



## Initial Context

This product can be used in any project that has been observed that there are programmers with a load factor too high.

## Result Context

Developers get themselves the assess how busy they are in the project.

## Problem

Developers should be able to know if they are too busy or otherwise if they are too lazy.

## 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

- Programmers who commit too much (the result of adding the estimates of the tasks that were assigned to them by their load factor is high compared to that of other team members) should be released some tasks to establish balance in the team work. If whole team is too involved it must be found a way to back into balance. All programmers should have time to do their jobs and to help their peers to make their own.
- Developers should add the estimations of all tasks and multiply by its load factor.
- In the output document Story Tasks version X Developer Y the document header is filled with the balance established by the developer in question.

## 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<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.
-