan** sobre la documentación exhaustiva.
3. Se valora la **colaboración del cliente** sobre las negociaciones contractuales.
4. Se valora la **respuesta al cambio** sobre el seguimiento de un plan.

3. Características

Hay muchos métodos de desarrollo ágil. La mayoría promueven el trabajo en equipo, la colaboración y la adaptabilidad a lo largo del ciclo de vida del proyecto:

- Iterativo, incremental y evolutivo;
- Comunicación eficiente y cara a cara;
- Circuito de retroalimentación y ciclo de adaptación muy cortos;
- Foco en la calidad.

3. Características

3.1. Iterativo, incremental y evolutivo.

La mayoría de los métodos ágiles dividen las tareas en pequeños incrementos con una planificación mínima y no involucran directamente a la planificación a largo plazo. Las iteraciones son períodos de tiempo cortos o

“timeboxes” que por lo general duran de una a cuatro semanas.



3. Características

3.1. Iterativo, incremental y evolutivo.

En cada iteración un equipo multidisciplinario trabaja en todas las funciones: planificación, análisis de requerimientos, diseño, codificación, pruebas unitarias y pruebas de aceptación. Al final de la iteración, el producto se muestra a las partes interesadas. Esto minimiza el riesgo global y permite que el proyecto se adapte rápidamente a los cambios.

3. Características

3.1. Iterativo, incremental y evolutivo.

Una iteración podría no agregar suficiente funcionalidad como para justificar una implementación, pero el objetivo es tener una versión disponible (con errores mínimos) al final de cada iteración.

Múltiples iteraciones podrían ser necesarias para lanzar un producto o nuevas funcionalidades.

3. Características

3.2. Comunicación eficiente y cara a cara.

Sin importar la disciplina de desarrollo requerida, un equipo ágil contendrá un representante del cliente.

Esta persona es nombrada por las partes interesadas y se compromete a estar disponible en cada iteración para atender las consultas de los desarrolladores.



3. Características

3.3. Circuito de retroalimentación y ciclo de adaptación muy cortos.

Una característica común de los métodos de desarrollo ágil son las reuniones de estado diario o "*stand-ups*".

En una breve sesión, los miembros del equipo reportan lo que hicieron el día anterior, lo que van a hacer hoy, y cuáles son sus obstáculos.

4. Filosofía

En comparación con la ingeniería de software tradicional, el desarrollo ágil se dirige principalmente a los sistemas complejos y proyectos con características dinámicas, donde las estimaciones precisas, los planes estables y las predicciones son a menudo difíciles de conseguir en las primeras etapas, y grandes diseños y acuerdos iniciales probablemente causarán una gran cantidad de desperdicio, es decir, no son económicamente viables.

4. Filosofía

Estos argumentos básicos y experiencia adquirida por años de éxitos y fracasos han ayudado a darle a las metodologías ágiles sus características **adaptativas**, **iterativas** y **evolutivas**:



Adaptativo vs. **Predictivo**
Iterativo vs. **Cascada**
Código vs. **Documentación**



4. Filosofía

4.1. Adaptativo vs. Predictivo.

Los métodos de desarrollo van desde lo **adaptativo** hasta lo **predictivo**. Los métodos ágiles se encuentran del lado de los adaptativos. Los métodos adaptativos se centran en adaptarse rápidamente a las cambiantes realidades. Cuando las necesidades de un proyecto cambian, un equipo adaptativo cambia también.

4. Filosofía

4.1. Adaptativo vs. Predictivo.

Tendrá dificultades para describir exactamente lo que sucederá en el futuro: cuanto más lejos esté una fecha, más pobre será la predicción.



4. Filosofía

4.1. Adaptativo vs. Predictivo.

Un equipo adaptativo no podrá informar exactamente las tareas que va a hacer la semana próxima; y si se le pregunta sobre una implementación a realizarse dentro de seis meses, quizás sólo pueda responder con la declaración de objetivos.

4. Filosofía

4.1. Adaptativo vs. Predictivo.

Por el contrario, los métodos predictivos se centran en el análisis y la planificación del futuro y los riesgos conocidos.

Un equipo de predicción puede informar exactamente qué tareas tiene planeadas para cada momento de todo el tiempo de desarrollo...



4. Filosofía

4.1. Adaptativo vs. Predictivo.

...pero como se basa en el análisis de las primeras etapas, tendrá grandes dificultades para cambiar de dirección en caso de ser necesario.

