

Metodologías Ágiles.

¿Qué son las metodologías ágiles?

Las metodologías ágiles no son una simple herramienta, sino una estrategia integral que impulsa a gestionar proyectos con rapidez y flexibilidad. Estas metodologías se originaron en el ámbito del desarrollo de software, pero su aplicación se ha expandido a diversas áreas. A diferencia de la forma tradicional de gestionar proyectos, estas metodologías no necesitan definir la totalidad del alcance al principio del proyecto. Definimos a las metodologías ágiles como una innovadora forma de trabajar y organizarse que divide los proyectos en fragmentos capaces de adaptarse a medida que avanza, complementarse y resolverse en poco tiempo. En otras palabras, los proyectos no se planifican ni se diseñan; sino que, a medida que se va desarrollando, se va definiendo gracias a un feedback constante.

¿Cuándo nació el concepto “Agile”?

En el año 2001, debido a la insuficiencia de las metodologías tradicionales, los principales directivos de empresas de desarrollo de software se reunieron para dialogar sobre la necesidad de crear nuevas herramientas. En ese momento nacen oficialmente las metodologías ágiles.

Principios clave del Manifiesto Agile.

1. Individuos e interacciones sobre procesos y herramientas.
2. Software funcionando sobre documentación extensiva.
3. Colaboración con el cliente sobre negociación de contratos.
4. Responder al cambio sobre seguir un plan.

Metodologías ágiles concretas.

Agile Unified Process (AUP)

La metodología AUP (Agile Unified Process) es un enfoque ágil para el desarrollo de software que combina las mejores prácticas de varias metodologías ágiles.

Características:

1. Es una versión simplificada de la metodología RUP.
2. Abarca siete flujos de trabajo, cuatro ingenieriles y tres de apoyo: Modelado, Implementación, Prueba etc

3. El modelado agrupa los tres primeros flujos de RUP.
4. Dispone de cuatro fases: Creación, Elaboración, Construcción y Transición.

Ventajas:

1. **Mayor adaptabilidad y flexibilidad:** Una de las mayores ventajas de la AUP es la capacidad de adaptarse a una gran velocidad a los cambios en los requisitos del proyecto.
2. **Mejora de la productividad:** El enfoque iterativo permite la entrega de resultados tangibles en cada interacción.
3. **Mayor participación del cliente:** La AUP promueve la colaboración del cliente a lo largo del proyecto.
4. **Menor riesgo del fracaso:** Al ser interactiva e incremental, reduce el riesgo de fracaso.
5. **Mayor eficiencia en el uso de recursos:** Los procesos de la Aup ayudan a conseguir más eficiencia en los recursos.
6. **Mayor satisfacción:** Al ser el cliente un participante directo su satisfacción con el resultado acaba siendo mayor.

Desventajas:

1. **Mayor dependencia de la comunicación:** Requiere estar en constante comunicación con el equipo y el cliente.
2. **Mayor dificultad para proyectos más grandes:** Es más adecuada para proyectos medianos o pequeños.
3. **Riesgo en escaso enfoque a la documentación:** La documentación se pasa a segundo plano al tener que presentar un software funcional.
4. **Requiere un equipo de desarrollo altamente especializado:** Para poder presentar todos los software funcionales.
5. **Mayor dificultad para estimar tiempo y precio:** Ya que al cabo de la elaboración del proyecto el cliente puede ir cambiándolo.

Opinión personal:

Si se tiene un buen equipo, es una de las mejores metodologías ágiles, sin duda, debido a que puede estar en constante elaboración y nunca se sabe hasta qué punto puede llegar, ya que el cliente es quien decide.

Posibles problemas:

Al ser una metodología ágil que no se puede usar para proyectos largos, la clientela es más reducida, lo que puede llevarnos a una pérdida económica.

Feature-Driven Development (FDD):

El desarrollo basado en funciones (FDD por sus siglas en inglés) es una metodología de desarrollo de software centrada en el cliente conocida por iteraciones cortas y lanzamientos frecuentes.

Características

1. **Procesamiento de objetos de dominio:** Se elaboran diagramas de clases con el propósito de describir los objetos pertenecientes a un dominio y las conexiones entre ellos.
2. **Diseño en función:** Las características se presentan en iteraciones prolongadas, generalmente de una a dos semanas.
3. **Propiedad de categoría individual:** Cada clase o grupo de código se asigna a un único usuario.
4. **Equipos de funcionamiento:** A pesar de que una persona es responsable del rendimiento y la calidad de cada clase, una función puede involucrar a más de una clase, de modo que todos los integrantes del equipo de funciones contribuyen a las decisiones de diseño e implementación.
5. **Las inspecciones son:** Se llevan a cabo pruebas con el propósito de detectar defectos y asegurar la calidad más alta.

Ventajas:

1. **Diseño de comunicación y colaboración:** FDD impulsa la comunicación y la colaboración dentro del equipo de desarrollo.
2. **Mayor presencia y control:** Proporciona una mayor comprensión y control sobre el progreso del proyecto.
3. **Entrega constante de incrementos de software:** Se brinda la entrega continua de valor mediante incrementos constante de software.
4. **Supervisión rápida de los riesgos:** Asistencia a detectar y solucionar problemas de forma rápida.
5. **Diseño de código mejorado:** FDD se centra en mantener los altos estándares de calidad en el código.
6. **Responsabilidad ante cambios:** Se encuentra flexible y puede ajustarse a cambios y requisitos emergentes.

Desventajas:

1. **Consistencia en la planificación inicial:** La etapa inicial de planificación puede ser difícil y requerir una definición detallada de las funciones.
2. **Disponibilidad de roles específicos:** FDD está sujeta a ciertos cargos esenciales, como el de arquitecto jefe, lo cual puede ser un desafío si no se dispone de personal con la formación necesaria.
3. **No hay flexibilidad en comparación con otras metodologías ágiles:** A pesar de que FDD es ágil, puede ser más flexible que otras metodologías como Scrum, particularmente en entornos en los que los requisitos cambian con frecuencia.
4. **Enfatiza el desarrollo de funciones:** Este enfoque puede involucrar otros aspectos relevantes del proyecto, como la vivencia del usuario o la integración con otros sistemas.
5. **Se requiere un documento adecuado:** Para que FDD funcione de manera adecuada, es necesario mantener una documentación detallada y actualizada, lo cual puede ser de gran ayuda.

Opinión personal

FDD es una excelente opción para equipos que trabajan en proyectos grandes y complejos donde es crucial mantener un enfoque estructurado. Al dividir el trabajo en características pequeñas y manejables, se facilita el seguimiento del progreso y el control de calidad. Además, su enfoque centrado en las características específicas del sistema permite una alineación más clara entre los requisitos del cliente y el producto final.

Posibles problemas

El Desarrollo Basado en Funciones (FDD) puede enfrentar problemas como la complejidad en la planificación inicial, dependencia de roles específicos, menor flexibilidad comparado con otras metodologías ágiles, enfoque limitado en el desarrollo de funciones y la necesidad de mantener una documentación detallada y actualizada.