

PROGRAMACIÓN XP

Extreme Programming

Anderson Rodriguez

Diego Fernando Murillo López

José Sebastián Salguero Carvajal

Marvy Martínez Sánchez

Nancy Rojas Gordillo

HISTORIA



Enfoque de la ingeniería de software formulado por Kent Beck.

Extreme Programming Explained: Embrace Change (2000). Ingeniero estadounidense.

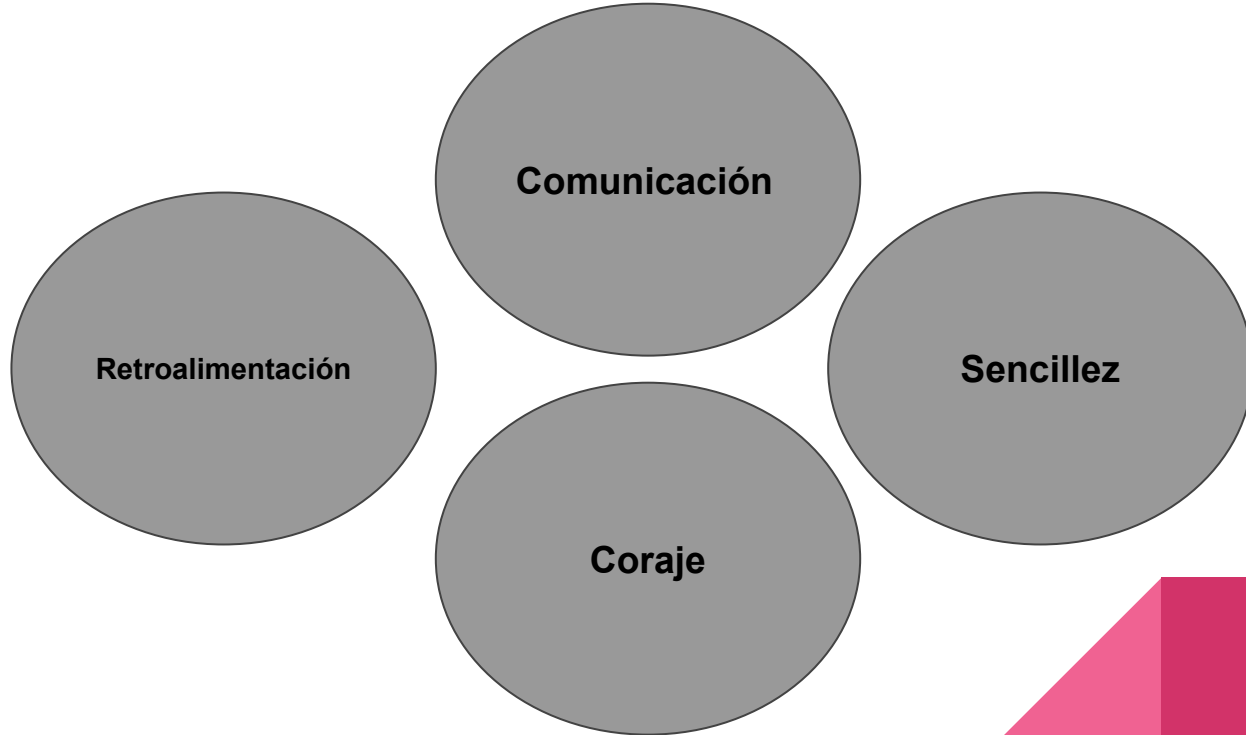
Introducción

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

¿Qué es PROGRAMACIÓN EXTREMA (XP)?

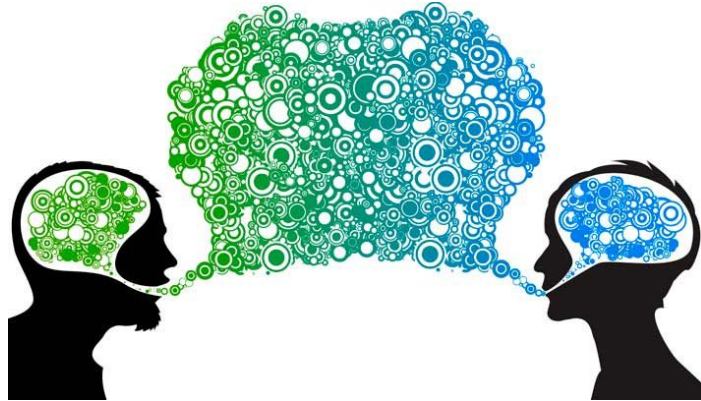
- ❖ Metodología liviana de desarrollo de software.
- ❖ Conjunto de prácticas y reglas empleadas para desarrollar software.
- ❖ Basada en diferentes ideas acerca de cómo enfrentar ambientes muy cambiantes.

