

# Integration Test



Español



## Entries

- Testing



## Exit

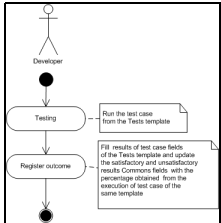
- Testing



## Solution

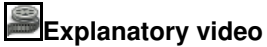


## Process



## 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

- Integrate Pattern
- Modify Integration Pattern
- Design Pattern



## Quality Controllers

- None



## Templates

- Integración\_Continua.doc



## Examples

- None



## Support Tools

- It will need a text editor such as [OpenOffice Writer](#) o [Microsoft Word](#).
- As well as a tool [Visual Paradigm for UML](#) for performing exposed diagrams.



## Initial Context

This product can be used in any project that, once made the job of integrating the individual work of a member of the team, it should be tested in conjunction with the work done by the other team members.

Integration testing is always carried out before adding any new class to the project, or after modifying any existing.



## Result Context

Developers get a result that shows them the ability to integrate their code with other work of their peers. Code duplication and redundancy is verified with the work of other partner.



## Problem

Testing should be done in minutes. Too long integration tests diminish time to the developer to proceed with the development of their work and thus to the identification, formulation and addition of new test cases to integration.



## 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

- Developers (2-12)



## Lessons Learned

- These tests should be performed every few hours (1 day maximum). Whenever you are working on a task, there are hundreds of things in mind. Work continues until there is a natural break (no more elements in the list of tasks) and then integration and its associated tests are done. The cycle must learn / test / program / version.
- The tests file is a generic document for all test types with unit testing box checked, which bind all test cases for a task. We must pay special attention to those cases of unsatisfactory test, it must be modified. In this document their responsible and history associated is reflected. Corresponding to the test results and associated comments field is filled.
- Gotta try together individual work with the work done by the other team members.



## 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://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>
  - Letelier, Patricio y Panadés M<sup>a</sup> 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.
-