



TECNOLOGICO NACIONAL CEICOM
CARRERA: SISTEMAS INFORMATICO

F

EL PROCESO XP

Materia: Análisis y Diseño de Sistemas

Elaborado por: Fabrica Plaz Estevan

Docente: Ing. Baltazar Llusco Ever Jaime

Fecha: 15 de octubre de 2018

Cochabamba – Bolivia

El proceso XP

El ciclo de vida de los proyectos bajo XP puede verse como una sucesión de definiciones por parte del cliente una aceptación y continuo desarrollo de parte de los programadores. La diferencia de las otras metodologías es que estas sucesiones ocurren en muy corto tiempo y esto le da la posibilidad de corregir y ajustar el rumbo. A grandes rasgos podemos mencionar los siguientes pasos:

1. El cliente decide el valor de negocio que quiere implementar.
2. Se estima el esfuerzo necesario.
3. El cliente selecciona que construir.
4. Se construye la selección.

Una vez terminada la construcción seleccionada, se retorna al punto 1. El ciclo se repite tantas veces como sea necesario hasta la finalización. En el ciclo de vida de XP encontramos seis fases.

Exploración

Es la primera fase, cuando el cliente plantea cuáles son sus inconvenientes y que soluciones considera apropiadas. Según generan las historias de usuario todo el equipo comienza a aprender a cerca de las herramientas que se va a utilizar, sus destacados, y sortean las dificultades iniciales. Esta fase puede durar semanas, dependiendo de los conocimientos del equipo.

Planificación de la entrega

Una vez finalizada la historia del usuario, el cliente determina la prioridad y los programadores estiman el esfuerzo necesario para cada una. Esta no deberá demorar más de tres meses.

Iteraciones

Esta fase es un conjunto de iteraciones de entre 2 y 3 semanas correspondientes a las planificaciones y a la definición de la arquitectura. En la iteración se tiene en cuenta las historias de usuario seleccionado, las faltantes y las tareas que se realizarán. Todas las historias de usuario se volcarán en tareas de programación que luego serán abordadas por las parejas de programadores. Una vez finalizado la etapa, el proyecto está listo para entrar a la fase de reproducción.

Muerte del proyecto

La muerte o finalización del proyecto es cuando se agota todas las historias de usuario. También puede ocurrir antes de lo pautado por falta de fondos o cambios en las condiciones del negocio del cliente.

Equipo

Definimos equipo como un pequeño conjunto de personas con habilidades complementarias que tiene un mismo objetivo. Su conformación es una tarea clave en toda metodología. Veamos que hace cada parte.

Ocupación del cliente

- Decidir que se implementa
- Saber el estado
- Conocer el progreso
- Añadir, cambiar o quitar requerimientos
- Obtener un sistema funcionando cada cierto periodo de tiempo

Tareas del programador

- Decidir cómo se implementan las soluciones
- Crear el sistema con la mayor calidad posible
- Pedir aclaraciones al cliente
- Estimar el esfuerzo de implementación
- Cambiar las estimaciones

Roles en XP

Las posiciones que cada recurso humano ocupa dentro de la metodología están contempladas y delimitadas por esta. Cada integrante del equipo tiene una posición determinada que se le da ciertos privilegios y lo hace responsable de las tareas. Los roles en XP son los siguientes:

- ❖ Cliente
- ❖ Programador
- ❖ Encargado de pruebas(tester)
- ❖ Encargado de seguimiento (tracker)
- ❖ Entrenador(coach)
- ❖ Gestor (big boss)

Existen otros roles que son complementarios en todo proceso, estos son:

- ❖ Consultor
- ❖ Analista
- ❖ Operador
- ❖ Manager del proyecto

Veamos ahora cuales son las tareas de cada uno de los integrantes de un equipo XP.

