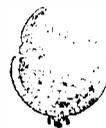


UNIVERSIDAD NACIONAL DE LA PLATA



FAC. DE INFORMATICA
U.N.L.P.

CERES

Administración y gerenciamiento de la etapa de mantenimiento del software

Trabajo de Grado

Director:

Lic. Rodolfo Bertone

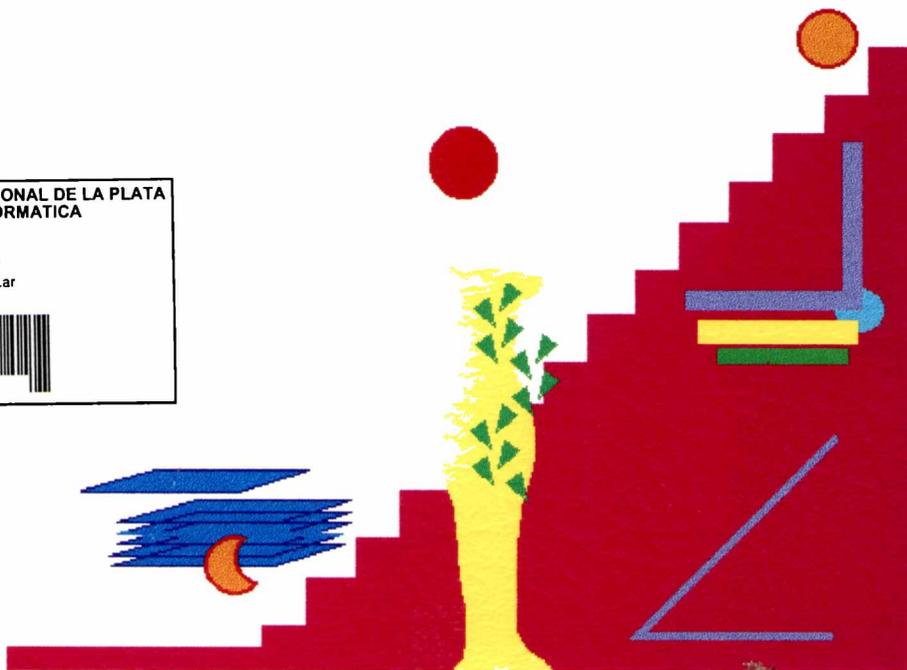
Alumnas:

María Belén Alonso

María Vanesa Amadeo

Octubre 1997

TES 97/2 DIF-01962 SALA	 <p>UNIVERSIDAD NACIONAL DE LA PLATA FACULTAD DE INFORMATICA Biblioteca 50 y 120 La Plata catalogo.info.unlp.edu.ar biblioteca@info.unlp.edu.ar</p>  <p>DIF-01962</p>
----------------------------------	---



