

Trabajo de Fin de Grado

Rubén Agudo Santos

6 de Junio del 2014

Contents

| | | |
|----------|--------------------------------------|-----------|
| 1 | Introducción | 2 |
| 2 | Objetivos del proyecto | 4 |
| 2.1 | Objetivos | 5 |
| 2.2 | Alcance | 5 |
| 2.3 | Planificación temporal | 5 |
| 2.3.1 | Scrum | 5 |
| 2.3.2 | Kanban | 7 |
| 2.4 | Herramientas | 7 |
| 2.5 | Gestión de riesgos | 8 |
| 2.6 | Planificación económica | 8 |
| 2.6.1 | Amortización de material | 9 |
| 2.6.2 | Otros gastos | 9 |
| 2.6.3 | Salarios | 9 |
| 3 | Antecedentes | 10 |
| 4 | Captura de requisitos | 12 |
| 5 | Análisis y diseño | 14 |
| 6 | Verificación y evaluación | 16 |
| 7 | Conclusiones y trabajo futuro | 18 |

Chapter 1

Introducción

Descripción y situación del trabajo, razones de elección del TFG, planteamiento del problema y justificación del TFG

Chapter 2

Objetivos del proyecto

2.1 Objetivos

Los objetivos del proyecto consisten en crear un software que extienda la funcionalidad actual del sistema Intrasim desarrollado por la universidad de Deusto y del grupo de investigación Galan perteneciente a la UPV/EHU.

Es un software que permite gráficamente visualizar las observaciones¹ del sistema. El objetivo principal es brindar al experto una herramienta para poder señalar gráficamente en que momento de esa observación sucede el hecho concreto que estaban buscando.

La herramienta permite visualizar los datos en bruto tanto en modo vídeo (si esta disponible), como en modo gráfico, con polilíneas, histogramas etc. Tales gráficos permiten la selección de rangos.

2.2 Alcance

Aquí va el alcance del proyecto.

2.3 Planificación temporal

En vez de seguir una planificación clásica, como puede ser COCOMO² o Waterfall³ he decidido utilizar las metodologías ágiles de desarrollo, concretamente Scrum y Kanban

2.3.1 Scrum

Scrum es una serie de herramientas (framework) para la gestión y desarrollo de software basado en un proceso iterativo e incremental. *cita a la wikipedia*

¹ Aquí va una explicación de lo que es una observación

² <https://en.wikipedia.org/wiki/COCOMO>

³ https://en.wikipedia.org/wiki/Waterfall_model

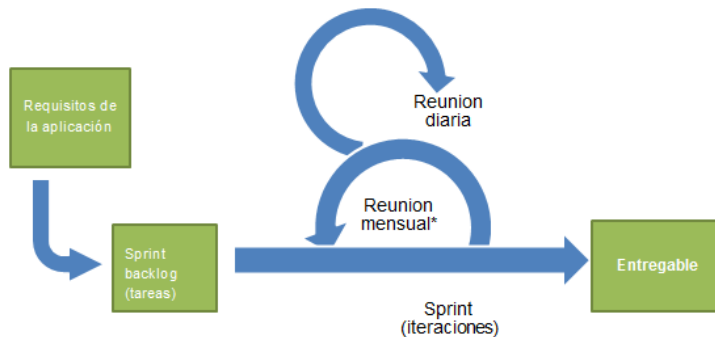


Figure 2.1: En este gráfico podemos ver el método de trabajo según Scrum, en el cual se observa lo importante que son los entregables

Scrum, de manera resumida consiste en lo siguiente:

1. Hablar con el cliente para ver si hay nuevos requisitos.
2. Crear backlog, es decir, las tareas de ese sprint.
3. Programar los requisitos especificados en el sprint.
4. Reunión de retrospectiva para ver que ha ido mal y mejorarlo.
5. Enviar entregable al cliente.
6. Repetir.