Cliente

Al utilizar el término cliente nos referimos a un conjunto de personas que actuarán juntas para definir el negocio. Sus deberes son:

1. Confeccionar historias de usuario
2. Asignar las prioridades a las historias de usuario
3. Estar enfocado en aportar mayor valor al negocio
4. Escribir o especificar los test de aceptación
5. Es el responsable de validar el producto

Programador

Los programadores deben mantener altos niveles de comunicación e interacción. Son los encargados de:

- ❖ Escribir el código del programa
- ❖ Estimar las historias del usuario
- ❖ Escribir las pruebas de unidad
- ❖ Participar en las propuestas por la metodología

Encargado de pruebas (tester)

El **tester**, también conocido como encargado de pruebas, es la persona que esta designada para realizar los test funcionales y asegurar que el sistema trabaje de una manera que resulte adecuada. Debemos dejar claro que el encargado de pruebas es un integrante más del equipo y no un consultor o revisor externo. Entonces el tester se encarga de cumplir las siguientes tareas:

- ❖ Ayudar al cliente a escribir las pruebas funcionales
- ❖ Ejecutar las pruebas regularmente
- ❖ Difundir los resultados en el equipo
- ❖ Seleccionar las herramientas adecuadas para ejecutar las pruebas

Encargado de seguimiento (traker)

El encargado de seguimiento debe comprobar que el conjunto de tareas se esté realizando de manera adecuada a si también medir introduciendo métricas simples para el proyecto. El encargado de seguimiento se ocupa de realizar las siguientes tareas:

- ❖ Discutir y colaborar con el equipo
- ❖ Adoptar métricas(en lo posible simple)
- ❖ Discutir y difundir esas métricas
- ❖ Verificar las desviaciones del proyecto
- ❖ Refinar los métodos de estimación utilizados
- ❖ Realizar el seguimiento de las iteraciones
- ❖ Reportar los progresos del equipo
- ❖ Conservar los valores históricos

Entrenador (coach)

El entrenador es el responsable de conocer a fondo el proyecto de XP para poder explicarlo a los demás y así facilitar la comunicación del equipo. Para cumplir de forma correcta con su rol, debe tener conocimientos técnicos acabados.

Gestor (big boss)

El gestor es el vínculo entre los clientes y los programadores tienen un rol de facilitador.

Artefactos de XP

Historias del usuario

La fase de requisitos existe en todas las metodologías de desarrollo debido a que es el cliente el que manifiesta sus deseos o necesidades por un producto en particular. En el caso de XP, las historias de usuario son la forma en que se especifican los requisitos del cliente. Consisten en documentos. Los datos a incluir en las historias de usuario dependen de cada equipo, pero generalmente se aceptan los que propone Beck, que son:

- ❖ Fecha
- ❖ Tipo de actividad
- ❖ Prueba funcional
- ❖ Numero de historia
- ❖ Prioridad técnica
- ❖ Prioridad del cliente
- ❖ Referencia a historias antiguas
- ❖ Riesgos

- ❖ Estimación técnica
- ❖ Descripción
- ❖ Notas
- ❖ Seguimiento (observaciones, tareas pendientes, fechas)

Tareas de ingeniería

Las tareas de ingeniería son elementos que se vinculan más al desarrollo, ya que permiten tener un acercamiento con el código.

Pruebas de aceptación

Las pruebas de aceptación surgen a partir de los elementos que el usuario destaca. Por lo tanto el cliente quien decide cual es el escenario correcto para superar una prueba. Las pruebas de aceptación son pruebas de caja negra ya que el enfoque está orientado a conocer el producto final.

Tarjetas CRC

Las tarjetas **clase-responsabilidad-colaborador** son elementos que intentan describir y facilitar el trabajo. Contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores. Una clase puede ser cualquier concepto, elemento o persona. Los eventos y realiza son sus responsabilidades, atributos y métodos. Colaborador es a todo aquel que interactúa con la clase. Las actividades que supone son:

- ❖ Encontrar clases
- ❖ Hallar responsabilidades
- ❖ Definir colaboradores
- ❖ Confección tarjetas

Las tarjetas de CRC son un elemento de discusión en la metodología, ya que han atravesado distintas etapas de aceptación y rechazo. Al ser una herramienta simple y de fácil construcción cuando no son útiles al proceso y descartarlas.

