

Proyecto: Gestión de un restaurante
Plan del proyecto del software
(Formato Pressman)

Miembros del equipo:

Laura Pérez	Esther Ávila
Miguel Ascanio	Guillermo Romero
Javier Toledano	Daniel Tocino
Álvaro Miguel Rodríguez	David Quirce

Control de cambios

[illegible]

Contenido

1	Introducción	5
1.1	Propósito del plan.....	5
1.2	Ámbito del proyecto y objetivos.....	5
1.2.1	Declaración del ámbito	5
1.2.2	Funciones principales.....	5
1.2.3	Aspectos de rendimiento	5
1.2.4	Restricciones y técnicas de gestión	6
1.3	Modelo de proceso.....	6
2	Estimaciones del proyecto.....	7
2.1	Datos históricos.....	7
2.2	Técnicas de estimación.....	7
2.3	Estimaciones de esfuerzo, coste y duración	7
3	Estrategia de gestión del riesgo	9
3.1	Análisis del riesgo	9
3.2	Estudio de los riesgos.....	9
3.3	Plan de gestión del riesgo	10
4	Planificación temporal	11
4.1	Estructura de descomposición del trabajo o Planificación temporal	11
4.2	Gráfico Gantt	11
4.3	Red de tareas.....	11
4.4	Tabla de uso de recursos	11
5	Recursos del proyecto	12
5.1	Personal	12
5.2	Hardware y software	12
5.3	Lista de recursos	12
6	Organización del personal	13
6.1	Estructura de equipo (si procede)	13
6.2	Informes de gestión	14
7	Mecanismos de seguimiento y control	15
7.1	Garantía de calidad y control	15
7.2	Gestión y control de cambios.....	15
7.3	Introducción.....	15
7.3.1	Propósito.....	15
7.3.2	Alcance	15
7.3.3	Definiciones.....	15
7.3.4	Referencias	15

7.4 Tipos de artefactos a gestionar (los ECSs)	16
7.5 Criterios y protocolos para nombrar los ECSs	17
7.5.1 ECSs dependientes de CU	17
7.5.2 Archivos de Código.....	17
7.5.3 Documentos de Proyecto.....	17
7.6 Quién es responsable de los procedimientos de GCS y de la creación de Líneas Base.....	17
7.7 Políticas para el Control de Cambios y la Gestión de Versiones.....	18
7.7.1 Nota importante sobre cambios leves	18
7.7.2 Clasificaciones del cambio	18
7.7.3 Dificultad	18
7.7.4 Proceso de notificación.....	18
7.7.5 Proceso de cambio	19
7.7.6 Gestión de versiones	19
7.8 Registros para mantener el rastro de los cambios	20
Registro general.....	20
7.8.1 Registro propio de cada ECS.....	20

1 Introducción

1.1 Propósito del plan

Con la realización de este plan de proyecto se pretende conseguir una estimación fiable de los costes, tanto en tiempo como en personal para realizar el software en los plazos acordados.

1.2 Ámbito del proyecto y objetivos

1.2.1 Declaración del ámbito

En un restaurante de comida rápida o FastFood, el funcionamiento habitual en el trabajo consiste en tomar el pedido, cobrarlo, emitir dos tickets, uno para el cliente y otro para la cocina y rápidamente tener el pedido listo para entregar.

Al ser un negocio se debe llevar un registro de ventas, hacer arqueos a diario o entre turno de distinto trabajadores y en caso de necesidad poder meter o sacar dinero de la caja para tener cambio o eliminar exceso de algún tipo de billete o moneda.

1.2.2 Funciones principales

Contabilidad

Llevar un balance con las ventas y los descuadres, saber qué se ha vendido más o quién ha vendido más son algunos de los datos más importantes para el negocio, el gestionarlo de forma correcta y fácil es vital.

Control de usuarios

Una caja registradora la usa mucha gente dentro de la tienda, con distintos niveles de responsabilidad. El gestionar de forma correcta quién se puede meter en cada función del programa supone un reto a superar para poder tener una aplicación robusta y segura.

Se pueden crear usuarios con distintos niveles de acceso, eliminarlos o cambiarles el rol para aumentar o disminuir su acceso al programa. Además se realizarán backups por seguridad de todos estos datos.