Por eso suelen tener un comité de control de cambios para asegurar que sólo los más relevantes sean considerados.

4. Filosofía

4.2. Iterativo vs. Cascada.

Una de las diferencias entre **Ágil** y **Cascada** es que las pruebas se llevan a cabo en diferentes etapas del ciclo de vida de desarrollo. En **Cascada** siempre hay una fase de prueba por separado al final de la fase de implementación. En cambio, en las metodologías **Ágiles** las pruebas se realizan generalmente de manera concurrente con la codificación.

4. Filosofía

4.3. Código vs. Documentación.

Las metodologías ágiles proponen que los programadores deben escribir documentación si esa es la mejor manera de lograr los objetivos importantes.



4. Filosofía

4.3. Código vs. Documentación.

Scott Ambler sostiene que la documentación debe ser “**apenas suficientemente buena**” o JBGE (*“Just Barely Good Enough”*) debido a que:



4. Filosofía

4.3. Código vs. Documentación.

- a. Mucha documentación suele provocar desperdicio;
- b. Los desarrolladores rara vez confían en la documentación detallada porque generalmente no coincide con el código;
- c. Muy poca documentación también puede causar problemas para el mantenimiento, la comunicación y el aprendizaje.

5. Métodos ágiles

Algunos frameworks y métodos de desarrollo de software:

- Adaptive Software Development (ASD)
- Agile Modeling
- Agile Unified Process (AUP)
- Crystal Methods (Crystal Clear)
- Disciplined Agile Delivery
- Dynamic Systems Development Method (DSDM)

5. Métodos ágiles

Algunos frameworks y métodos de desarrollo de software:

- Extreme Programming (XP)
- Feature Driven Development
- Lean Software Development
- Kanban
- Scrum
- Scrum-ban



6. Prácticas ágiles

El desarrollo ágil se apoya en un conjunto de prácticas concretas sugeridas por los métodos ágiles, que cubren áreas como

requerimientos, diseño,
modelado, codificación,
pruebas, gestión de
proyectos, procesos y
calidad.



6. Prácticas ágiles

Algunas prácticas ágiles:

- Acceptance test-driven development (ATDD);
- Agile Modeling;
- Agile testing;
- Behavior-driven development (BDD);
- Continuous integration (CI);
- Cross-functional team;
- Domain-driven design (DDD);

6. Prácticas ágiles

Algunas prácticas ágiles:

- **Information radiators** (Scrum board, Kanban board, Task board, Burndown chart);
- **Iterative and incremental development** (IID);
- **Pair programming**;
- **Planning poker**;
- **Refactoring**;
- **Scrum meetings** (Sprint planning, Daily scrum, Sprint review y Sprint retrospective);

6. Prácticas ágiles

Algunas prácticas ágiles:

- Story-driven modeling;
- Test-driven development (TDD);
- Timeboxing;
- Use case;
- User story;
- Velocity tracking.



7. Medición de agilidad

Agility Index Measurements (AIM) asigna puntaje a los proyectos sobre una serie de factores de agilidad.

Agility Measurement Index (AMI) asigna puntaje a los desarrollos en cinco dimensiones: duración, riesgo, innovación, esfuerzo e interacción.



7. Medición de agilidad

Otras técnicas se basan en objetivos mensurables o la velocidad del proyecto.

Existen autoevaluaciones para determinar si un equipo está utilizando prácticas ágiles, como las pruebas Nokia o Karlskrona.

La aplicación práctica de estas mediciones es objeto de debate.

8. Resultados

Uno de los primeros informes sobre mejoras en la calidad, la productividad y la satisfacción de las empresas mediante el uso de métodos ágiles, fue una encuesta realizada por la consultora australiana [Shine Technologies](#) en 2003. [IBM Rational](#) informó mejoras similares en 2006.

Otros dicen que los métodos ágiles son aún muy jóvenes como para medir su éxito.



8. Resultados

Pero las metodologías ágiles
también pueden ser **ineficientes**
en grandes organizaciones
y ciertos tipos de
proyectos.



8. Resultados

Hay organizaciones que creen que las metodologías ágiles son demasiado extremas y adoptan un enfoque híbrido que combina elementos de los enfoques ágiles y fuertemente planificados.

El término "ágil" también ha sido criticado por ser una “moda de gestión” que se limita a describir las buenas prácticas ya existentes, pero con una nueva jerga.