Pruebas Funcionales Historia

English

Entradas

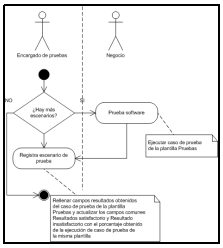
- Pruebas

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 Escribir Pruebas Funcionales para Historia
- Patrón Escribir Casos de Prueba
- Patrón Transformar Historia en Tareas
- Patrón Escribir Historia

Controladores de Calidad

- Ninguno

Plantillas

- Pruebas.xls

Ejemplos

- Ninguno

Herramientas de Soporte

- Será necesario un editor de texto como OpenOffice Writer o Microsoft Word.
- Será necesario un editor de hojas de cálculo como OpenOffice Calc Microsoft Excel
- 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 deban realizarse pruebas funcionales del software generado.

A partir de las historias de usuario se hacen las pruebas funcionales y a cada una le puede corresponder una varias. No se considera que una historia ha sido completada hasta que no pase todas sus pruebas funcionales.Las pruebas funcionales deben hacerse antes de empezar a depurar.

Contexto Resultante

El encargado de pruebas y los clientes obtienen una visión del porcentaje de pruebas satisfactorias para poder así establecer el grado de avance del proyecto.

Problema

Debe proporcionarse al cliente una herramienta lo suficientemente potente como para poder traducir sus ideas a un conjunto de escenarios sobre los que probar el software. Además, dicha herramienta debe permitir la ejecución y mantenimiento de los escenarios descritos por las ideas del cliente .

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

- Encargado de pruebas (1)
- Usuarios del área de negocio (2 como mucho)

Lecciones Aprendidas

- No todas las pruebas funcionales tienen que funcionar al 100% al mismo tiempo. El resultado de las pruebas funcionales no puede ser binario, como en el caso de las pruebas unitarias, ya que provienen de distintos códigos generados por distintos desarrolladores del equipo de trabajo. Con el tiempo, rectificando las pruebas funcionales, el resultado de aproximarse al 100%. Conforme se vaya cerrando una versión, el cliente necesitara clasificar las pruebas funcionales que vayan fallando, estableciendo una prioridad para su corrección.
- El cliente utiliza el software generado en cada uno de los escenarios descritos.
- El documento de entrada Pruebas es un archivo .xls (Microsoft Excel) genérico, para todos los tipos de prueba con la casilla pruebas usuario marcada, que recoge la traducción de las ideas del cliente, en el lenguaje del ámbito técnico, a escenarios de prueba del software generado. En este documento, a la salida, se añade el porcentaje de satisfacción del cliente después de haber ejecutado las pruebas funcionales correspondientes. Ademas se incluye la prioridad para la corrección de las pruebas funcionales que no fueron satisfactorias.

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. Addisson 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 Mª 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 Addisson Wesley.