

Escribir Pruebas Funcionales para Historia



English



Entradas

- Ideas Cliente



Salidas

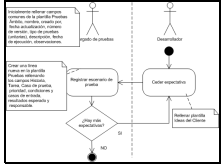
- Pruebas



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 Registrar Progreso
- Patrón Recuperación Tareas
- Patrón Pruebas Funcionales Historia



Controladores de Calidad

- Ninguno



Plantillas

- Ideas_Cliente.doc
- Pruebas.xls



Ejemplos

- Proyecto C3 de Daimler-Chrysler.



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](#) o [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 se deban realizar pruebas sobre escenarios descritos por el cliente.



Contexto Resultante

El equipo de trabajo, clientes incluidos, obtienen un conjunto de escenarios sobre los que probar el código escrito.



Problema

El encargado de pruebas del proyecto, debe ser capaz de traducir las ideas del cliente a pruebas reales del ámbito técnico. Además, debe utilizar dichas ideas del cliente como punto de partida para las variaciones que probablemente harán que falle el software generado.



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

- Los clientes escriben pruebas historia a historia. Se deben preguntar que tendrían que probar antes de estar seguros de que la historia en cuestión se ha completado. Los clientes necesitan alguna herramienta que les permita escribir, ejecutar, revisar y mantener las pruebas funcionales, es decir, una especie de traductor entre el lenguaje del cliente y de la parte técnica.
- El encargado de pruebas del proyecto se reúne con los clientes y transforma sus ideas y expectativas sobre el software en escenarios (casos de prueba) reales que pueden realizarse dentro del ámbito tecnológico.
- El documento de entrada Ideas Cliente contiene las ideas y expectativas del cliente de las funcionalidades que espera que realice el software.
- El documento de salida Pruebas es un archivo genérico de tipo .xls (Microsoft Excel), 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.



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.
-