Las balas de plata

Normalmente, todos los libros de desarrollo y de metodologías dicen que no existe la bala de plata y que solo el trabajo duro. Entonces Michele Marchisi y Don Wells nos ofrece algunas balas de plata para el desarrollo basado en XP.

1. No observar el proceso como una competencia de tiempos
2. Recopilar la información previa existente y determinar cómo utilizar
3. Automatizar siempre que sea posible. Debemos utilizar todas aquellas herramientas que nos permitan automatizar la gran cantidad de tareas diarias al igual que todas las partes del proceso.
4. No confiar en las estadísticas, las estadísticas obtenidas con métodos simples a partir de la información recopilada no deben ser tomadas en cuenta a menos que sean revisadas y refinadas con métodos orientados al software.

Programación en pares

La programación en pares es una de las prácticas más discutidas de XP, y consiste en que dos programadores desarrollen juntos, la programación en pares está pensada para atacar un problema no solo resolviendo, sino pensando en la mejor forma de hacerlo y confrontando esas soluciones en el momento de escribirlas. La programación en pares debe seguir ciertas pautas:

- Las tareas que resulten difíciles para realizar con un duo de programadores pueden ser llevadas a cabo de forma individual.
- Difundir métodos similares de corrección del código a todos los desarrolladores.
- Facilitar la comunicación entre los programadores, teniendo especial cuidado en la forma de transmitir sus fallas.
- Las habilidades interpersonales deben ser las adecuadas para este tipo de desarrollo
- Los objetivos del trabajo deben ser claros y similares para el conjunto
- La correcta distribución física de los desarrolladores y un buen ambiente de trabajo son esenciales para que estén cómodas.

El intercambio de ideas y conocimientos que se produce entre los desarrolladores los ayuda a elevar su nivel de codificación.

Pruebas en XP

Probar el software no es patrimonio de XP y, sin embargo, existe una serie de tipos de prueba que si son utilizados por esta metodología estos son:

Inspección visual: este tipo de prueba es normalmente asignado al cliente

Pruebas de rendimiento: estas pruebas se enfocan en la capacidad de respuesta del sistema y sus límites de operación

Pruebas de unidad: este enfoque para probar el software está actualmente muy difundida, consiste en testear los elementos o módulos por separado y asegurar su correcto funcionamiento. Las pruebas de unidad se adaptan perfectamente al desarrollo orientado a objetos debido a que es posible utilizar una clase o más.

Pruebas de aceptación: implican confrontar lo desarrollado con el deseo o la idea que tiene el usuario. Cada historia de usuario existente tendrá una o más pruebas de aceptación.

Solucionar problemas

Con la programación extrema tenemos distintas metodologías para confrontar con los conflictos, por ejemplo:

- **Retrasos y desviaciones:** son soluciones con las entregas periódicas en cortos lapsos de tiempo. Esto permite estar sobre la fecha de entrega y no tener sorpresas de último momento.
- **Cancelación de proyecto:** suele producirse por muchas causas.
- **Sistemas con errores:** los ambientes de reiteración, refactorización y pruebas continúa favorecer la disminución notoria de errores que en otros ciclos tardan más en ser descubiertos y corregidos.
- **Requisitos mal comprendidos:** los problemas de requisitos, su obtención y la dificultad para interpretarlos quedan a un lado gracias a la incorporación del cliente en el equipo.
- **Cambios de negocio:** versiones cortas por si surgen cambios de negocio.
- **Falsa riqueza:** la falsa riqueza o características innecesarias no se producen en XP porque al asignar prioridades a las tareas, tenemos control de estar desarrollando absolutamente lo requerido
- **Cambios de personal:** tienen menos impacto en esta clase de equipos debido a que en general todos los trabajadores se sienten a gusto, además de que la comunicación favorece la resolución de conflictos del equipo.