DONACION.....
\$.....
Fecha..... 29-8-05
Inv. E..... Inv. 1962

TES
97/2

INDICE

INTRODUCCIÓN	1
WORKFLOW	3
UNA CLASIFICACIÓN PARA WORKFLOW.....	4
PASOS PARA LA CONSTRUCCIÓN DE UN PROCESO WORKFLOW.....	7
MITOS DE LA AUTOMATIZACIÓN WORKFLOW.....	9
CONCLUSIONES.....	10
MANTENIMIENTO	12
TIPOS DE MANTENIMIENTO.....	13
MANTENIMIENTO DEL SOFTWARE	14
<i>Características del mantenimiento</i>	15
Mantenimiento estructurado frente al no estructurado.....	15
Costos del mantenimiento.....	16
Problemas del mantenimiento.....	16
<i>Efectos laterales del mantenimiento</i>	17
<i>Cómo desarrollar el mantenimiento</i>	18
<i>Mantenimiento y Calidad del Software</i>	19
CONCLUSIONES.....	21
INTEGRACIÓN DE HERRAMIENTAS	23
OBJECT LINKING AND EMBEDDING (OLE) Y OLE AUTOMATION.....	24
CONTROLES MAPI.....	25
CONTROLES CRYSTAL REPORTS.....	25
ABSTRACCIÓN DE LA SOLUCIÓN	27
EL MÓDULO DE ADMINISTRACIÓN.....	27
EL MÓDULO DE GESTIÓN DEL CAMBIO.....	28
EL MÓDULO DE DESARROLLO.....	30
<i>Metodología de Ingeniería de Software adoptada</i>	30
<i>Utilización de la metodología en el módulo de desarrollo</i>	31
IMPLEMENTACIÓN DE LA HERRAMIENTA	34
ADMINISTRACIÓN DE RECURSOS.....	34
<i>Proyectos protagonistas del cambio</i>	34
<i>Clientes que usan los proyectos</i>	37
<i>Staff encargado del mantenimiento</i>	38
SEGUIMIENTO DE LOS CAMBIOS A LOS PROYECTOS.....	39
<i>Gestión de cambios</i>	40
<i>Desarrollo de cambios</i>	45
<i>Control de Versiones</i>	51
ADMINISTRADOR DE RECURSOS PARA EL DESARROLLO DEL CAMBIO.....	52
INTERACCIÓN CON OTRAS HERRAMIENTAS.....	54
<i>Manipulación de documentos</i>	54
<i>Correo electrónico</i>	55
INGRESO AL SISTEMA.....	57
CONCLUSIONES.....	58
APÉNDICE A: DOCUMENTOS DE LA GESTIÓN DEL CAMBIO	61
<i>I. Documento del Pedido de Cambios</i>	63
<i>II. Documento de Autorización del Pedido de Cambio</i>	64
<i>III. Documento del Informe Técnico por Pedido de Cambio</i>	65
<i>IV. Documento del Informe Técnico por Pedido de Cambio para el Area Costos</i>	66
<i>V. Informe del Administrador de Base de Datos por Pedido de Cambio</i>	67
<i>VI. Informe del Administrador de Base de Datos para el Area Costos</i>	68
<i>VII. Informe del Area Costos por Pedido de Cambio</i>	69
<i>VIII. Informe de Pedido de Cambio Rechazado</i>	70

<i>IX. Plan de Trabajo para la Ejecución del Cambio</i>	71
1. Documento Preliminar	73
2. Plan de Ejecución de Tareas	74
3. Equipo de Trabajo	75
APÉNDICE B: DOCUMENTOS DEL DESARROLLO DEL CAMBIO	77
<i>I. Documento de Especificación del Sistema</i>	79
<i>II. Documento de Especificación de Requerimientos del Sistema</i>	81
<i>III. Documento del Diseño Preliminar de Datos</i>	83
<i>IV. Documento del Diseño Preliminar de Estructura</i>	84
<i>V. Documento del Diseño Detallado de Datos</i>	85
<i>VI. Documento del Diseño Detallado de Estructura</i>	86
<i>VII. Documento asociado a la Codificación</i>	87
<i>VIII. Documento de Prueba del Soft</i>	88
<i>IX. Manual del Usuario</i>	90
<i>X. Manuales de operación e Instalación</i>	92
<i>XI. Documentos de la Instalación del Sistema</i>	93
Descripción de la Instalación	95
Cronograma de Instalación	96
BIBLIOGRAFÍA	97



“...En sus viajes por el mundo buscando a su hija, robada por el rey de los infiernos, Plutón, la diosa Ceres llegó al Ática. Allí fue muy bien recibida, con mucho afecto y reverencia....Queriendo premiar a sus amables amigos, Ceres les enseñó a sembrar y a cosechar el trigo, y a hacer el pan, dando a Triptolemo, hijo del rey, un carro mágico, para que recorriera el mundo enseñando tan útil menester....”

Elegimos el nombre de la diosa griega de la agricultura, Ceres, para este trabajo que constituye la cosecha de nuestros años de estudio y esfuerzo compartidos.

Agradecimientos

- Al Licenciado Rodolfo Bertone, que nos acompañó y guió a lo largo de todo el proyecto.
- Al LIDI y a todos sus integrantes.
- A nuestros familiares y a Dani, que estuvieron siempre que los necesitamos.