VALORES



Comunicación

Muy importante. La XP ayuda mediante sus prácticas a la comunicación entre los integrantes del grupo de trabajo: jefes de proyecto, clientes y desarrolladores.



<https://prezi.com/ljawkezgv4bb/la-comunicacion-empresarial/>

Sencillez

Los programas deben ser los más sencillos posibles y tener la funcionalidad necesaria que se indican en los requisitos. No hay que añadir algo que no se necesite *hoy*.



<http://www.gettyimages.es/detail/ilustraci%C3%B3n/female-business-idea-assembly-ilustraciones-libres-de-derechos/184825065>

Retroalimentación

Las pruebas que se le realizan al software nos mantiene informados del grado de fiabilidad del sistema.



<http://carbonilla.net/2013/09/03/retroalimentacion-afectos-y-motivacion/>

Coraje

Asumir retos, ser valientes ante los problemas y afrontarlos.

El intentar mejorar algo que ya funciona.



PRÁCTICAS

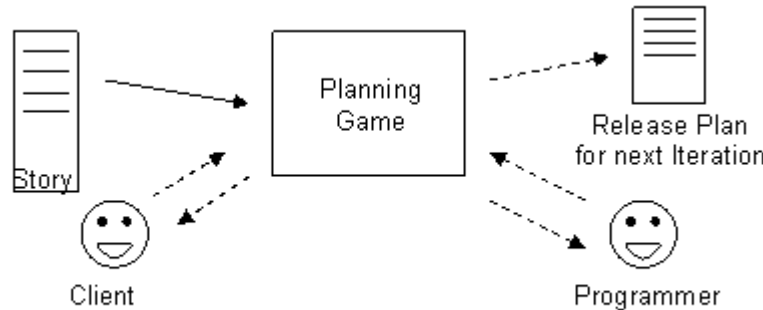
- The Planning Game
- Small Releases
- Metaphor
- Simple Design
- Test-Driven Development
- Refactoring
- Pair Programming
- Collective Ownership
- Continuous Integration
- 40-Hour Workweek
- On-site Customer
- Coding Standards



El Juego de la Planificacion (The Planning Game)

Determinar Alcance de las Nuevas Versiones

Prioridades del Negocio-Estimaciones Técnicas

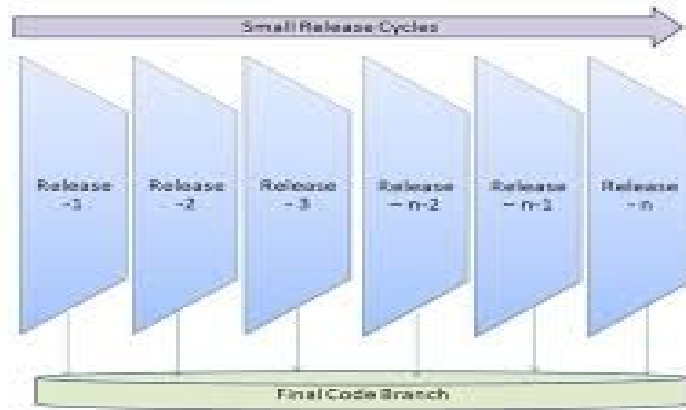


http://www.jucs.org/jukm_1_2/learning_to_tailor_documentation/jukm_01_02_0103_0111_knauss.html

Pequeñas Entregas (Small Releases)

Versiones Pequeñas Como sea Posible

Contiene los requisitos del negocio más importantes



Metáfora (Metaphor)

Historia de cómo funciona el Sistema

Ayuda a cualquier persona a entender el objeto del programa.



<http://metaxulugardeencuentro.blogspot.com.co/2014/05/simbolo-creatividad-y-metaphora-14.html>

Diseño Sencillo (Simple Design)

- Mayor valor es el programa más sencillo que cumpla los requerimientos.
- Cubrir la necesidades inmediatas del cliente.
- Permite eliminar redundancias y rejuvenecer los diseños obsoletos de forma sencilla.



Pruebas (Testing)

- No debe existir ninguna característica sin probar.
- Los programadores escriben pruebas.
- Los clientes realizan pruebas funcionales.
- El Resultado es un programa mas seguro capaz de aceptar nuevos cambios.



Refactorización (Refactoring)

Programadores reestructuran el Sistema sin cambiar su comportamiento Para:

- Eliminar Duplicidad
- Mejorar la Comunicación
- Simplificar
- Añadir Flexibilidad



Programación en Parejas (Pair Programming)

- Dos personas en una sola máquina.
- En cada pareja existen 2 roles.
 - Escritor de código.
 - Pensador estratega.
- Parejas dinámicas.
- Socio experto cooperativo.
- Equipo conjunto y colaborativo.