Carta

Generar los productos que se van a vender, el precio que tienen y poder editar los mismos. Además la aplicación permite crear backups de la carta y cargarlos, por seguridad ante posibles ampliaciones del programa. Dándose el caso de varios restaurantes que tienen la misma carta sería un desastre perderla y no poderla recuperar.

Pago y uso de la caja

Registrar las ventas, cobrar con distintos métodos de pago y llevar de forma correcta las retiradas y la introducción de cambio en la caja es imprescindible para poder trabajar correctamente vendiendo lo que sea en nuestro restaurante.

1.2.3 Aspectos de rendimiento

Nuestra aplicación en un principio no necesita internet, por lo tanto la conexión con los archivos, la carga y descarga de backups se debería de realizar en tiempos muy cortos.

En cuanto al uso habitual, el que más horas va estar funcionando es el del pago, por lo que debería ser una aplicación rápida hasta tal punto que los tiempos de ejecución sean imperceptibles para cualquier usuario.

1.2.4 Restricciones y técnicas de gestión

Esta aplicación se hará para Windows y con Java, por tanto la portabilidad a otros TPV deberá ser única y exclusivamente bajo estas premisas, en cuanto a internet no lo vamos a utilizar, por lo tanto no afectará en primera instancia a la aplicación.

1.3 Modelo de proceso

En cuanto al modelo del proceso, se está utilizando un modelo evolutivo dentro del Proceso Unificado de Desarrollo Software, el cual está dirigido por los casos de uso, en particular la Espiral Boehm. Sin haber creado todavía ningún prototipo estamos de forma incremental analizando el problema a resolver, generando los casos de uso, revisándolos, corrigiéndolos y después volviendo a analizar para generar más casos de uso.

PUNTOS DE FUNCION	COMPLEJIDAD						TOTAL
	SIMPLE		MEDIA		COMPLEJA		
	#N	PESO	#N	PESO	#N	PESO	
ENTRADAS	15	3	2	4	0	6	53
SALIDAS	5	4	3	5	0	7	35
CONSULTAS DEL USUARIO	2	3	2	4	0	6	14
FICHEROS LOGICOS	5	7	0	10	0	15	35
INTERFACES EXTERNAS	0	5	0	7	0	10	0
TOTAL DE PUNTOS DE FUNCION SIN AJUSTAR							137

Para ajustar estas estimaciones se contemplan 14 factores de complejidad que se detallan a continuación:

ID	FACTOR DE COMPLEJIDAD	TOTAL
1	COMUNICACIÓN DE DATOS	0
2	PROCESO DISTRIBUIDO	0
3	OBJETIVOS DE RENDIMIENTO	1
4	INTEGRACIÓN DE LA APLICACIÓN	1
5	TASA DE TRANSACCIONES	3
6	ENTRADA DE DATOS INTERACTIVA	0
7	EFICIENCIA PARA EL USUARIO FINAL	3
8	ACTUALIZACIONES INTERACTIVAS	3
9	LOGICA DEL PROCESO INTERNA COMPLEJA	3
10	REUSABILIDAD DEL CÓDIGO	0
11	CONVERSIÓN E INTALACIÓN	0
12	FACILIDAD DE OPERACIÓN	5
13	INSTALACIONES MÚLTIPLES	1
14	FACILIDAD DE CAMBIOS	0
	FACTOR DE COMPLEJIDAD TOTAL	20

FC1: 0, FC2: 0, FC3: 1, FC4: 1, FC5: 3, FC6: 0, FC7: 3, FC8: 3, FC9: 3, FC10: 0, FC11: 0, FC12: 5, FC13: 1. FC14: 0. La suma total asciende a 20, si nos fijamos un poco más que FC12 asciende a 5, esto se debe a que tiene que poder generar una serie de copias de seguridad de forma automática para su correcto funcionamiento.

Con estos datos en mente y aplicando una fórmula sencilla podremos calcular los puntos de función ajustados que nos ayudarán a estimar mejor nuestro proyecto.

Puntos de función ajustados = Puntos de función sin ajustar * (0.65 + (0.01 * Factor Complejidad total)).

Al realizar el cálculo obtenemos que los puntos de función ajustados son 116.45.

En cuanto al coste y la duración, el coste en horas será el necesario para cumplir con los plazos de entrega exigidos con la asignatura. Calculamos el proyecto en un total de 28 semanas por la duración del curso, vistos los puntos de función ajustados, basándonos en datos de proyectos anteriores y poniéndonos en el peor de los casos con una equivalencia de 6 horas por punto de función se hace un total de 698.7 horas para realizar el proyecto. Dicho de otra forma, si no hay retraso y todo va según lo previsto con unas 25 horas de trabajo semanal el proyecto saldrá adelante.