Belén y Vanesa

Octubre, 1997

Introducción

El soft es un producto de naturaleza dinámica. Si bien no se puede pensar en un sistema que permanezca eternamente brindando servicios, se puede aspirar a que sobreviva y se adapte a una buena parte de las modificaciones que el medio ambiente y sus usuarios le exijan.

La posibilidad de modificación, está condicionada en primer lugar por la calidad del desarrollo del soft: para que un sistema pueda ser mantenido eficazmente, es necesario que haya sido desarrollado siguiendo una metodología dictada por la ingeniería de soft, que garantice la documentación del sistema.

Por otra parte, la empresa que desarrolla soft debe entender que el mantenimiento del soft producido es una de las funciones que tiene que cumplir organizadamente, de la misma manera que cumple con el desarrollo de nuevos sistemas.

No obstante, la gestión de mantenimiento sufre una administración deficitaria en la mayoría de los casos. Entre los motivos más relevantes se cuentan:

- la mayor parte de las veces, el mantenimiento del soft es una actividad intermitente, azarosa, que está determinada por la aparición inesperada de errores, o por los cambios en el entorno (otras veces, es iniciativa de los desarrolladores). Es decir, en general no se puede planificar con anticipación sobre qué cambios se va a trabajar.
- el crecimiento de la demanda de soft hace que se concentren los recursos de la empresa en satisfacer los nuevos requerimientos. Esta tarea sí puede planificarse.
- además, los desarrolladores de soft suelen motivarse mucho más ante el trabajo que propone un nuevo desarrollo que ante el trabajo que implica mantener un sistema.

Las principales consecuencias de una gestión de mantenimiento pobre o nula son:

- los cambios se insertan sin un método de gestión, lo que genera un problema de descontrol sobre el soft con el que se cuenta. Para la empresa que desarrolla soft, esta forma de proceder genera problemas aunque el resultado del cambio sea exitoso.
- seguir circuitos informales para el mantenimiento de sistemas suele redundar en una gran cantidad de especialistas dedicados a soluciones rutinarias.
- no se realizan análisis de costos, por lo que no se cuenta con un registro de la frecuencia de los distintos tipos de cambios que son necesarios, ni de cuanto tiempo y recursos se utilizan en la tarea de mantenimiento.
- no se realizan análisis de impacto, por lo que la verdadera dimensión del cambio se va descubriendo a medida que se trabaja en él, de modo que no se puede prever la duración del trabajo.

De lo anteriormente expuesto resulta que, dado que los cambios en el soft son inevitables, y tarde o temprano alguien deberá desarrollarlos, es preferible enfrentar a la etapa de mantenimiento de forma organizada.

La solución que se presenta pretende que la ejecución del cambio se someta a dos etapas: una de gestión propiamente dicha, en la cual se evalúa la necesidad del cambio, el costo económico que implica, el impacto sobre otros elementos del sistema, la cantidad y calidad de recursos humanos necesarios, para finalmente decidir si se da curso o no a la segunda etapa: el desarrollo del cambio. Esta etapa de desarrollo implica una serie de tareas similares a las cumplidas durante el desarrollo del soft (especificación, diseño, codificación, prueba, instalación).

Para las dos etapas mencionadas es de fundamental importancia la documentación generada. La automatización de la solución administra la documentación, teniendo siempre disponible el documento necesario de la versión necesaria.

La documentación de la primera etapa asegura que se conoce de antemano la magnitud del cambio y por lo tanto se puede organizar su desarrollo, definiendo tanto el equipo como los tiempos de desarrollo.

Para la segunda etapa en principio se requiere la documentación original del sistema. Cada cambio que genere una nueva versión del soft, genera también la documentación asociada a la versión, por lo que nunca el soft queda modificado e indocumentado.

Workflow

Actualmente, las empresas deben enfrentarse con la competencia global, necesitan reducir el costo de los negocios y desarrollar rápidamente nuevos servicios y productos para mayor satisfacción del cliente.