Imagen extraida de <http://thumbs.dreamstime.com>

Propiedad Colectiva (Collective Ownership)

- Cualquiera puede modificar el código en cualquier momento.
- Nadie es dueño del código que escribe.
- Todos son responsables de todo el sistema.
- Edición con pensamiento global y conjunto.
- Uso de control de versiones con enfoque distinto.



Imagen extraida de <http://p4pconsulting.com/employability/image>

Integración Continua (Continuous Integration)

- Disminuir el tiempo de integración.
- Integrar continuamente nuevas características.
- Existe solo la mitad de cambios que integrar (P.Programming).
- Reduce los conflictos de integración (Small Releases).
- Reduce costos de integración.



k18880828 fotosearch.com ©

Imagen extraída de <http://fscomps.fotosearch.com/>

40 Horas Semanales (40-Hour Week)

- Programador cansado = código de menor calidad.
- 40 horas de trabajo a la semana.
- Nunca 2 semanas seguidas de trabajo extra.

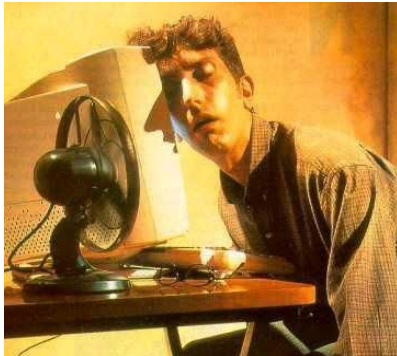


Imagen extraida de <http://4.bp.blogspot.com/>

Vs



Imagen extraida de <http://k22.kn3.net/>

Cliente en sitio (On-Site Customer)

- Se requiere de la participación del cliente.
- Cliente aporta ideas y responde inquietudes.
- Disminuye la documentación.
- Mayor satisfacción del cliente.



Imagen extraída de <http://2.bp.blogspot.com/>

Estándar de Codificación (Coding Standards)

- Patrones en la creación de código.
- Comentarios de código.
- Organización de código (Indentación).
- Patrones de diseño (MVC).



Imagen extraída de <http://wiki.kuali.org>

Proceso de Desarrollo

- Equipo de gestión (o diseño) , equipo de desarrollo y los clientes finales.
- La relación entre el equipos y clientes es diferente a las metodologías tradicionales
 1. Interacción con el cliente.
 2. Planificación del proyecto.
 3. Diseño, desarrollo y pruebas.

Interacción con Cliente

- Interacción continua en toda la fase de desarrollo.
- Revisión de cada cambio e integración.
- Historias de usuario.
 - a. Fase Uno: Listar Características del producto final.
 - b. Fase Dos: Dividir las historias en trabajos a realizar.
- 1 a 3 semanas de trabajo por historia.
- Catalogar cada historia en tarjetas

A quien hace feliz la programación XP?

- Al cliente porque no tiene que esperar mucho para poder ver resultados.
- Al programador(es) porque les evita papeleo
- Al programa porque queda más lógico
- A tu vecina porque tienes dinero siempre

A quien hace feliz la programación XP?

En el mundo paralelo de XP todo parece estar al revés

- Lo rápido queda bien.
- Lo importante no es tan importante como lo urgente
- Las personas y su interacciones son prioritarios, no los proceso, no las herramientas.
- La documentación queda en segundo plano.

A quien hace feliz la programación XP?

En el mundo paralelo de XP todo parece estar al revés

- Colaborar al cliente por encima del papeleo.
- El cambio es bienvenido
- Reducir la burocracia, aumentar productividad.
- Más sentido común, menos complicaciones.

PREGUNTAS:

1). Que es la refactorización Quien creo la programación extrema?

2). Qué es la Integración en el desarrollo de software?

3). Cuales son las variables en XP?

4). Que se utiliza en la metafora?

Para qué sirve la metáfora en la programación XP?

5). Ventajas y desventajas de XP?

Bibliografía

- Beck, K. (2000). *Extreme programming eXplained*. Reading, MA: Addison-Wesley.
- Xprogramming.com, (2015). *RonJeffries.com*. [online] Available at: <http://www.xprogramming.com> [Accessed 19 Nov. 2015].
- David Valverde Blog, (2007). *Introducción a la Programación Extrema (XP) - David Valverde Blog*. [online] Available at: <http://www.davidvalverde.com/blog/introduccion-a-la-programacion-extrema-xp> [Accessed 19 Nov. 2015].
- Procesosdesoftware.wikispaces.com, (2015). *ProcesosdeSoftware - METODOLOGIA XP*. [online] Available at: <https://procesosdesoftware.wikispaces.com/METODOLOGIA+XP> [Accessed 19 Nov. 2015].