Design

Español

Entries

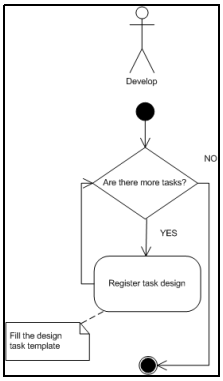
- Testing

Exit

- Design Task

Solution

Process



Development time

- To acquire the necessary knowledge to develop the software product:
- To create the Product Pattern: 45 minutes.
- To apply the Product Pattern: .

Explanatory video

- None

Related Patterns

- Integration Test Pattern
- Modify Integration Pattern
- Generate Code Pattern

Quality Controllers

- None

Templates

Diseño_Tarea.doc

Examples

- None

Support Tools

- It will be necessary a text editor such as OpenOffice Writer o Microsoft Word.
- As well as a tool Visual Paradigm for UMLfor performing the exposed diagrams.

Initial Context

This product can be used in any project in which the philosophy of work in pairs to implement code is followed.

The design creates a structure that organizes the logic of the system, a good design allows the system to grow with changes in one place.

Result Context

Simple designs of small parts of the system are obtained.

Problem

The design in pairs is not that a person type and the other look, it is a dialogue between two people trying to simultaneously schedule, design, analyze and test. Simple design strategy that results in a simple design must be chosen, and must quickly find a way to verify the quality of elaborate design. Uncertainty may arise (When we add more to the designed?), For example:

1. Sometimes the client design eliminates the feature in advance.
2. Sometimes you learn a better way to work when work is being done otherwise.

Designs should be simple, if any part of the system is complex development, it is appropriate to divide it into several .If weaknesses in the design or bad design, they should be corrected soon.

Restrictions (*Forces*)

- **Characteristics of organizations:** This pattern can be used in existing projects in any company.
- **System Type to develop:** This product can be used in projects in which user requirements are changing.
- **Type of Customer:** It must exist or be achieved, the target area development business being involved in achieving it.
- **Heuristics of use:** If you need urgent application or dispose of some of its functionality.

Roles

- Developers (2-12)

Lessons Learned

- Not to be ambitious initially. We should not expect a very high profit initially. Be patient and adding to the design as development proceeds. The design strategy should be:

1. Start with a test, so we know what we are doing.
2. Design and implement just enough for the test to work.
3. Repeat.
4. If you always see the possibility of doing the simplest design, do it.

- Initially designed as simple as possible to be increasing its value.
- The tests file is a generic document for all test types with unit testing box checked, which brings all test cases for a task. This document reflects their responsible and the story associated.
- The Design Task file is a document with the design of the task to implement.

Capability Level

- Not applicable.

Basic Knowledge and Skills

Knowledge

- Knowledge of coding standard that defines the shared code ownership and the rules for writing and documenting code and communication between different pieces of code developed by different teams. Programmers have to follow the so that the code in the system look like if it had been written by one person.
- Knowledge of the common vision of how the program works in which the activities take place.

Abilities

- Ability to work in group. All on an XP computer contribute in any way they can.
- Predicting what will be completed by the deadline, and determining what to do next.
- Programming capability in pairs. Besides to generate better code and tests, used to communicate knowledge through teams.

Information Resources

- 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.