

Relación 1.1

Salvador Romero Cortés

8. ¿En qué consiste la programación extrema (XP)?

La programación extrema es una metodología de ingeniería de software con el objetivo de mejorar la calidad del software y reducir los tiempos de respuesta para adaptar el código a los cambiantes requisitos del cliente.

Se asume que es imposible prevenir todos los imprevistos que puedan surgir durante el desarrollo. Consiste entonces en adaptarse a estas circunstancias en lugar de intentar preverlas. Para ello el cliente toma un rol fundamental en el desarrollo, evaluando frecuentemente una serie de mini-versiones que cumplan con sus requisitos.

Se trata de hacer el mínimo código y lo más simple posible que cumpla con la necesidad del cliente, mortificándolo en las siguientes mini-versiones para adaptarse a los cambios de requisitos.

Durante las primeras versiones se definen pruebas para comprobar que el programa vaya realizando lo que debe con los resultados correctos. Estas pruebas son automáticas y se utilizan para cada mini-versión. Esto proporciona un feedback automático que además sirve para demostrar el funcionamiento al cliente.

Usualmente se trabaja en parejas que se van mezclando en cada versión.

Finalmente, la programación extrema se basa en los principios de simplicidad y buena comunicación.

(Programación extrema: <http://www.chuidiang.org/ood/metodologia/extrema.php> ,
What is extreme programming?: <https://ronjeffries.com/xprog/what-is-extreme-programming/>)

9. La “alianza Ágil” definió una serie de 12 principios que debería tener una metodología para alcanzar niveles aceptables de agilidad. Puede encontrar estos principios en el enlace que aparece en el ejercicio 1. Comente los problemas que intenta solucionar alguno de estos principios.

Algunos principios:

- Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua del software con valor:
Este principio intenta solucionar el problema de que lo solicitado por el cliente y lo entregado por el desarrollador sea distinto. Mediante una evaluación continua cada poco tiempo el cliente puede ver como va evolucionando el programa y le permite cambiar el rumbo en cualquier momento.
- Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente:
Muy relacionado con el punto anterior. Se aprovecha los cambios de necesidad del cliente durante el desarrollo para poder entregar una versión que se adapte mejor al mercado o al cliente de manera que llega antes que otro software en el que se tenga que adaptar tras terminar la primera fase de desarrollo.
- Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
Relacionado con los dos puntos anteriores. Entregando versiones cada 2 semanas se permite al cliente poder ver como va evolucionando el programa y poder solicitar cambios o novedades durante el desarrollo, de manera que en la siguiente mini-versión podrá verlos funcionar. En general, proporciona bastante flexibilidad.
- La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
Como se van lanzando mini-versiones cada poco tiempo se debe asegurar que cada una de estas versiones funciona bien. Si se cumple esto, se agiliza el desarrollo puesto que aparecen múltiples versiones que se sabe con seguridad que funcionan correctamente.
- El método más eficiente y efectivo de comunicar información al equipo de desarrollo es la conversación cara a cara.

Relación 1.1

Salvador Romero Cortés

Puesto que el cliente debe comunicarse continuamente con el equipo de desarrollo es importante que la comunicación sea clara y se transmitan las ideas correctamente para que en la siguiente mini-versión no haya que volver a repetir o adaptar lo que no se entendió. El cara a cara es probablemente la mejor manera de una comunicación clara.

- El software funcionando es la medida principal del progreso.
Para poder enseñar al cliente como va evolucionando el programa es necesario poder demostrar que funciona en cada pequeña versión. Por esto mismo, los tests cobran una gran importancia.

(Principios del Manifiesto Ágil : <http://agilemanifesto.org/iso/es/principles.html>)