

06/03/09 - Scrum: 7 sprints y 3 proyectos después

Mi última entrada sobre Scrum, hablaba de la implementación del gráfico Burndown, de esto hace ya casi 3 meses, aunque realmente, comenzamos la andadura en algo antes.

En principio, tengo que tener presente que he estado usando algunas otras metodologías, como la Espiral de Boehm y Metrica-3, con lo que, he podido ver y sufrir *en mis propias carnes*, lo que significa e implica usar una metodología ágil, las ventajas que aporta y lo fáciles que tienden a ser, realmente.

Después de haber realizado siete iteraciones, o *sprints*, y haber completado tres versiones de un proyecto, ya en producción, y otros tres proyectos más casi en producción, puedo dar mis conclusiones acerca de esta metodología de planificación del desarrollo de software:

- Scrum permite el desarrollo ágil, esto es, sobre todo, basándonos en un diseño global, y aplicando un poco la metodología de los prototipos evolutivos, conseguir una versión simple, lo más rápidamente posible y que asiente las bases de lo que será la aplicación futura.
- Las iteraciones cortas, de dos a tres semanas, permiten una supervisión de lo que se va haciendo cada poco tiempo, pudiendo cambiar el rumbo entre *sprint* y *sprint*, para adecuarse a los cambios de requisitos funcionales o cambios de prioridad que puedan surgir a lo largo del desarrollo del programa.
- El gráfico de *burn-down* es una muy útil herramienta que permite tener una visión dinámica de la velocidad con la que se va produciendo el desarrollo y prepararse, de forma anticipada, a posibles variaciones en las fechas de entrega, ya sea resumiendo o eliminando funcionalidad. En caso positivo, si la marcha del proyecto va más rápido, permite agregar otras tareas no planificadas o prioritarias que se pensaban hacer en futuros *sprints*.
- Las tarjetas y el *tablero de juego* que plantea esta metodología, le da una visión lúdica al trabajo y hace que se convierta en un juego. Las tareas se suceden y, todos saben lo que hace cada cual. Incluso los directivos inmediatamente superiores, en lugar de interrumpir el trabajo, con solo mirar el tablero, ya saben cómo va el proyecto.
- El hecho de partir tareas, hace que se puedan paralelizar mejor, con lo que aumenta la potencia del trabajo en equipo, pudiendo hacer varias personas tareas sin solaparse. En esto, el trabajo con

metodologías de tipo Xtreme Programming, programando por parejas, tareas concretas de corto espacio de tiempo y basándonos en el desarrollo orientado a pruebas hace que sea más óptimo.

A todo esto, sumamos el orgullo de poder presentar los proyectos en la fecha pactada, junto con la sensación de que el trabajo se ha realizado en el tiempo adecuado, sin *quemar* a nuestro personal, y tenemos una combinación muy óptima.

¿Qué es lo siguiente? En principio me planteo seguir profundizando en las técnicas de Xtreme Programming para integrarlas en el desarrollo de las tareas individuales que propone Scrum, ya que así son completamente compatibles y muy potentes. Así mismo, adentrarme más aún en los frameworks de pruebas unitarias tanto para PHP, como Ruby, Erlang y C. Espero así, conseguir incluso hasta mejores resultados, ya no hablando de tiempos, sino de calidad de producto.