

No empezar la casa por el tejado

Por: José Luis Pérez (jose.luis.perez@the-mobility-group.com)

Las empresas que se dedican a los proyectos de software a medida, muchas veces se dejan deslumbrar por las bondades de las herramientas CASE que hábilmente presentan los comerciales del sector. Se compran herramientas carísimas que no se amortizan nunca.

Antes, se deberían definir claramente los procesos del ciclo de desarrollo, personalizar unos procesos de acuerdo a la estructura de equipo con la que se consta.

¿ Cuáles son los procesos que yo considero mínimos a contemplar en el ciclo de desarrollo ?

- 1) Control de Requerimientos.** Es necesario registrar cada uno de los requerimientos del sistema a desarrollar en algún tipo de documento, manteniendo un control con los cambios efectuados y enlaces a la documentación generada, esto se puede efectuar con herramientas tipo Excel + Word aunque existen herramientas específicas mucho más sofisticadas dedicadas a este propósito que pueden aumentar la productividad en una segunda etapa posterior cuando esté completamente consolidado la utilización de estas prácticas por parte del equipo de desarrollo.

Beneficios Conseguídos:

- Entendimiento común entre el cliente y el equipo de proyecto de los requerimientos del proyecto.
- Conseguir un acuerdo con el cliente del alcance del proyecto a ejecutar.
- Se mejora el desarrollo de las actividades de diseño
- Mejora de la funcionalidad, integridad durante la etapa de construcción y rendimiento en la ejecución del proyecto.
- Se facilita la etapa de pruebas, desarrollo de interfaces y la mantenibilidad del sistema.
- Mejora en la gestión de proyectos, tenemos controlado perfectamente el alcance y todos aquellos cambios que aparezcan están documentados para poder negociar con el cliente.
- Es una herramienta de comunicación muy buena entre el cliente y el jefe de proyecto.
- Se mejora la satisfacción del cliente y la calidad global, el sistema se comporta de acuerdo a lo que se ha solicitado y documentado.
- Mejora de costes, se reduce la cantidad de trabajo rehecho.

- 2) Planificación del desarrollo iterativo.** Es un hecho probado que nadie utiliza una metodología de desarrollo en cascada, es decir, una fase de análisis, seguida de una fase de diseño, construcción, pruebas e implantación del software.

De una manera más o menos intencionada se trabaja de forma iterativa, refinando el producto final en cada vuelta, no obstante el gran problema que sucede es que las iteraciones no son planificadas, simplemente lo que no se puede acabar se planifica para una futura versión pero de una manera bastante descontrolada.

Beneficios Conseguídos:

- Se consigue mostrar al cliente un progreso sobre el proyecto con mayor frecuencia.
- Se minimiza el impacto ante cambios grandes solicitados.

3) Definición de un plan de pruebas

Para cada iteración planificada se debe diseñar un plan de pruebas, intentando automatizar juegos de prueba unitarios para ejecutar test de regresión (probar todo el sistema entero cada vez que se hace un cambio). En cada iteración se debe ejecutar el plan de pruebas correspondiente y ejecutar un test de regresión para comprobar que las nuevas funcionalidades que se integran no han introducido ningún error en las que ya estaban funcionando.

Beneficios Conseguídos:

- Se consigue mostrar al cliente un progreso sobre el proyecto con mayor frecuencia.
- Se minimiza el impacto ante cambios grandes solicitados.
- Se garantiza la calidad del sistema entregado en base a sus características operativas:
 1. Corrección: ¿Hace lo que se le pide?
 2. Fiabilidad: ¿ Lo hace de forma fiable todo el tiempo ?
 3. Eficiencia: ¿ Se puede ejecutar con una cantidad de recursos hardware razonable ?
 4. Facilidad de uso: ¿ Es de fácil manejo ?

4) Control de cambios

Para cada requerimiento que se haya definido en el sistema se introducirán cambios, por el sencillo motivo que es materialmente imposible conocer el 100% de los requerimientos de un sistema en el momento temporal en el que se efectúa su definición.

En el control de cambios se registrará quien ha efectuado la petición, el motivo, la prioridad, el coste en horas y el beneficio que aporta al proyecto así como una descripción del cambio requerido y a que subsistemas afecta

Beneficios Conseguídos:

- Proporciona un control de proyecto fundamental.
- Permite tener un alcance de proyecto conocido en cualquier momento por el cliente y por la empresa de software.
- El punto anterior nos permite negociar con el cliente al tener controladas todas las nuevas actividades introducidas que estaban fuera del alcance inicial del proyecto.

- Es un proceso básico en el aseguramiento de la calidad, tenemos controlados los cambios fundamentales en el proyecto para garantizar que el software se comporta como se ha dicho que debe comportarse.

Como conclusión, me gustaría señalar que quizá es muy espectacular conseguir un nivel de certificación CMM (*Capability Maturity Model: Modelo de Carnagie Mellon para analizar los niveles de madurez en el desarrollo de software de una empresa*) pero la inversión necesaria para conseguirlo puede ser inabordable para muchísimas empresas.

En este artículo he comentado procesos que se consideran de un nivel 2 de CMM pero yo considero que se deberían practicar en cualquier empresa que se dedique al software como uno de los procesos básicos que deben implantarse desde un inicio.

Un primer paso debería ser estos procesos básicos, recurriendo en un primer momento a herramientas de bajo coste como puede ser StarOffice para la documentación (Excel y Word), WinCVS y CVS para el control de versiones de código y TrackPlus para controlar requerimientos.

Si la empresa tiene mayores necesidades a nivel de la gestión de proyectos ya se introducirán herramientas más complejas y sofisticadas basadas en un repositorio central de requerimientos en torno al cual se organiza todo el desarrollo.