3 Estrategia de gestión del riesgo

3.1 Análisis del riesgo

Nos enfrentamos a una serie de riesgos que se describen a continuación. Estos riesgos se clasifican por coste, en nuestro caso tiempo perdido, del proyecto, del cliente o negocio y técnicos:

- **Riesgo de desarrollo (Técnico):** dado que no todo el equipo sabe programar todavía en Java, que los miembros que saben no compartan la misma visión del programa o simplemente por una falta de entendimiento puede darse el caso de que al programar cada uno su módulo por separado luego no coincidan y haya que empezar de nuevo. La probabilidad es media ya que el equipo de programación es reducido, pero el impacto siendo pesimistas, puede ser intolerable dado que a falta de poco para entregar puede darse algún fallo que nos retrase tanto que nos quedemos fuera de plazo.
- **Riesgo de falta de presupuesto (Coste):** debido a que el tiempo es oro y que somos un equipo un tanto dispar, la probabilidad de quedarnos sin tiempo para hacer alguna entrega a tiempo es muy alta. El impacto es intolerable dado que cuesta mucho remontar un retraso.
- **Riesgo de contrato (Técnico):** el SRS es el contrato con el cual el equipo se compromete a hacer y el cliente sabe lo que va a obtener. En este caso el cliente tiene un amplio conocimiento de cómo son estos contratos y por tanto la probabilidad de que haya que rehacer algunas partes o agregar o quitar funcionalidades es alta y el impacto sería alto.
- **Riesgo de abandono (Proyecto):** dado que cada integrante del equipo tiene una carga lectiva distinta y además le da una importancia distinta a la asignatura, la posibilidad de que un miembro renuncie al equipo o haya que despedirle del mismo debido a una falta reiterativa de trabajo es muy alta, y el impacto es alto.
- **Riesgo de comunicación (Proyecto):** debido a que somos un grupo a veces se da el caso de no entendernos o de explicarnos mal. Esto pasa más a menudo de lo que parece, por tanto hay una probabilidad media de que ocurra y el impacto es alto, ya que si esto ocurre el proyecto no avanza.
- **Riesgo planificación (Proyecto):** la mala estimación del trabajo a entregar y que por lo tanto las fases de desarrollo sean más lentas de lo previsto tiene un impacto medio y de probabilidad alta.
- **Riesgo de ámbito (Negocio):** dado que hemos elegido una aplicación de uso para el comercio con un enfoque muy concreto, puede ocurrir que no tengamos en cuenta determinados aspectos que hagan que el trabajo al final crezca o peor aún que lo ya realizado no sirva. La probabilidad de que esto ocurra es media dado que solamente un miembro del grupo conoce el funcionamiento de un TPV y además no es del mismo tipo de comercio. El impacto ha sido bajo hasta ahora ya que la idea general está bien consolidada y son los pequeños detalles lo que falla.

3.2 Estudio de los riesgos

Se han detectado una serie de riesgos en el proyecto y se ha estudiado la probabilidad y el impacto en el mismo basándonos en la experiencia, la intuición y el sentido común.

Siempre habrá riesgos que no podamos predecir o ver venir, pero aquellos que ya tenemos previstos podemos tenerlos en cuenta a la hora de planificarnos con el fin de poderlos evitar y si ya ha ocurrido subsanar a tiempo.

3.3 Plan de gestión del riesgo

Ya hemos estimado los riesgos que puede surgir, ahora vamos a ir uno por uno analizando las posibles soluciones:

