

Modificar Casos de Prueba



English

Entradas

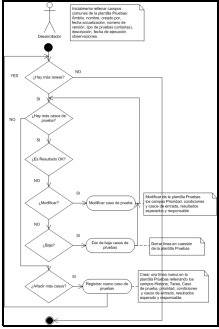
- Pruebas

Salidas

- Pruebas

Solución

Proceso



Tiempo de Desarrollo

- ◆ Para adquirir el conocimiento necesario para desarrollar el producto software:
- ◆ Para crear el Patrón de Producto: 45 minutos.
- ◆ Para aplicar el Patrón de Producto:

Video Explicación

- No aplica

Patrones Relacionados

- Patrón Pruebas Unitarias

Controladores de Calidad

- Ninguno

Plantillas

- Pruebas.xls

Ejemplos

- Ninguno

Herramientas de Soporte

- Será necesario un editor de texto como [OpenOffice Writer](#) o [Microsoft Word](#).
- Además de una herramienta como [Visual Paradigm for UML](#) para la realización de los diagramas expuestos.



Contexto Inicial

Este producto puede usarse en cualquier proyecto en el que una vez ejecutados los casos de prueba alguno de estos no es satisfactorio porque no se haya contemplado alguna casuística en concreto.



Contexto Resultante

Los desarrolladores obtienen casos de prueba modificados, añadiendo o quitando elementos, ya sea por falta de amplitud en su alcance o por exceso.



Problema

Los desarrolladores deben ser capaces de averiguar si el caso de prueba identificado es completo y determinante para, en caso contrario, poder modificarlo adecuadamente. La modificación de los casos de prueba puede realizarse bajo estas circunstancias, entre otras:

1. Si se encuentra un problema, se escribe una prueba que aísla el mismo.
2. Si se esta haciendo recodificación en algún código, y no se esta seguro de cómo debe ser su comportamiento, y todavía no hay una prueba para dicho comportamiento en cuestión, escribe una primera prueba.



Restricciones (*Forces*)

- **Características de las organizaciones:** Este patrón puede utilizarse en los proyectos existentes en cualquier tipo de compañía.
- **Tipo de sistema a desarrollar:** Este producto puede utilizarse en proyectos en los que los requerimientos de usuario sean cambiantes.
- **Tipo de Cliente:** Debe existir, o debe conseguirse, que el área de negocio destinataria del desarrollo se implique en la consecución del mismo.
- **Heurísticas de uso:** :Si se necesita disponer urgentemente del aplicativo o de algunas de sus funcionalidades.



Roles

- Desarrolladores (2 a 12)



Lecciones Aprendidas

- La pareja de programadores debe volver a analizar la tarea relacionada con el caso de prueba no satisfactorio. La actividad de corregir los casos de prueba es la más importante de las que tienen que realizar los miembros del equipo de trabajo, aunque debe mantenerse el equilibrio entre el tiempo que se dedica a la ejecución de las pruebas y a la corrección de las mismas.
- Si el caso de prueba no es satisfactorio debe volver a analizarse la tarea para modificar los casos de prueba identificados.
- El archivo Pruebas es un documento genérico, para todos los tipos de prueba con la casilla pruebas unitarias marcada, que aglutina todos los casos de prueba para una tarea en cuestión. Hay que poner especial atención en aquellos casos de prueba no satisfactorios, pues han de ser modificados. Se pueden añadir o eliminar casos de prueba según se considere oportuno.



Nivel de Madurez

- Este Patrón de Producto no se relaciona con ningún nivel de madurez(N/A).



Conocimientos y Habilidades Básicos



Conocimientos

- Conocimiento del estándar de codificación que define la propiedad del código compartido así como las reglas para escribir y documentar el código y la comunicación entre diferentes piezas de código desarrolladas por diferentes equipos. Los programadores las han de seguir de tal manera que el código en el sistema se vea como si hubiera estado escrito por una sola persona.
- Conocimiento de la visión común de cómo funciona el programa en el que se desarrollan las actividades.



Habilidades

- Capacidad de trabajo en grupo. Todos en un equipo XP contribuyen de la manera que pueden.
- Predicción de qué se habrá terminado para la fecha de entrega, y determinación de qué hacer después.
- Capacidad de programación de a pares. Además de generar mejor código y pruebas, sirve para comunicar el conocimiento a través de los equipos.



Recursos de Información

- Álvarez, José R. y Arias Manuel. Método Extreme programming. Recuperado el 2010-03-05 de <http://www.ia.uned.es/ia/asignaturas/adms/GuiaDidADMS/node61.html>
- Anaya Villegas, Adrian. A proposito de programación extrema XP(extreme Programming). Recuperado el 2010-02-10 de <http://www.monografias.com>
- Beck, K.(2000), Una explicación de la programación extrema. Aceptar el cambio. Ed. Addison Wesley.
- De Seta, Leonardo. Una introducción a Extreme Programming. Recuperado el 2010-03-02 de <http://www.dosideas.com/noticias/metodologias/822-una-introduccion-a-extreme-programming.html>
- Extreme Programming: A gentle introduction. Recuperado el 2010-03-15 de <http://www.extremeprogramming.org/>
- Joskowicz, José. Reglas y prácticas en Xtreme Programming. Recuperado el 2010-03-15 de <http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>
- Letelier, Patricio y Panadés M^a Carmen. Metodologías Ágiles en el desarrollo de software: extreme programming. Recuperado el 2010-03-15 de <http://www.willydev.net/descargas/masyxp.pdf>
- Newkirk, James y Martin, Robert C.(2001), La programación Extrema en la Práctica. Ed Addison Wesley.