Es fundamental entre sprint y sprint que el valor del producto haya aumentado. Es decir, lo importante es el producto, no como vamos nosotros en el propio proyecto. Podríamos haber avanzado mucho en el diseño del software, pero si eso es algo que el cliente no puede ver estaremos fallando en nuestra agilidad.

Por otro lado, si cometemos algún error será muy sencillo corregir el error ya que si los sprints son de dos semanas por ejemplo, no habremos perdido apenas tiempo. Y además para el cliente los entregables serán predecibles en el tiempo.

Scrum es también muy importante, y se posiciona como una alternativa muy potente frente a COCOMO o waterfall, es porque en la creación de software hay un componente de incertidumbre muy grande. No sabemos como, ni cuando pueden cambiar los requisitos de un software. Por ello toman importancia los sprints de nuevo.

En los modelos antiguos, si ya habíamos terminado la fase de análisis y diseño, y se requería una nueva funcionalidad es necesario paralizar el desarrollo del software y volver a analizar y diseñar. En cambio, con Scrum es posible integrar ese cambio en el siguiente sprint.

2.3.2 Kanban

Kanban es un método de organización del conocimiento del trabajo que se está realizando con un gran énfasis en el la entrega justo a tiempo ⁴ sin sobrecargar al equipo *Cita a la wikipedia*

Kanban se centra mucho en que lo importante no es empezar muchas cosas, si no que acabemos aquello que empezamos. Sirve sobre todo para ver de una manera visual:

- Que tenemos pendiente
- Que estamos haciendo ahora mismo
- Que hemos terminado.

El método kanban es algo muy extenso, pero yo simplemente he aplicado el tablero Kanban mezclado con Scrum. Cada dos semanas, me creo una lista de tareas que debo completar (Backlog) y lo coloco en la columna de "Tareas pendientes", ordenadas por prioridad. Después, coloco en la lista de "En progreso" como máximo dos tareas. Y no empiezo ninguna otra hasta que esas dos hayan sido pasadas a la columna de "Tareas finalizadas".

2.4 Herramientas

El proyecto, al no ser enteramente mío, no he tenido una libertad total para la elección de herramientas.

Las herramientas que he utilizado para llevar a cabo este proyecto han sido:

- **Visual Studio**

Para el desarrollo propiamente dicho del proyecto he usado Visual Studio 2013 Ultimate, proporcionado de manera gratuita gracias al acuerdo que mantiene la UPV/EHU con Microsoft.

⁴JIT Delivery: Just-In-Time Delivery

- **TeXstudio y L^AT_EX**

Dos herramientas gratuitas y de código abierto para generar documentos.

- **Git y GitHub**

Fundamental en los proyectos que se basen en metodologías ágiles. Git es un software de gestión de control de versiones distribuido, esto es, que cada desarrollador dispone de una copia completa del repositorio, desarrollado por Linus Torvalds y Junio Hamano en 2005 y similar a SVN.

2.5 Gestión de riesgos

Aquí va la gestión de riesgos

2.6 Planificación económica

Nuestra empresa la forman:

- **Ángel Agudo:** CEO ⁵ de la empresa.
- **Asun Santos:** Directora de calidad y administración.
- **Elena Agudo:** Directora de arte.
- **Rubén Agudo:** CTO ⁶ y responsable de I+D.

Cómo es una epoca complicada por la crisis, tanto Ángel como Asun se han visto en la necesidad de pluriemplearse como administrativos de Osakidetza.

Su empleo en Osakidetza reportan a la empresa familiar una cantidad neta de 5000€ mensuales. Nuestro único proyecto de software es este proyecto de fin de carrera por lo que todos los ingresos provienen de sus otros empleos.

Como somos una empresa muy concienciada con la investigación y el desarrollo hemos decidido que el proyecto sea gratis y de código abierto, licenciado bajo GPLv3.