- **Riesgo de contrato (Técnico):** dado que la probabilidad de que ocurra es alta y las consecuencias son graves estamos procurando realizar una labor preventiva, consultando todas las fuentes posibles para evitar el fallo. En caso de ocurrir se repartirá la carga de trabajo en función del responsable del fallo, de tal modo que desde que el fallo se detecta hasta la siguiente reunión el responsable del fallo tendrá como prioridad corregirlo y su carga de trabajo respecto a nuevos objetivos será menor.
- **Riesgo de ámbito (Negocio):** los fallos que pueden surgir en el ámbito no son de carácter grave por tanto como norma, si es un miembro del grupo el que detecta el fallo lo comunica y lo corrige en el acto, de no poder hacerlo se asignaran más horas de trabajo.
- **Riesgo de desarrollo (Técnico):** todavía no hemos planteado como queremos hacerlo, ni sabemos muy bien nada del segundo cuatrimestre. Por tanto y hasta más información la única medida que se tomará será, en caso de que se pueda, que el equipo de programación sólo se dedique a programar el último mes y medio aproximadamente.
- **Riesgo planificación (Proyecto):** la única forma de subsanar esto es retocar la planificación y asumir la carga entre todos los miembros del equipo.
- **Riesgo de falta de presupuesto (Coste):** se pueden hacer dos cosas en este caso, dado que el impacto para el proyecto sería enorme si el retraso fuera muy grande, se podría intentar sacar adelante asumiendo más trabajo o reduciendo el tamaño de proyecto, aunque esta última opción de momento está descartada.
- **Riesgo de abandono (Proyecto):** la experiencia nos dice que esto es algo típico por desgracia en esta asignatura, lo único que se puede hacer es asumir entre todo el grupo la carga de trabajo del miembro que falta, ya sea reutilizando lo que lleve hecho o empezando desde el principio si hiciera falta.
- **Riesgo de comunicación (Proyecto):** ya ha ocurrido y supone retraso en los plazos, entregas inacabadas y más trabajo de corrección. Para evitarlo se han dispuesto varias vías de comunicación entre los miembros del grupo para que esto no vuelva a ocurrir.

