

Trabajo No 1 Ayudantía Ingeniería en Software

Nicolás Páez Morgado

17 de octubre de 2013

Ayudante : Cristian Garrido L^AT_EX.

Introducción

Las metodologías ágiles han sido identificadas como una alternativa a los enfoques tradicionales para el desarrollo de software. Las metodologías tradicionales, conocidas también como de planificación pesadas u orientadas, sólo deben aplicarse en situaciones donde las necesidades y requisitos del sistema futuro son estables y predecibles. Sin embargo, en proyectos en los que hay muchos cambios en los requisitos, probablemente cambios que requieran rehacer el código no es una actividad que genere altas ganancias, los equipos deben ser pequeños, las fechas de entrega del software serán cortas y la rapidez del desarrollo es fundamental, no pueden haber tampoco requisitos estáticos.

Proceso de desarrollo de documentación orientada de software son factores de alguna manera limitante para los desarrolladores y muchas organizaciones no tienen los recursos o preferencia a los desarrollos tradicionales para la producción de software. Por esta razón, las organizaciones pequeñas optan por no usar ningún proceso. Esto puede llevar a consecuencias desastrosas para la calidad del producto final, además de impedir la entrega del software a tiempo y los no alcanzar los costes preestablecidos. En particular, el modelo secuencial o clásico se presenta como ejemplo de metodología tradicional.

Entre todas las metodologías ágiles que existen ha habido un creciente número de partidarios a la Programación Extrema (XP). Las metodologías ágiles surgen con la propuesta de aumentar el enfoque en las personas y no en los procesos de desarrollo. Además, existe la preocupación de pasar menos tiempo trabajando en la documentación y más en la resolución de problemas de forma iterativa.

Metodologías tradicionales

Metodologías tradicionales también llamadas documentación pesada u orientada. Estas metodologías surgen en un contexto de desarrollo de software muy diferente del actual, basado sólo en una unidad central y terminales tontos [Royce (1970)]. El tiempo, costo de realizar cambios y correcciones, eran muy alto, ya que el acceso a los ordenadores se limitaban y no habían herramientas modernas de apoyo del desarrollo de software, tales como depuradores y analizadores de código. Así que el software era todo planificados y documentados, antes de su aplicación. La principal metodología tradicional y muy utilizada hoy es el modelo clásico.

Metodologías ágiles

El término "Ágil" se volvió popular en el 2001, cuando diecisiete expertos en procesos representaron los métodos de desarrollo de software Scrum [Schwaber y Beedle (2002)], Extreme Programming (XP) [Beck (1999)] y los demás estableciendo principios comunes compartidos por todos estos métodos. Luego se estableció la Alianza Ágil y el establecimiento del "Manifiesto Ágil "[Manifiesto Agile (2004)].

El "Manifiesto Ágil" no rechaza los procesos y herramientas tales como, documentación, negociación de contratos o la planificación, sino que simplemente muestra que tienen una importancia secundaria en comparación con los individuos y las interacciones con el software ejecutable, con la colaboración del cliente y las respuestas rápidas a cambios y alteraciones. Estos conceptos son afines a mejorar la forma en que las pequeñas y medianas organizaciones trabajan y responden a los cambios. Entre las metodologías ágiles la más conocida es la programación extrema.

Comparación de Metodologías

Metodologías ágiles no tienen nada nuevo [Cockburn et al. (2001)]. Lo que diferencia a las metodologías es el enfoque y los valores tradicionales. Las ideas de metodologías ágiles van orientadas a la atención de las personas en lugar de los procesos o algoritmos. Además, existe la preocupación de gastar menos tiempo en la documentación y más en la aplicación. De alguna manera, a pesar de que XP es una nueva metodología y se considera por muchos como una revolución, no presenta muchos puntos revolucionarios. De hecho, XP incluye una serie de prácticas que se han utilizado desde el comienzo de la computación electrónica, tales como la programación en parejas y la propiedad colectiva del código.

Una característica de metodologías ágiles es que son adaptativas en lugar de ser predictivas. Con esto, se adaptan a los nuevos factores derivados del desarrollo del proyecto, en lugar de tratar de analizar previamente todo lo que pueda ocurrir durante el desarrollo. Este análisis preliminar es difícil y presenta altos costos, y puede convertirse en un problema si no se quieren hacer cambios en los planes. Por ejemplo, seguir estrictamente la planificación, será necesario que el equipo pueda trabajar bajo presión y hacer una gran cantidad de horas extras, lo que afecta la calidad de software.

Para ser considerado verdaderamente metodología ágil deben aceptar el cambio en lugar de tratar de predecir el futuro. El problema no es el cambio en sí mismo, porque se producirá de todos modos. El problema es cómo recibir, evaluar y responder a los cambios. Un ejemplo, aplicaciones basadas en Web se modelan mejor usando metodologías ágiles, ya que el entorno Web es muy dinámico. Los métodos tradicionales deben aplicarse sólo en situaciones en las que los futuros requisitos del software son estables y predecibles. Estas situaciones son difíciles de alcanzar, ya que los requisitos para el desarrollo de software son cambiantes. Se estima que si todos los cambios han costado "1x", cuando se hacen los requisitos, tendrán un costo de "60x a 100x" cuando se realicen en la fase de ejecución [Pressman (2001)], cuando se utiliza el modelo clásico. Por lo tanto, los cambios en los requisitos del modelo clásico no son deseables.

Debido a que son relativamente nuevas, son pocas las herramientas disponibles que apoyan el proceso ágil de desarrollo. Entre los soportes existentes, la mayoría de los Extreme Programming solamente están en la fase de investigación de desarrollo. Este es un xplanner, herramienta de código abierto que soporta el bien conocido XP, principalmente ayudando sobre todo en la fase de planificación [Xplanner, (2004)]. El desarrollo de esta herramienta todavía está en marcha, pero ya hay una versión estable para la gestión de casos de usuario, la gestión de tarea, el avance del proyecto de verificación y la gestión de métricas individuales y de equipo. Entre las características futuras de la herramienta están integración con otras metodologías ágiles, especialmente Scrum.

El XP es ideal para su uso en proyectos en los que los interesados no saben exactamente lo que quieren y puede cambiar muchas ideas durante el desarrollo del proyecto. Con retroalimentación constante, es posible adaptarse rápidamente a cualquier cambio en los requisitos. Estos cambios en los requisitos son a menudo críticas en los métodos tradicionales, que no presentan medios para adaptarse rápidamente a los cambios.

Otro punto positivo de las metodologías ágiles es las entregas constantes de partes operativas del software. De esta manera, el cliente no tiene que esperar mucho tiempo para ver el programa de trabajo, no como las metodologías tradicionales.

La integración en curso y las pruebas también permiten mejorar la calidad del software. Hay más fases que tienen que ser un módulo de integración, ya que se integran continuamente y cualquier problema se resuelve al instante.

[https://github.com/griffith1989/Ing-en-Software-Tarea-1- Ayudantia](https://github.com/griffith1989/Ing-en-Software-Tarea-1-Ayudantia)