

Metodologías ágiles

Es un enfoque para la toma de decisiones en los proyectos de desarrollo de software, que hace referencia a métodos basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto. Estas metodologías proponen un conjunto de buenas prácticas

Dado que estas metodologías están apoyadas en los procesos empíricos y estos se basan en la idea de que el contexto de cada proyecto tiene sus particularidades y es poco extrapolable a otros proyectos se acordaron una serie de valores en lo que se llama “Manifiesto Ágil”.

Manifiesto ágil



- 1) Se valora más a las personas y las interacciones que a los procesos y herramientas
- 2) Se valora más el software funcionando que una documentación completa y detallada
- 3) Se valora más trabajar en forma conjunta con el cliente que las negociaciones y los contratos firmados
- 4) Se acepta que van a haber cambios en los requerimientos que aferrarnos a un plan

Para reforzar el cumplimiento de estos valores se definieron 12 principios

- 1 Nuestra mayor prioridad es **satisfacer al cliente.**
- 2 **Aceptar** que los requisitos **cambien.**
- 3 **Entregar** software funcional **frecuentemente.**
- 4 Los responsables de negocios, diseñadores y desarrolladores deben **trabajar juntos** día a día durante el proyecto.
- 5 Desarrollamos proyectos en torno a **individuos motivados.**
- 6 El método más eficiente de comunicar información es **conversaciones cara a cara.**
- 7 El **software funcionando** es la principal **medida de éxito.**
- 8 Los procesos ágiles promueven el **desarrollo sostenible.**
- 9 La **atención continua** a la **excelencia técnica** y al **buen diseño** mejor la Agilidad.
- 10 La **simplicidad** 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 y de acuerdo a esto **ajustan su comportamiento.**

- 1) Satisfacer al cliente es la mayor prioridad
 - 2) Se debe entregar software funcionando frecuentemente
- Estos dos principios nos dan la pauta de que el ciclo de vida debe ser iterativo
- 3) Se acepta que los cambios pueden suceder en cualquier momento aún en etapas tardías del proyecto
 - 4) Los técnicos y no técnicos deben trabajar conjuntamente. Se suma a un referente del negocio para que opine y defina cuales son las características más importantes que se tienen que tener en cuenta
 - 5) Hay que tener personal con habilidades, entrenamiento, pero sobre todo motivados
 - 6) La simplicidad es esencial
 - 7) Las comunicaciones se tienen que hacer cara a cara
 - 8) El software funcionando es la medida principal de éxito
 - 9) El desarrollo debe ser sostenible, es decir, que el equipo tome un ritmo de trabajo con el que se pueda comprometer con el cliente y cumplir ese compromiso
 - 10) Se debe tener un enfoque de mejora continua de la técnica y la calidad
 - 11) El equipo debe estar auto organizado y tener poder para tomar decisiones
 - 12) A intervalos regulares el equipo debe reflexionar en cómo puede mejorar

En esta ideología se habla de requerimientos ágiles porque tenemos un proyecto que se va a gestionar con un proceso empírico basado en algún framework ágil y la gestión de los requerimientos tiene que estar en sintonía con esta filosofía

Partimos aceptando que los requerimientos pueden cambiar en cualquier momento y también partimos aceptando que la construcción del producto correcto es un proceso iterativo que va a ir refinándose y evolucionando. De esta manera hacemos una aceptación de que no podemos definir el 100% del producto de antemano antes de empezar a trabajar y esperando que esa definición inicial se mantenga estable a lo largo del tiempo

Big Requirements Up Front

Es un enfoque que dice que el diseño debe estar completo y detallado antes de comenzar a trabajar lo que trae como consecuencia que tenemos una gran cantidad de requerimientos que son muy poco usados

Por tanto, en las metodologías ágiles aparece un artefacto llamado Product Backlog que es una lista de características priorizadas ubicando en la parte superior aquellas características que tienen prioridad más alta y abajo las de menor prioridad. Esta priorización la hace el cliente

Por tanto, se utiliza el enfoque just in time que va definiendo los requerimientos de a poco. Esto permite evitar tener requerimientos que no se usarán

Framework: Es un marco de trabajo o conjunto de buenas prácticas

Scrum

Es un marco de trabajo para el desarrollo de productos

Se compone de tres roles principales

- Product Owner o dueño del proceso: Es la persona que está del lado del cliente. Es comparable con el ingeniero de requerimientos. Se encarga de levantar toda la información para conocer de primera mano todas las necesidades del cliente

- Scrum master: Es un moderador, se lo puede tomar como el líder del equipo de trabajo, pero no da ordenes ni dice como deben hacerse las cosas. Lo que hace es moderar y ayudar al team developer a entender cuál es la necesidad del cliente.

- Team developer: Son personas altamente capacitadas para dar solución y construir lo que el cliente necesita. Pueden ser analistas, testers, desarrolladores, etc.

Ciclo de vida

El Product Owner define un artefacto, un documento que tiene la lista completa de funcionalidades o necesidades del cliente. Este artefacto se llama Producto Backlog. Se plasman todos los requerimientos e ideas que harán cumplimiento a la solicitud del cliente.

Estas necesidades son comunicadas al equipo de desarrollo y al scrum master. Esto se realiza en una reunión llamada Sprint Planning Meeting. En esta reunión se define como se va a dar solución a una primera fase del producto final

Como resultado de esta reunión obtenemos una lista de funcionalidades llamada Spring Backlog. Es el conjunto de requisitos que se deben construir de 1 a 4 semanas. Este tiempo se llama Sprint

El Sprint corresponde al proceso de desarrollo y construcción de las necesidades del cliente. Esta actividad se divide en un módulo funcional o en un producto incremental. En el Sprint intervienen el Scrum Master y el Team Developer

Una de las actividades son las Daily Scrums que son reuniones diarias que tienen el propósito de dar seguimiento diario a todas las actividades que tengamos dentro del Sprint. Se reúnen el Scrum Master y el Team Developer y se pregunta que se hizo ayer, que se está haciendo hoy, que se va a hacer mañana y que problemas se encontró. La idea es que esa reunión sea muy corta, no mayor a 15 min. y que diariamente se pueda tener un contexto global de cuál es el estado actual del Sprint. Esto facilita el seguimiento del proyecto y la toma de decisiones

El éxito del Sprint depende de todo el equipo de desarrollo. Se busca que todas las personas tengan asignaciones de las cuales sean responsables y que cuando terminen una asignación puedan ayudarle a otro compañero para poder dar cumplimiento al objetivo del Sprint y a los tiempos definidos

Al finalizar el Sprint se hace una nueva reunión donde pueden estar involucrados el Scrum master, el producto Owner y el equipo de desarrollo para verificar el cumplimiento de las metas o los objetivos del Sprint y se entrega el producto

Luego de la entrega del producto se hace una nueva reunión que se llama Retrospectiva del Sprint en la que se busca analizar cuáles fueron los resultados del Sprint para poder encontrar alguna problemática, falencias en el proceso o mejoras que se puedan aplicar al siguiente Sprint.

Luego se inicia otro Sprint tomando otras de las funcionalidades del Product Backlog para sacar nuevamente el Sprint Backlog e iniciar otra vez el proceso hasta tener un nuevo producto funcional.

Comparación entre metodologías ágiles y tradicionales

- Las metodologías ágiles tiene el objetivo de entregar valor al cliente mientras que las tradicionales tiene el objetivo de cumplir con un plan
- Las tradicionales buscan especificar todos los requerimientos antes de empezar a trabajar y las ágiles desarrollan y evolucionan pequeñas fracciones de requerimientos a través de iteraciones