

Ingeniería del Software:

Proyecto SIGREM v.2

SIGREM: Sistema Integral de Gestión de Recursos de Multas

Autores:

- Félix Martos Trenado
- Javier Rodríguez Horcajo
- Miguel Ángel Ruiz Ortega
- Francisco José Sánchez Hernando
- Sergio Tarancón Faus

ÍNDICE

1. Introducción

- 1.1 Propósito del plan
- 1.2 Ámbito del proyecto y objetivos
 - 1.2.1 Declaración del ámbito
 - 1.2.2 Funciones principales
 - 1.2.3 Aspectos de rendimiento
 - 1.2.4 Restricciones y técnicas de gestión
- 1.3 Modelo de proceso

2. Estimaciones del proceso

- 2.1 Datos históricos
- 2.2 Técnicas de estimación
- 2.3 Estimaciones de esfuerzo, coste y duración

3. Estrategia de gestión del riesgo

3.1 Análisis del riesgo

3.2 Estudio de los riesgos

3.3 Plan de gestión del riesgo

4. Planificación temporal

4.1 Estructura de descomposición del trabajo

4.2 Gráfico Gantt

4.3 Red de tareas

4.4 Tabla de uso de recursos

5. Recursos del proyecto

5.1 Personal

5.2 Hardware y software

5.3 Lista de recursos

6. Organización del personal

6.1 Estructura de equipo

6.2 Informes de gestión

7. Mecanismos de seguimiento y control

7.1 Garantía de calidad y control

7.2 Gestión y control de cambios

1. Introducción

1.1 Propósito del plan

El propósito de este Plan de Proyecto es detallar todos los aspectos relativos a la planificación para desarrollar SIGREM (*Sistema Integral de Gestión de REcursos de Multas*).

1.2 Ámbito del proyecto y objetivos

1.2.1 Declaración del ámbito

Este proyecto se encuadra dentro de las actividades de la asignatura de Ingeniería del Software del 4º Curso de la carrera de Ingeniería Informática. El objetivo es la planificación, diseño y desarrollo de un proyecto software de dimensión industrial usando técnicas actuales de desarrollo de software.

El software se integrará en una empresa de gestión de recursos de multas. Al utilizarlo todos los empleados, tanto administrativos como otros de la empresa debe ser de manejo intuitivo.

1.2.2 Funciones principales

El propósito del diseño y desarrollo de SIGREM es crear un producto que satisfaga las necesidades de una empresa de gestión de recursos de

multas de tráfico, cuyo principal servicio será prestar asesoramiento jurídico a sus clientes. Éstos podrán crear contratos con la empresa que harán referencia a un vehículo sobre el que pesa alguna multa, y a su vez podrán solicitar un recurso para intentar anularla. La empresa asignará entonces un abogado de su plantilla a dicho recurso para intentar resolverlo favorablemente.

Las funcionalidades prestadas por el producto desarrollado son las siguientes:

- Permitir la creación, modificación consulta y cancelación de los contratos de los clientes
- Añadir, modificar, consultar y eliminar multas a los contratos
- Facilitar la asignación de los abogados contratados por la empresa a los recursos interpuestos por los clientes sobre sus multas
- Añadir, consultar, modificar y eliminar recursos a las multas asociadas a los contratos de los clientes
- Agilizar el proceso de facturación de la empresa en relación a los contratos establecidos con los clientes
- Controlar la gestión de nóminas de los empleados
- Consultar y modificar los datos de los empleados de la empresa y de los clientes
- Gestionar la plantilla de la empresa

1.2.3 Aspectos de rendimiento

En la versión monousuario del producto, habrá un único terminal de comunicación con la aplicación, y un único usuario interactuando con ella. Además, la persistencia de los datos será consistente y fiable, al tiempo que se garantizará su seguridad y confidencialidad. La respuesta de la aplicación a las peticiones del usuario será inmediata.

En la versión web:

- El sistema deberá soportar la carga simultánea generada por un porcentaje de usuarios estimado en su momento por la empresa gestora, que dependerá del número de clientes y de los casos que lleve cada día.
- El sistema completo deberá correr sobre un servidor de características medias.

1.2.4 Restricciones y técnicas de gestión

En la versión web el sistema basará sus comunicaciones en protocolos estándar de Internet y los servidores deben ser capaces de atender consultas concurrentemente.

Los distintos subsistemas deberán tener un diseño e implementación sencillos, independientes de la plataforma o el lenguaje de programación.

