

Functional Test Story



Español



Entries

- Testing



Exit

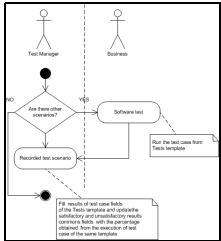
- Testing



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

- [Writing Functional Tests Pattern](#)
- [Writing Test Cases Pattern](#)
- [Change Story in Tasks Pattern](#)
- [Writing Story Pattern](#)



Quality Controllers

- None



Templates

- Pruebas.xls



Examples

- None



Support Tools

- It will need 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 UML](#) for performing exposed diagrams.



Initial Context

This product can be used in any project in which functional test should be performed about the generated software.

From user stories functional tests are made and each may correspond one or more. It is not considered that a story is complete until you pass all your functional tests. The functional tests should be done before you start debugging.



Result Context

Responsible for testing and customers get a glimpse of the percentage of successful test in order to establish the degree of project progress.



Problem

The client should be provided with a tool powerful enough to translate their ideas into a set of scenarios on which to test the software. In addition, the tool must allow the execution and maintenance of the scenarios described by the ideas of the customer.



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 their funcionalidades.



Roles

- Test Manager (1)
- Business Area Users(2 at most)



Lessons Learned

- Not all functional tests have to function at 100% at the same time. The results of functional tests can not be binary, as in the case of unit testing, since they come from different codes generated by different developers team. Over time, correcting the functional tests, the results of approach 100%. As you become closing one version, the client need to classify functional tests to be failing, establishing a priority for correction.
- The client uses the software generated in each of the scenarios described.
- Textual Input document is a generic .xls (Microsoft Excel), for all test types with user box marked tests, which includes the translation of the client's ideas, in the language of the technical field, to tests scenarios of the software generated. In this paper, in the output, percentage of customers satisfaction after performing the corresponding functional tests are added. Also included priority for correction of functional tests that were not satisfactory.



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. 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.
-