TECNOLÓGICO NACIONAL
CEICOM

CARRERA DE SISTEMAS INFORMÁTICOS

B

Práctica 02

CUESTIONARIO

Materia: Análisis y diseño de sistemas II

por: Jonathan Ponce Rojas

Docente: Ing. Baltazar Llusco Ever Jaime Fecha: 28 de septiembre del 2018

Cochabamba – Bolivia

Exploración

Es la primera fase, cuando el cliente plantea cuáles son sus movimientos y que soluciones considera apropiadas. Todo el equipo comienza a aprender acerca de las herramientas que va a utilizar, sus puntos destacados, y se sortean las dificultades iniciales. La fase puede durar varias semanas, dependiendo de los conocimientos del equipo como de la dificultad del proyecto en cuestión.

Planificación de la entrega

Una vez finalizadas las historias de usuario, el cliente determina la prioridad y los programadores estiman el esfuerzo necesario para cada una. Esta no debería demorar más de tres meses, aunque como hemos comentado la metodología puede presentarse a proyectos más grandes.

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 tienen en cuenta las historias de usuario seleccionadas, las faltantes y las tareas que se realizaran.

Producción

En la fase de producción se realizan y se evalúan si es necesario incorporar funcionalidad extra.

Mantenimiento

La fase de mantenimiento, al igual que en otras metodologías, es el intento de sostener funcionando la versión actual. Por eso, al finalizar la producción de la primera versión, inmediatamente se le brinda soporte al cliente.

Muerte del proyecto

La muerte o finalización del proyecto es cuando hemos agotado todas las historias del usuario. También puede ocurrir antes de lo pautado por falta de fondos o cambios en las condiciones del negocio del cliente que hacen no deseable el producto.

EQUIPO

Definimos equipo como un pequeño conjunto de personas con habilidades complementarias que tienen un mismo objetivo.

Ocupaciones 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 determinadas por esta. Los roles en XP son los siguientes:

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

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

- Consultor
- Analista
- Operador
- Manager del proyecto

Cliente

Al utilizar el término cliente nos referimos a un conjunto de personas que actuaran juntas para definir el negocio.

- Confeccionar historias de usuario
- Asignar las prioridades de las historias de usuario
- Estar enfocado en aportar mayor valor al negocio
- Escribir o especificar los test de aceptación
- 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 programador
- Estimar las historias de usuario
- Escribir las pruebas de unidad
- Participar en las propuestas por 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. Los desarrolladores continuamente están trabajando sobre

el código y la resolución de defectos, por lo tanto, los errores encontrados por los encargados de pruebas tienden a ser menores.

El tester se encarga de cumplir las siguientes tereas:

- 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 (tracker)

El encargado de seguimiento debe comprobar que el conjunto de tareas se esté realizando de manera adecuada, acercando las estimaciones con la realidad. Se encarga de medir introduciendo métricas simples para el proyecto. Para resumir mejor las tareas que corresponden, podemos decir que el encargado se seguimiento se ocupa de realizar lo que vemos a continuación:

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

Entrenador (coach)

En extreme programming comenzamos a ver un rol que luego es utilizado por otras metodologías y que se conoce como coach o entrenador. El entrenador es responsable de conocer a fondo el proceso de XP para poder explicarlo a los demás y así facilitar la comunicación del equipo.

Gestor (big boss)

El gestor es el vínculo entre los clientes y los programadores. Al igual que en otras metodologías encontramos un rol facilitador, podemos decir que esa es la característica principal del big boss.

ARTEFACTOS DE XP

Historias de 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, generalmente de tipo gráfico, en donde se describen las características esperadas. Los datos a incluir en las historias de usuario dependen de cada equipo, pero generalmente se aceptan los que proponen Beck, que son:

- Fecha
- Tipo de actividad
- Prueba funcional
- Numero de historia
- Prioridad técnica
- Prioridad del cliente
- Referencia a historias antiguas
- Riegos
- Estimación técnica
- Descripción
- Notas
- Seguimiento

Tareas de ingeniería

Las tareas de ingeniería son elementos que se vinculan más al desarrollador, ya que permiten tener un acercamiento con el código. Como una historia de usuario puede ser dividida en varias tareas de ingeniería, es conveniente que estas incorporen una numeración que permita localizar la historia de usuario original. Los datos necesarios son:

- Identificador
- Relación con la historia
- Tipo de tarea
- Responsable

Pruebas de aceptación

Las pruebas de aceptación surgen a partir de los elementos que el usuario destaca. Por lo tanto, es 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 y el resultado que genera.