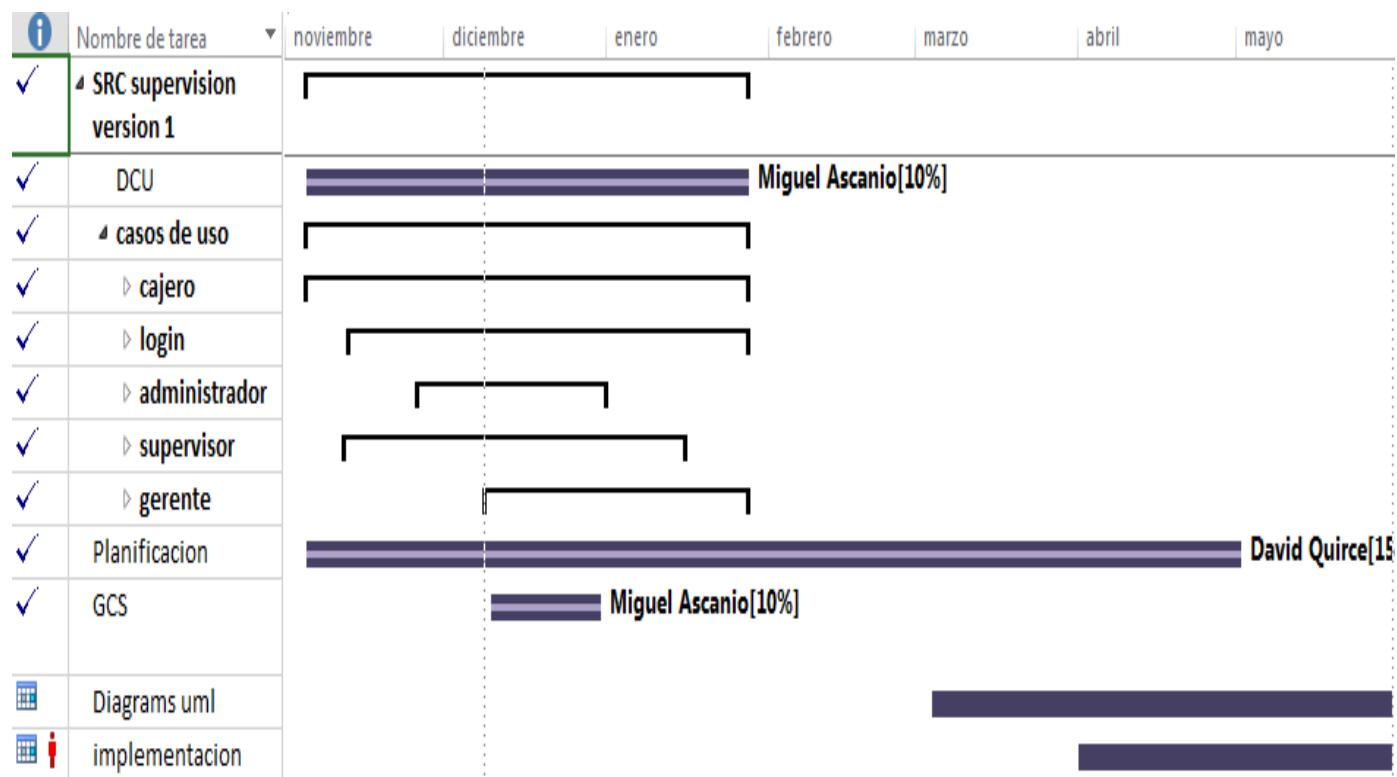
4 Planificación temporal

4.1 Estructura de descomposición del trabajo o Planificación temporal

La forma de organización del proyecto es:

- Estudio del ámbito de trabajo y funcionamiento de los TPV
- Análisis y planteamiento de los módulos a tratar
- Módulo del Login
- Módulo del Supervisor
- Módulo del Cajero
- Módulo del Gerente
 - contabilidad
 - estadísticas
- Módulo del administrador

4.2 Gráfico Gantt



4.3 Red de tareas

4.4 Tabla de uso de recursos

5 Recursos del proyecto

5.1 Personal

A continuación una breve explicación de que cualidades y atributos o defectos se han detectado en el personal de equipo.

- Álvaro Miguel Rodríguez: es una persona activa y dispuesta además tiene conocimientos de diseño gráfico, no cursa TP y esto supondrá un problema en la fase de implementación.
- David Quirce: es comunicativo, expresivo y serio lo cual ayuda en su labor de jefe del grupo, además ya ha aprobado TP con que jugará un papel importante en la implementación. Por otro lado trabaja y esto suele traducirse en menos horas de disponibilidad.
- Javier Toledano: conocedor de la asignatura lo cual supone una ventaja a la hora de trabajar. También tiene conocimientos de Java. Por otro lado también trabaja y esto supone un problema ya que dispone de menos horas libres.
- Daniel Tocino: sabe Java y conoce la asignatura, cumple con los plazos y es serio con el proyecto.
- Guillermo Romero: actualmente cursa TP por lo que se espera que programe más adelante.
- Miguel Ascanio: responsable e implicado con el proyecto, cursa TP por lo que se espera que implemente la aplicación más adelante.
- Esther Ávila: tiene una gran capacidad de trabajo, por desgracia no cursa TP y esto supondrá un problema en la fase de implementación.
- Laura Pérez: es una persona responsable, con capacidad de aprendizaje que cumple los plazos y siempre está dispuesta a trabajar, no cursa TP y esto supondrá un problema en la fase de implementación.

5.2 Hardware y software

Queda pendiente para más adelante la investigación sobre modelos de impresoras térmicas, cajones de dinero y pantallas (táctiles) TPV, dado que todavía se desconoce los recursos del cliente.

5.3 Lista de recursos

Además de los equipos hardware, en este caso ordenadores personales y de la universidad, cabe destacar el uso de las siguientes aplicaciones:

- Word: para la edición y creación de casi todos los documentos.
- Excel: para la creación de tablas y el boceto de la planificación.
- Proyect: con esta herramienta se ha creado y exportado en pdf el diagrama de Gantt adjunto con este proyecto.
- Adobe Photoshop: para la creación de la mayoría de las pantallas. Dada la gran cantidad de herramientas de las que dispone y que las imágenes venden más que el texto buscamos calidad en su edición.
- Eclipse: se usara en el desarrollo de la aplicación, como editor y compilador de Java.
- Google Drive: almacenamiento en la nube con múltiples opciones que permiten al grupo acceder al proyecto.

6 Organización del personal

6.1 Estructura de equipo (si procede)

Para la realización del Proyecto se ha dispuesto un jefe de grupo (David Quirce), que se encarga de repartir tareas, editar y revisar la planificación y el SRS. En esta labor se apoyará en el encargado de la gestión de cambios (Miguel Ascanio). Por otro lado más adelante se crearan dos grupos de trabajo, el equipo de documentación y el equipo de programación, dado la falta de personal con conocimientos para programar en Java. Con lo cual queda decir que hasta la fecha se ha realizado un reparto lo más equitativo posible de las tareas y que todos los miembros del grupo conocen las áreas de trabajo del mismo.