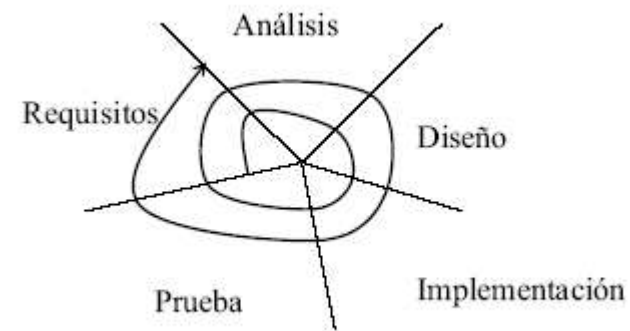
1.3 Modelo de proceso

Hemos elegido como **modelo de proceso** el llamado *Proceso Unificado de Desarrollo (RUP)*

Este modelo de proceso está compuesto por las siguientes actividades estructurales:

- Requisitos
- Análisis
- Diseño
- Implementación
- Prueba

Y puede representarse mediante el siguiente esquema:



Descripción del Modelo de Proceso:

- Cada vuelta en la espiral se denomina iteración.
- La agrupación de iteraciones se denomina fase.
- La agrupación de fases se denomina ciclo, que termina con una versión del producto.
- Tiene las siguientes fases:
 - Fase de inicio: se desarrolla una descripción del producto final. El trabajo se centra en los **Requisitos**.
 - Fase de elaboración: se especifican los casos de uso y se diseña la arquitectura del sistema. Se trabaja en **Análisis y Diseño**.
 - Fase de construcción: se crea el producto. El trabajo se centra en la **Implementación**.
 - Fase de transición: el producto se convierte en versión beta. El trabajo se centra en **Prueba**.

2. Estimaciones del proceso

2.1 Datos históricos

No contamos con datos históricos de proyectos anteriores, y este grupo nunca ha trabajado como grupo de desarrollo anteriormente. Por tanto no se utilizarán datos previos para la estimación del esfuerzo del proceso.

2.2 Técnicas de estimación

Para la estimación del esfuerzo se ha utilizado la técnica de descomposición del proceso. Se identifica un conjunto pequeño de actividades o tareas de trabajo y se estima el esfuerzo requerido para llevar a cabo cada tarea, de forma proporcional a los requisitos funcionales exigidos. Estas tareas se denominan *Módulos*

2.3 Estimaciones de esfuerzo, coste y duración

Comenzamos con la descomposición del problema en los siguientes módulos, y estimaremos el esfuerzo basándonos, fundamentalmente en los requisitos funcionales de cada uno de ellos:

MED: Estructuras de datos

MIG: Interfaz gráfica

MCL: Gestión de clientes (6 requisitos)

MCO: Gestión de contratos (5 requisitos)

MMU: Gestión de multas (6 requisitos)

MRE: Gestión de recursos (4 requisitos)

MEM: Gestión de empleados (6 requisitos)

MEC: Gestión económica (3 requisitos)

Además se añade una tarea *Proyecto* para controlar la evolución del trabajo. Esta tarea se reparte en tres puntos claves del Proyecto:

- Al comienzo, para realizar el análisis y diseño general del software
- Tras la realización de los tres primeros módulos base (Estructuras de Datos, Interfaz Gráfica y Clientes), en que se efectúa su ensamblaje
- Y al finalizar la codificación y prueba de todos los módulos restantes, para ensamblar y probar el código completo del Proyecto.

Tabla de estimación del esfuerzo

| A.E. | Requisitos | Análisis | Diseño | | Implementación | | Prueba | Esfuerzo |
|-----------------|------------|----------|-----------|------------|----------------|------------|-----------|---------------|
| Tarea | | | Diseño | Evaluación | Codificación | Ensamblaje | | |
| Proyecto | | 5 | 10 | 0,5 | | | | 15,5 |
| MED | | | 6 | | 12 | | 4 | 22 |
| MIG | | | 6 | | 12 | | 4 | 22 |
| MCL | 6 | | 2 | | 6 | | 2 | 10 |
| Proyecto | | | | 0,5 | | 5 | | 5,5 |
| MCO | 5 | | 2 | | 6 | 2 | 4 | 14 |
| MMU | 6 | | 2 | | 6 | 2 | 4 | 14 |
| MRE | 4 | | 1 | 0,5 | 4 | 2 | 4 | 11,5 |
| MEM | 6 | | 2 | | 4 | 2 | 4 | 12 |
| MEC | 3 | | 2 | | 4 | 2 | 4 | 12 |
| Proyecto | | | | 0,5 | | 5 | 15 | 20,5 |
| Esfuerzo | | 5 | 33 | 2 | 54 | 20 | 45 | 159 pd |

El esfuerzo para el desarrollo del proceso se estima en 159 personas/día en total. Para hacer este cálculo hemos planificado que cada uno de los miembros tiene el siguiente horario: De Lunes a Sábado se trabaja 1 hora y 20 minutos. Por tanto se trabajan 8 horas a la semana, durante todo el tiempo que dure el proceso.