Tarjeta 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. La confección de tarjetas supone llevar a cabo las actividades que mencionamos a continuación:

- Encontrar clases
- Hallar responsabilidades
- Definir colaboradores
- Confeccionar tarjetas

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 y a conciencia puede llevarnos al éxito.

- No observar el proceso como una competencia de tiempos. De acuerdo con los principios generales, la medición posibilita obtener datos que luego de ser analizados nos permiten conocer cuál es el estado general del proyecto
- Recopilar la información previa existente y determinar cómo utilizarla. Es interesante y productiva la idea de recopilar información anterior al proyecto de distintas fuentes, como por ejemplo el departamento contable, resultados de pruebas, atención al cliente, gestión de configuración.
- Automatizar siempre que sea posible. Debemos intentar utilizar todas aquellas herramientas que nos permitan automatizar gran cantidad de tareas diarias al igual que todas las partes del proceso.
- 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 (pair programming) es una de las practicas más discutidas de XP, y consiste en que dos programadores desarrollen juntos. Se aplica en cualquier momento del proceso y es una técnica que ha sido criticada debido a que tradicionalmente se asocia que dos personas trabajando por separado producen más cantidad de código. La programación en pares debe seguir ciertas pautas para ser exitosa:

- Las tareas que resulten difíciles para realizar con dúo 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 cómodos.

PRUEBAS 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 (decimos que las pruebas de inspección y aceptación corresponden al cliente mientras que las otras son del ámbito del programador)

Pruebas de rendimiento: estas pruebas se enfocan en la capacidad de respuesta del sistema y sus límites de operación. Deben intentar establecer aquellos elementos que aún no responden como se espera para poder optimizarlos. Este tipo de pruebas supone un conocimiento tecnológico acabado, por eso son generadas por

profesionales de aseguramiento de la calidad que cuentan con el asesoramiento de arquitectos de software.

Pruebas de unidad: este enfoque para probar el software está actualmente muy difundido, consiste en testear los elementos o módulos por separado y asegurar su correcto funcionamiento. Se observa el comportamiento del código, haciendo que los desarrolladores intenten mejorarlo en un proceso continuo denominado refactorización.

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

Hemos comentado en anteriores oportunidades que las distintas metodologías pretenden solucionar los problemas del desarrollo.

- **Retrasos y desviaciones:** son solucionados con las entregas periódicas en cortos lapsos de tiempo. Esto permite estar sobre las fechas de entrega y no tener sorpresas de último momento que, en calendarios ajustados, suelen producir grandes inconvenientes.
- **Cancelación del proyecto:** suele producirse por muchas causas; sin embargo, al estar mostrando la funcionalidad y el cliente estar en contacto con su producto, hemos que varias de las razones desaparezcan
- **Sistemas con errores:** los ambientes de reiteración, refactorización y prueba continua favorecen 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 en el 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 y la puesta a punto del equipo.

CRITICAS A XP

Cuando nos referimos a las críticas que ha tenido (y tiene) la metodología Extreme Programming, debemos notar que la mayoría están dirigida a las practicas sobre las que se asienta y no a la filosofía de fondo. Las metodologías agiles han venido a quedarse en una industria cada vez más cambiante en donde esperar el tiempo de reacción puede ser muy costoso. La aceptación y convivencia con el cambio son elementos que cualquier metodología debe tener. Las exigencias del mercado actual

hacen que los desarrolladores pocas veces puedan trabajar solo la cantidad del tiempo que se han comprometido cargas horarias muchos mayores.

Actualmente, existen experiencias de desarrollo de software para distintos tipos de industria que han superado todos los factores críticos de la metodología, como el tamaño del proyecto, la cantidad de esfuerzo necesario y las practicas más extremas. Esto debería ser un aliciente más que suficiente para todos aquellos que no confían en XP para su tipo de proyecto.