Se utiliza un sistema centralizado controlado, con un jefe de grupo que ayuda en muchos aspectos, el más importante la comunicación, todos informan al jefe de los avances, las dudas o sobre qué hacer. Esto supone que en las reuniones no se forma ninguna discusión y evita que al final no se llega a nada por ejemplo, además hay comunicación vertical entre los miembros del grupo a modo de consultas por ejemplo, cuando un caso de uso depende de otro.

Concretando un poco más, una pequeña lista de los trabajos realizados por cada miembro del grupo:

- Álvaro Miguel Rodríguez: además de la realización de los casos de uso, desde el primer momento revisa y apoya a todo el equipo a la hora de realizar los pantallazos.
- David Quirce: fue autoproclamado líder del grupo al ver que en la primera reunión nadie concretaba nada, al final el grupo en general está contento con su labor, se encarga de revisar las entregas, que se cumplan los plazos, de la planificación en general, además de realizar casos de uso.
- Javier Toledano: además de realizar casos de uso, se ha encargado de la matriz de traza por dos razones, la primera que no suele poder venir a la reuniones y así tiene que revisar el proyecto para hacerla, la segunda y casi la primera es que al ya tener conocimientos de IS le sería más fácil hacerlo.
- Daniel Tocino, Guillermo Romero, Esther Ávila y Laura Pérez: de momento han realizado la mayoría de casos de uso y esquemas incluidos en el SRS.
- Miguel Ascanio: responsable de la gestión y control de software, está asignado a esta tarea dado que es una persona meticulosa y detallista, además ha realizado y ampliará si es necesario el DCU y es el autor de algún caso de uso.

6.2 Informes de gestión

En cuanto a la gestión del grupo cabe destacar que es una tarea complicada, y debido a la inexperiencia ha habido veces que no se ha realizado de la manera correcta. Dicho esto y en líneas generales el encargado de la gestión del grupo es el jefe del equipo.

La realización de la gestión ha comenzado por la imposición necesaria de reuniones semanales desde noviembre todos los jueves. En las reuniones se ha seguido la siguiente dinámica. Primero se informa al grupo del día y hora de la reunión, siempre son en la biblioteca de la facultad de informática de la UCM, en las salas para trabajos en grupo, la hora siempre es la misma las 13:00, la duración una hora y el tema a tratar en líneas generales siempre ha sido el mismo, qué hemos entregado, cómo lo hemos hecho y si se puede mejorar, y por otro lado qué vamos a hacer, cómo y si las fechas previstas son factibles o no, y en caso de que no cómo hacer que lo sean sin retrasar el proyecto.

Una vez comenzada la reunión lo primero es escuchar al jefe de equipo, él marca los defectos y cosas a mejorar y pregunta al grupo su opinión y si hay algo más que comentar. Entonces comienza la segunda parte donde se exponen ideas y se analiza el proyecto, para acabar teniendo en cuenta lo dicho y la planificación ya marcada se reparten tareas y concluye la reunión.

Las reuniones han ido evolucionando conforme avanza el proyecto, hasta noviembre parte de la reunión se ha dedicado a la conceptualización del proyecto, qué va hacer nuestra aplicación y cómo, en diciembre y enero a esta fase se le ha dedicado menos tiempo, en su lugar se ha hecho hincapié en la gestión de los cambios, el control de las versiones, el mantenimiento de unos estándares y sobre todo en seguir intentando cumplir con la planificación.

7 Mecanismos de seguimiento y control

7.1 Garantía de calidad y control

Para garantizar la calidad del proyecto hemos de tener varias cosas en cuenta, la primera es una buena gestión y control de cambios, la segunda es la coherencia del proyecto y la tercera es tener contento al cliente.

Empecemos por el cliente, debemos ser capaces de captar lo que necesita y de hacer un software que cumpla con sus expectativas, pero también debemos ser capaces de hacerle ver el trabajo que se realiza y lo importante que es tener claro lo que quiere, porque una vez arranque el proyecto hacer grandes cambios sería lo mismo que parar de golpe una bola de nieve cayendo por una ladera. La bola se romperá y quedará algo parecido a una bola y más pequeña y será difícil que al llegar debajo de la ladera (nuestra fecha de entrega) sea igual de grande que si la hubiéramos dejado caer.