4. Planificación temporal

4.1 Estructura de descomposición del trabajo

| A.E. | Requisitos | Análisis | Diseño | | Implementación | | Prueba |
|----------|------------|---|---|---|--------------------------|---|--------------------------|
| Tarea | | | Diseño | Evaluación | Codificación | Ensamblaje | |
| Proyecto | | D1 Sergio, Javi, Frank, Félix, Miguel | D2-D3 Sergio,Javi, Félix Miguel, Frank, | D4 Javi (50%) Evaluación diseño proyecto | | | |
| MED | | | D5-D7 Sergio, Frank | | D9-D14 Sergio, Frank | | D16-D17 Sergio, Frank |
| MIG | | | D5-D7 Miguel, Javi | | D9-D14 Javi, Miguel | | D16-D17 Javi, Miguel |
| MCL | 6 | | D5-D6 Félix | | D9-D14 Félix | | D16-D17 Félix |
| Proyecto | | | | D8 Frank (50%) Evaluación diseño MED, MIG, MCL | | D15 Sergio, Javi, Miguel, Frank, Félix | |
| MCO | 5 | | D18 Frank, Sergio | | D20-D22 Frank, Sergio | D23 Frank, Sergio | D24-D25 Frank, Sergio |
| MMU | 6 | | D18 Félix, Javi | | D20-D22 Félix, Javi | D23 Félix, Javi | D24-D25 Félix, Javi |
| MRE | 4 | | D18 Miguel | D19 Sergio (50%) Evaluación diseño MCO, MMU, MRE | D20-D23 Miguel | D24-D25 Miguel | D26-D29 Miguel |

| A.E. | Requisitos | Análisis | Diseño | | Implementación | | Prueba |
|----------|------------|----------|----------------------|---|--------------------------|---|---|
| MEM | 6 | | D26 Frank, Sergio | | D28-D29 Frank, Sergio | D30 Frank, Sergio | D31-D32 Frank, Sergio |
| MEC | 3 | | D26 Javi, Félix | | D28-D29 Javi, Félix | D30 Javi, Félix | D31-D32 Javi, Félix |
| Proyecto | | | | D27 Félix (50%) Evaluación diseño MEM, MEC | | D33 Javi, Félix, Frank, Sergio, Miguel | D34-D36 Javi, Félix, Frank, Sergio, Miguel |

4.2 Gráfico Gantt

Mirar Apéndice A

4.3 Red de tareas

Mirar Apéndice B

5. Recursos del proyecto

5.1 Personal

El proyecto cuenta con un personal de 5 miembros.

5.2 Hardware y software

Cada uno de los miembros dispone de un ordenador personal para realizar las tareas. En momentos determinados se hará uso también de los equipos disponibles en los laboratorios de la Facultad de Informática.

La intención del equipo de desarrollo es utilizar, siempre que sea posible, software libre.

Tres de los miembros utilizarán como Sistema Operativo Mandrakelinux 10.1 Oficial y los otros dos Microsoft Windows XP.

El entorno de desarrollo utilizado es el mismo para todos, Eclipse 3.0.1. Utilizando simultáneamente distintas versiones del SDK de Java, 1.4 y 1.5, para asegurar la compatibilidad.

Para la documentación se ha utilizado OpenOffice 1.1.4, que permite el manejo de ficheros de Microsoft Word, como exige el profesor, además de permitir la creación de PDFs.

Para el diseño UML, por exigencias externas, se ha utilizado Rational

Rose, aunque también se ha aprovechado el soporte de este formato por Visual Paradigm, que permite ser usado en distintas plataformas y es gratuito para uso personal.

5.3 Lista de recursos

- Lista de correo para coordinar el desarrollo (Mailman)
- Servidor web con un wiki (DokuWiki) para coordinar el proceso de documentación
- Servidor CVS para coordinar el código (Berlios.de)

6. Organización del personal

6.1 Estructura de equipo

El equipo se organiza con un sistema Democrático Descentralizado (DD). La comunicación se produce mediante la lista de correo, mientras que la sincronización del código entre los equipos temporales que se establecen se realiza con el servidor CVS. De este modo todos los miembros del equipo tienen siempre la última versión del código.

7. Mecanismos de seguimiento y control

7.1 Garantía de calidad y control

Para garantizar la calidad del software desarrollado se utilizan técnicas de Ingeniería del Software. En el proceso de desarrollo se han establecido dos tipos de actuaciones. La primera de ellas es el establecimiento de una serie de tareas de prueba de cada uno de los módulos desarrollados. Además se establecen hitos tras las pruebas de los módulos, momento en que se produce la integración y prueba de los módulos en su conjunto.

7.2 Gestión y control de cambios

Los documentos de planificación y especificación del software están sometidos a un proceso de revisión continua de acuerdo con el cliente.

El código se almacena en un sistema de control de versiones (CVS) que almacena los cambios, de modo que pueden revisarse de forma continua.