

Modificar Integracion



English



Entradas

- Pruebas



Salidas

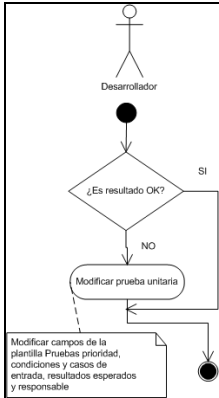
- Pruebas



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 de Integración



Controladores de Calidad

- Ninguno



Plantillas

- Integración_Continua.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 este presente el concepto de integración continua y propiedad colectiva, por el cual cualquier miembro del equipo de trabajo puede modificar lo realizado por otro miembro.



Contexto Resultante

Los desarrolladores obtienen conclusiones de las pruebas de integración que no fueron satisfactorias o que pueden ser simplificadas.



Problema

Los desarrolladores deben mantener los principios de la integración continua para que nunca se pierda demasiado tiempo en la identificación de un problema causado por un error cometido en las últimas semanas. En la integración deben tenerse en cuenta que el sistema completo (código y pruebas) debe poder ser revisado por cualquier miembro del equipo.



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

- Recodificar continuamente tiene el efecto de fraccionar el sistema en montones de pequeños objetos y montones de pequeños métodos. Esto disminuye la probabilidad de que dos parejas de programadores cambien la misma clase o el mismo método al mismo tiempo. Si lo hacen, el esfuerzo requerido para reconciliar los cambios es pequeño, por que cada uno representa solo unas pocas horas de desarrollo.
- La modificación de la integración en función de pruebas no satisfactorias o identificación de elementos que puede simplificarse más aun.
- 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 las pruebas de integración no satisfactorias, pues han de ser modificadas. En este documento se refleja su responsable e historia asociada. Se rellena el campo correspondiente al resultado de la prueba y las observaciones asociadas.



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, Adrián. 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://ie.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.