En cuanto a la coherencia con el proyecto parece sencillo pero no lo es, al trabajar en grupo hay que conseguir coordinarse lo suficientemente bien para que el trabajo sea uno sólo y todo cuadre a la perfección. Por ello es importante planificar bien el proyecto, tener claro lo que vamos a hacer y tener una buena comunicación en el grupo para poder llevar a cabo el proyecto, de forma que al finalizar parezca todo igual y sobre todo que cumpla con los requisitos prometidos, que mantenga una estética unificada y que el proyecto no se contradiga a sí mismo.

En conclusión, gestionar correctamente los cambios y mantener unos estándares de calidad en todo el proyecto son claves para poder finalizar el proyecto con éxito.

7.2 Gestión y control de cambios

7.3 Introducción

7.3.1 Propósito

El objetivo de este documento es definir una serie de estándares, normas y procesos para evitar confusiones y problemas durante todo el proceso de creación de Software, consiguiendo así un trabajo más barato y eficiente, aumentando los beneficios.

7.3.2 Alcance

Este documento pretende cubrir las necesidades anteriormente descritas sobre todos los tipos de artefactos de este proyecto.

7.3.3 Definiciones

- ECS: elementos de configuración de Software
- CU: caso de uso
- DCU: diagrama de casos de uso
- ECP: encargado de cambios del proyecto

7.3.4 Referencias

7.4 Tipos de artefactos a gestionar (los ECS)

7.4.1.1 CU y DCU

7.4.1.1.1 Esquemas asociados

7.4.1.1.2 Notas asociadas

7.4.1.2 Pantallazos

7.4.1.2.1 Fichas asociadas

7.4.1.3 Archivos de código (java)

7.4.1.3.1 Javadoc

7.4.1.4 Documentos de proyecto

7.4.1.4.1 Documento SRS

7.4.1.4.2 Plan de proyecto

7.4.1.4.2.1 Planificación

7.4.1.4.2.2 Plan GCS

7.4.1.4.2.2.1 Estándares, normas y procesos

7.4.1.4.3 Manuales de usuario

7.4.1.5 Otros esquemas / notas especiales

7.5 Criterios y protocolos para nombrar los ECS

7.5.1 ECS dependientes de CU

El nombre de los artefactos se basará en el CU al que haga referencia, y por norma general se usará el siguiente esquema:

`tipoArchivo_nombreCU_resp_ver _ESTADO.EXT`

- Tipo de archivo:
 - CU: caso de uso
 - DCU: diagrama/esquema de funcionamiento del CU asociado
 - Pant: pantallazo asociado
 - PantF: ficha del pantallazo
 - CRC: tarjeta CRC asociada
- Nombre CU
 - El nombre de CU asociado
- Estado
 - <<sin estado>> : se considera una versión que ha terminado de revisarse
 - PndR: el archivo está pendiente de revisión, que ha sido notificada, pero no ha comenzado
 - InR: el archivo está siendo revisado
- Resp
 - El responsable del archivo. Iniciales del nombre y apellidos.
- Ver
 - Versión del archivo (ver gestión de versiones)

7.5.2 Archivos de Código

Los archivos de código java se nombrarán con un nombre breve y descriptivo. La primera letra en mayúscula. La versión del código se indica como comentario dentro del mismo.

7.5.3 Documentos de Proyecto

Se nombrarán como en el apartado 2

7.6 Quién es responsable de los procedimientos de GCS y de la creación de Líneas Base.

En consenso de todos los miembros del proyecto, el encargado de cambios del proyecto, de los procedimientos, los estándares y las líneas base será Miguel Ascanio Gómez.

7.7 Políticas para el Control de Cambios y la Gestión de Versiones

A continuación se detallan los protocolos a seguir para cambios en los ECS del proyecto.

Existen básicamente dos tipos de cambio: por **mejora** del ECS o por **error** del mismo. A pesar de la diferencia entre ambos tipos, se tratarán de forma similar.

7.7.1 Nota importante sobre cambios leves

Algunos cambios leves, tales como correcciones ortográficas, pequeños fallos en lo gráfico de una pantalla, mínimos cambios en un código, etc.; se corregirán en cuanto sean detectados por quien los detecte o, si no fuera posible, por el autor del fallo. Estos cambios, por tanto, no seguirán el proceso descrito en adelante, aunque se tendrá cuidado de notificar el cambio si algún otro artefacto debiera ser modificado.