Además, el avance de la tecnología, la capacidad y facilidad de manejo de las PC actuales, lleva a que las organizaciones cuenten con una gran cantidad de información de todo tipo: documento, hojas de cálculo, análisis, estadísticas, gráficos y diagramas. La utilización del correo electrónico, las redes de área local y global permiten compartir toda esta información, facilitando el trabajo en grupo y cooperativo. Por lo tanto, el volumen de actividades que se resuelven por medio de una computadora en una empresa aumenta considerablemente. Así, surge la necesidad para los responsables de la misma, de monitorear cómo y cuándo se realiza cada trabajo.

Para cumplir con los requerimientos anteriormente expuestos, las empresas deben constantemente reconsiderar y optimizar el modo en que hacen los negocios y cambiar sus sistemas de información y aplicaciones para soportar los procesos de la evolución de los negocios.

Workflow es una de las tecnologías capaz de facilitar estos requerimientos, proveyendo metodologías y software para soportar:

- I. Modelado de los procesos de negocios para capturarlos como especificaciones del workflow.
- II. Re ingeniería de los procesos de negocios para optimizar los procesos especificados.
- III. automatización del workflow de modo tal que se puedan generar implementaciones del workflow a partir de las especificaciones.

No hay un criterio común establecido que diga qué es un workflow y qué características debe proveer un sistema manejador de workflow. Bajo el término workflow, que se usa muy a menudo, las personas pueden estar refiriéndose a un proceso de negocios, especificaciones de un proceso, software que implementa y automatiza un proceso, o software que simplemente soporta la coordinación y colaboración de personas que implementan un proceso.

Existen diversas definiciones que se pueden adecuar:

Workflow es cualquier tarea realizada en serie o en paralelo por dos o más miembros de un grupo de trabajo para alcanzar un objetivo común.

Cuando se dice *cualquier tarea* se refiere a que workflow implica un amplio rango de actividades de negocios relacionadas. Estas actividades se pueden desarrollar en etapas, una después de la otra, o simultáneamente por diferentes individuos o una combinación de ambas, por ello se utilizan las palabras en *serie o en paralelo*. Se pide también que *dos o más personas* realicen la tarea porque sino no existiría un workflow. Como el nombre lo sugiere, una tarea se considera workflow si "fluye" (flow) de un individuo a otro.

Los individuos que participan en un workflow deben trabajar buscando un *objetivo común*. Si trabajan en proyectos distintos, no conforman un workflow.

Se define workflow como una colección de tareas organizadas para realizar algún proceso de negocios. Una tarea puede ser desarrollada por uno o más sistemas de software o uno solo o un equipo de humanos o una combinación de ambos. Las tareas humanas incluyen desde un alto nivel de interacción (por medio de comandos) con la computadora hasta uno muy bajo (por ejemplo usarla sólo para indicar el proceso de una tarea).

Ejemplos de tareas incluyen modificación de un archivo o base de datos y generación de una cadena de mensajes. Además de una colección de tareas, un workflow define el orden de invocación de las mismas o condiciones bajo las cuáles éstas deben ser invocadas, sincronizadas y el flujo de la información entre ellas (data flow).

De las definiciones anteriores se observa que las aplicaciones que pueden ser manejadas por paquetes workflow van desde la automatización de oficinas que operan con datos y aplicaciones tradicionales, hasta el manejo de misiones críticas con datos y aplicaciones a medida. Por lo cual hay varias actividades de negocios en una organización que caen en la categoría de workflow.

Una clasificación para Workflow.

No hay un acuerdo general para caracterizar o categorizar los workflow y los sistemas que los manejan. Además la mayoría de las clasificaciones se basan en los workflows altamente automatizados que acceden a información compartida.

Se puede hablar de dos clasificaciones interesantes. La primera, basada en los tipos de aplicaciones, (ideada por Mc Cready), distingue entre tres clases de workflow: *ad Hoc*, *administrativos* y *de producción*.

Los puntos que se tienen en cuenta en esta clasificación incluyen:

repetitividad y predictibilidad del workflow y las tareas.

cómo se inicia y cómo se maneja el control en un workflow (por ejemplo, va desde control por humanos hasta el automatizado)

requerimientos para la funcionalidad de los sistemas manejadores de workflow.