Por lo tanto el coste de venta de nuestro coste sera 0€

⁵Chief Executive Officer: Director ejecutivo

⁶Chief Technical Officer: Director de tecnología

2.6.1 Amortización de material

Oficina

La oficina que utilizamos es el domicilio familiar y no hay que pagar alquiler ya que es de nuestra propiedad. Por lo que el gasto de alquiler/hipoteca es 0€

Material informático

El material informático que se ha usado ha sido un portátil Acer TravelMate 5742G comprado en Diciembre de 2010 y usado intensivamente durante estos 4 años. El portátil estaba pensado para ser amortizado en 4 años. *ASI que por aqui habra que hacer algun calculo tonto*

2.6.2 Otros gastos

| Concepto | Gastos en € / mes |
|---------------------|-------------------|
| Luz | 60 |
| Agua | 50 |
| Internet + Teléfono | 60 |
| Gas | 70 |
| Seguro | 45 |
| IBI | 40 |
| Total Mes | 325 |
| Total Año | 3900 |

2.6.3 Salarios

De los 5000€ que se ingresan mensualmente se reparten de la siguiente manera:

| Miembro de la empresa | Salario en € |
|-----------------------|--------------|
| Ángel Agudo | 1000 |
| Ásun Santos | 1000 |
| Elena Agudo | 80 |
| Rubén Agudo | 80 |
| Remanente Empresa | 2840 |

Chapter 3

Antecedentes

Situación actual, estudio de diferentes alternativas existentes o distintos posibles enfoques del problema a solucionar

Chapter 4

Captura de requisitos

Que tiene que cumplir el trabajo que se va a desarrollar, que necesidades tiene que satisfacer, quienes van a ser sus usuarios, etc.

En un trabajo de desarrollo de software tradicional se documentará mediante el Modelo de casos de uso y la jerarquía de actores, acompañando cada caso de uso y cada actor de una pequeña definición.

Chapter 5

Análisis y diseño

Cómo se plantea el desarrollo del trabajo. En que partes se divide la solución, justificando dicha división y explicando cada una de las partes.

En aquellos trabajos donde sea aplicable se documentará usando el modelo de dominio, el diagrama de clases, los módulos/paquetes del software desarrollado, etc.

Chapter 6

Verificación y evaluación

Definición del trabajo desarrollado usando un lenguaje natural. Se debe dar respuesta a cuestiones como ¿Qué se ha hecho? ¿Cómo se ha hecho? Cambien se explicara que problemas se han encontrado durante el desarrollo del trabajo, que soluciones se han estudiado, cual fue la elegida etc. Todo ello justificando las decisiones tomadas.

Chapter 7

Conclusiones y trabajo futuro

Resumen del trabajo elaborado y de los principales hitos conseguidos.

Conclusiones de la gestión: explicación y justificación de los cambios respecto a la propuesta original, planificación, costes, objetivos.

Conclusiones desde un punto de vista crítico: Utilidad del trabajo realizado, problemas durante la realización del TFG, opiniones de usuarios, comparativas con otras soluciones.

Posibles mejoras y ampliaciones a lo realizado con una pequeña estimación del trabajo que supondría afrontarlas y una valoración sobre las mismas. Aspectos relacionados con el mantenimiento del trabajo desarrollado.

Bibliography

- [1] Robert Bringhurst, *The elements of typographic style*, Hartley & Marks, 1992.
- [2] L. Brunetti, M. Coisson, P. Tiberto, and F. Vinai, *Magneto-impedance measurements in amorphous Co-based magnetic wires at high frequency*, Journal of Magnetism and Magnetic Materials **249** (2002), no. 1, 310--314.
- [3] Pabsy González Irisarri, *Los peligros de cometer plagio*, Universia **10** (2010/10/5).
- [4] André Miede, *A classic thesis style*, 2011.
- [5] Lorenzo Pantieri, *Introduzione allo stile* , 2011, in Italian.
- [6] Lorenzo Pantieri and Tommaso Gordini, *L'arte di scrivere con* , 2012, in Italian.