7.7.2 Clasificaciones del cambio

7.7.2.1 Prioridad

La urgencia del cambio será evaluada por el ECP, dando prioridad a los cambios más importantes que, por norma general, serán los errores.

7.7.2.2 Impacto

El impacto de un cambio tiene en cuenta cuantos artefactos pueden ser afectados por el propio cambio, ya que un mismo cambio puede generar más cambios. Dichos cambios generados por otro cambio contarán como parte del cambio principal, quedando reflejado como un único cambio en los documentos.

7.7.3 Dificultad

Mide la dificultad del cambio. Se tiene en cuenta su impacto.

7.7.4 Proceso de notificación

7.7.4.1 Notificación de un cambio

- Se informará al ECP del cambio a realizar.
- El ECP se pondrá en contacto con los encargados de los artefactos afectados por el cambio.
- El ECP calificará el cambio (como se describe anteriormente).

7.7.4.2 Análisis de viabilidad de un cambio

Una vez contactadas las partes implicadas en un cambio, el ECP se encargará de estudiar la viabilidad del mismo. Para ello se tendrá en cuenta el tiempo, la dificultad, y verificar que cumple los requisitos del proyecto. Si un cambio no es viable, se desecha.

Las conclusiones de este análisis se guardan en el archivo de cambio del propio ECS (ver ap. 6)

7.7.4.3 Archivo de prioridad

Cuando se decide que un cambio es viable, se incluirá en el archivo de cambios pendientes. Este archivo contempla:

- El nombre del ECS
- La calificación del cambio
- Fecha de notificación
- Estado, indica si el cambio ha comenzado o no
- Breve descripción del cambio

En el archivo se proporciona un ejemplo para seguir el formato.

Este archivo queda ordenado por la prioridad de los cambios pendientes, quedando los más prioritarios arriba.

7.7.5 Proceso de cambio

Los cambios a realizar se abordarán según el archivo de prioridad, y se procederá como sigue:

7.7.5.1 Realizar la modificación

Se procede a realizar el cambio según se analizó.

7.7.5.2 Verificación

Después se procede a validar el cambio: si es correcto, se fija el cambio y se refleja en los documentos del proyecto; si no es correcto, se revisa, se vuelve a estudiar la viabilidad y se realiza el cambio, repitiendo el proceso hasta ser correcto.

Todo esto se reflejará en los archivos de log de cambio general, y en los archivos de cambio de cada ECS, como se describe en el apartado 6.

7.7.6 Gestión de versiones

7.7.6.1 Formato de numeración

g.p.e.m

- G (general): versión general del ECS. Se aumenta con cambios muy importantes. Cada nueva versión paralela reinicia el resto de contadores. Será 0 hasta que no se considere una versión final, en ese caso será un 1.
- P (paralela): si se tratan varios archivos del mismo ECS, se diferenciarán aquí. Valor 0 si no hay versiones paralelas, si las hubiera se numeran en orden cronológico empezando por el 1 (no marcar la primera con un 0). Cada nueva versión paralela reinicia los contadores error y mejora.
- E (errores): cada corrección de un error incrementa este número. Comienza en 0.
- M (mejoras): cada vez que se instaure una mejora se incrementa este número. Comienza en 1. Reinicia en 0.

7.8 Registros para mantener el rastro de los cambios

Registro general

En este registro se almacenará la información básica de todos los cambios que han sido realizados en los ECS del proyecto. Se almacenará la siguiente información:

- Nombre del ECS
- Autor del cambio
- Versión que se ha procedido a cambiar
- Versión después de completar el cambio
- Fecha de inicio del cambio
- Fecha del final del cambio

El archivo se mantendrá con los cambios ordenados cronológicamente. Para información más detallada se accederá a los archivos de cada ECS.

7.8.1 Registro propio de cada ECS

Para disponer de toda la información necesaria, no se guardará toda la información en un mismo archivo, sino que cada ECS tendrá su propio archivo de cambio, cuando ocurra un cambio en dicho ECS se añadirá un apartado dedicado a dicho cambio, donde se reflejará lo siguiente:

- El motivo del cambio
- Las conclusiones del proceso de análisis de viabilidad
- Reiterativamente:
 - Los cambios realizados y quien los realiza
 - La validación y prueba de los mismos

Además, al comienzo del archivo se mantendrá una lista con todas las versiones y una breve descripción del cambio.

Fuente: Ingeniería del software Escrito por Ian Sommerville.