Los *workflows ad Hoc* realizan procesos de oficina, como por ejemplo, generación de documentos o administración de actividades, donde no hay un patrón establecido para el movimiento de la información entre las personas. Este tipo de workflow, generalmente involucra la coordinación de los humanos, colaboración, o co-decisión.

Así, el orden y la coordinación de las tareas en un *workflow ad Hoc* no están automatizadas pero en cambio, están controladas por humanos. Además, el orden de las tareas y las decisiones de coordinación se desarrollan mientras se construye el workflow. Estos workflow, típicamente involucran pequeños grupos de profesionales y se utilizan para soportar actividades a corto plazo que requieren una solución workflow rápida, por ejemplo, el soporte para armar el programa de un congreso.

Los sistemas que soportan *workflow ad Hoc* deben proveer facilidades para la coordinación de los humanos, colaboración y co-decisión. No es necesaria una función

que controle el orden de las tareas. Así, los usuarios de un *workflow ad Hoc* necesitan acceder al sistema para determinar si un trabajo se completó.

Además, los *workflows ad Hoc* no son de misión crítica, es decir, que fallas periódicas de tales workflows no tienen importancia y no interfieren con el proceso total. Estos workflow generalmente usan una base de datos para almacenar la información compartida (por ejemplo, documentos). Los sistemas que soportan *workflows ad Hoc* muchas veces son denominados groupware.

La siguiente figura muestra un workflow ad Hoc que involucra el proceso de revisión de los papers de una conferencia.

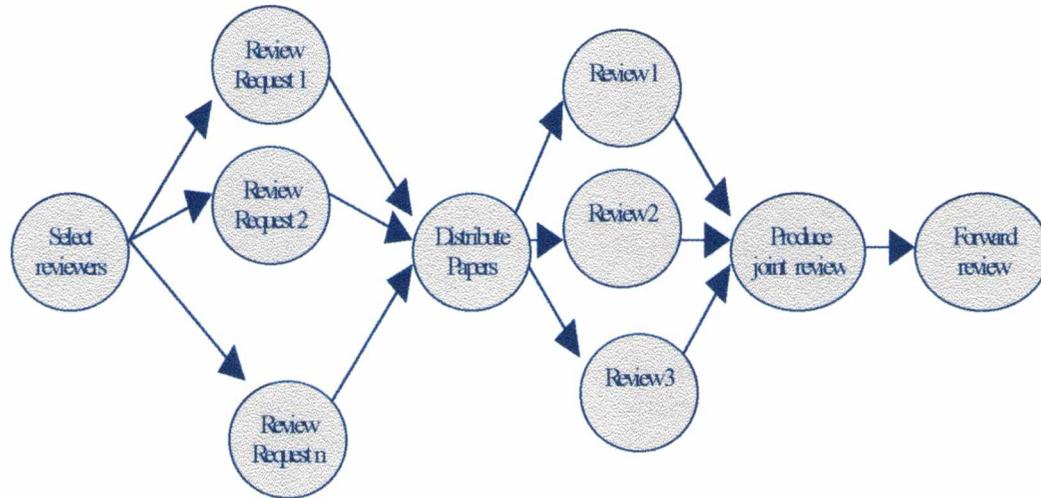


figura 1

El proceso de revisión consiste en seleccionar los revisores, distribuir los papers entre ellos, para su revisión, juntar los resultados obtenidos, y finalmente enviar la respuesta a los autores. Es un *workflow ad Hoc* porque involucra:

- negociación para seleccionar los revisores
- colaboración entre los revisores para producir la revisión final.

Los *workflows administrativos* involucran procesos repetitivos y predecibles con reglas simples de coordinación de tareas, tal como seguir un reporte de expensas o de viáticos a través de un proceso de autorización. El orden y la coordinación de tareas en un *workflow administrativo* se pueden automatizar.

Los sistemas que soportan *workflows administrativos*, manejan funciones para distribuir información y documentos. Este tipo de workflow no abarca un procesamiento complejo de la información ni tampoco requiere acceso a sistemas de información múltiples usados para soporte de producción y/o atención al cliente. Estos sistemas no son, por lo general, de misión crítica. La tecnología de infraestructura que usan está típicamente basada en e-mail.

