Writing Test Cases

Español

Entries

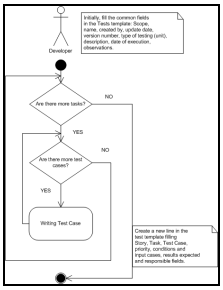
- Story Task version X Developer Y

Exit

- Test.xls

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

- Not applicable

Related Patterns

- Set Balance Pattern
- Unit Testing Pattern

Quality Controllers

- None

Templates

Pruebas.xls

Examples

Test case

Support Tools

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

Initial Context

This product can be used in any projectin which you will have to identify test cases for a particular functionality before implementation. The software features that can not be proven by evidence simply does not exist. Testing provides the opportunity to know if implemented is what actually had in mind. The tests indicate that our work works, when we can not think of any evidence that could cause a failure in our system, then we will finish completely.

Result Context

Developers get a set of test cases related with a functionality specifically to develop.

Problem

Developers should write tests that help to get a program that works and keeps working over time. Programmers write test method by method under the following circumstances:

1. If the interface for a method is not clear, write a test before writing the method.
2. If the interface is clear, but they imagine that implementation could be a little less complicated, write a test before writing the method.
3. If they think on an unusual circumstance in which the code should work as written, write a test to communicate the circumstances.

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 it need urgent application or dispose of some of their funcionalidades.

Roles

- Developers (2-12)

Lessons Learned

- Do not be tempted to underestimate the evidence. Working in pairs of programmers can reduce the above possibility. Tests must be isolated and automated. The test results must be positive or negative without intermediate points. As it is impossible to test all, it be necessary must to write test cases than you think you can fail. Be especially careful with the evidence that work when we had not planned (we should review that test case). Therefore to write test cases we must reflect on the evidence worth doing and what not.
- The output file Test.xls is a generic document for all test types with unit testing box checked, which covers all test cases for a task.
- A developer chooses an assigned task and together with a partner analyze functionality.

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.