Diseñar

English

Entradas

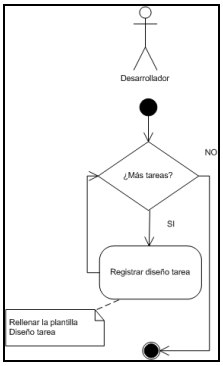
- Pruebas

Salidas

- Diseño Tarea

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 de Integración
- Patrón Modificar Integración
- Patrón Generar Código

Controladores de Calidad

- Ninguno

Plantillas

Diseño_Tarea.doc

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 se siga la filosofía de trabajo en parejas para la implementación de código.

El diseño crea una estructura que organiza la lógica del sistema, un buen diseño permite que el sistema crezca con cambios en un solo lugar.

Contexto Resultante

Se obtienen diseños sencillos de pequeñas partes del sistema.

Problema

El diseño en parejas no consiste en que una persona escriba y la otra mire, es un dialogo entre dos personas que intentan simultáneamente programar, diseñar, analizar y probar. Debe elegirse una estrategia de diseño sencilla que de lugar a un diseño sencillo, además debe encontrarse rápidamente la manera de comprobar la calidad del diseño elaborado. Puede surgir incertidumbre (¿Cuándo añadimos más a lo diseñado?), por ejemplo:

1. Algunas veces el cliente elimina la característica del diseño por anticipado.
2. A veces se aprende una manera mejor de trabajar cuando se esta trabajando de otra manera.

Los diseños deben de ser sencillos, si alguna parte del sistema es de desarrollo complejo, lo apropiado es dividirla en varias.Si hay fallos en el diseño o malos diseños, estos deben ser corregidos cuanto antes.

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

- No hay que ser ambiciosos inicialmente. No debemos esperar un beneficio muy alto inicialmente. Hay que ser pacientes e ir añadiendo al diseño según avanza el desarrollo. La estrategia de diseño debe ser:

1. Comenzar con una prueba, así sabremos lo que estamos haciendo.
2. Diseñar e implementar justo lo suficiente para que la prueba funcione.
3. Repetir.
4. Si siempre se ve la posibilidad de hacer el diseño mas simple, hacerlo.

- Diseñar inicialmente lo mas sencillo posible para ir incrementando su valor.

- 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. En este documento se refleja su responsable e historia asociada.

- El archivo Diseño Tarea es un documento con el diseño de la tarea a implementar.

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.