Como ejemplo, consideremos otra vez el proceso de revisión de papers. Sin embargo, esta vez supongamos que los revisores se conocen de antemano (por ejemplo, los mismos revisores para todos los papers). Además supongamos que los revisores no colaboran entre ellos para producir una revisión en conjunto. En vez de esto, producen revisiones individuales que son consideradas por el editor quien toma la decisión final. Bajo estas consideraciones, el ejemplo de la revisión de papers se transforma en un workflow administrativo como se muestra en la siguiente figura.

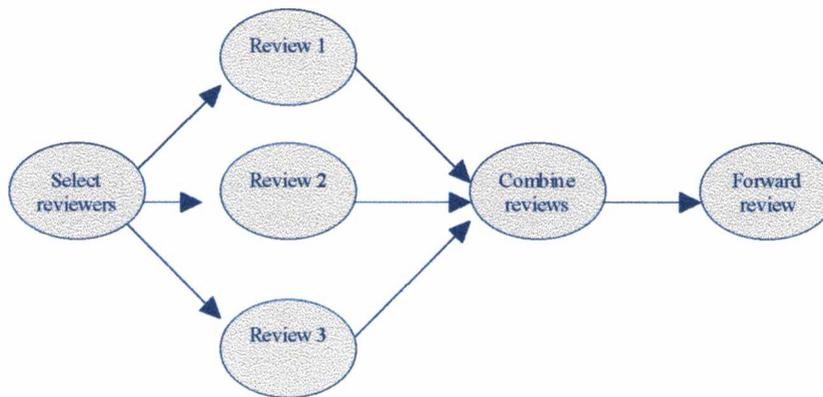


figura 2

A los usuarios de un *workflow administrativo* se les “avisa” cuando tienen que realizar su tarea. Mientras que los revisores que usan un *workflow ad Hoc* necesitan acceder al sistema para determinar si el trabajo fue completado, los revisores que usan un *workflow administrativo* pueden recibir un e-mail con las instrucciones de revisión junto con el paper que tiene que revisar y un formulario para los comentarios del revisor. Cuando el formulario está completo, automáticamente es enviado al Comité Principal, a quien se le avisa cuando todos los revisores terminaron.

Los *workflow de producción* involucran procesos de negocios repetitivos y predecibles, tal como aplicaciones de préstamos o reclamos de seguros. A diferencia de un *workflow administrativo*, este tipo de workflow implica típicamente un proceso complejo de la información, involucrando acceso a múltiples sistemas de información. El ordenamiento y la coordinación de las tareas en dichos workflows se pueden automatizar. Sin embargo, la automatización de los workflows de producción es complicada debido a:

- la complejidad del proceso de la información
- el acceso a múltiples sistemas de información para realizar el trabajo y recuperar datos para la toma de decisiones (los workflows administrativos dependen de los humanos para la mayoría de las decisiones y el trabajo realizado)

Los sistemas que soportan los *workflows de producción* deben proveer facilidades para definir las dependencias de las tareas, y controlar la ejecución de éstas sin o casi sin la intervención del humano.

Estos workflow, por lo general son de misión crítica y deben enfrentarse con la integración e interoperabilidad de los sistemas de información heterogéneos, autónomos y/o distribuidos (HAD).

Consideremos un ejemplo simplificado de los reclamos de salud de la figura 3. En este workflow se escanea un formulario de reclamo y se almacena en un objeto de la base de datos. Luego se indexa manualmente en una base de datos relacional. Esta información es analizada por una tarea “Assess Claim” automatizada. Si el reclamo es rechazado, un representante discute con el cliente y pueden ponerse de acuerdo y arreglar algún pago o rechazar el reclamo. Si el pago se lleva a cabo, otra tarea accede a las bases de datos de Finanzas y registra el pago.

Las diferencias más importantes entre estos workflow y los ad Hoc y administrativos son:

- la interacción de sistemas de información con procesos de negocios.
- el uso de realizadores de tareas automatizadas (no humanos)

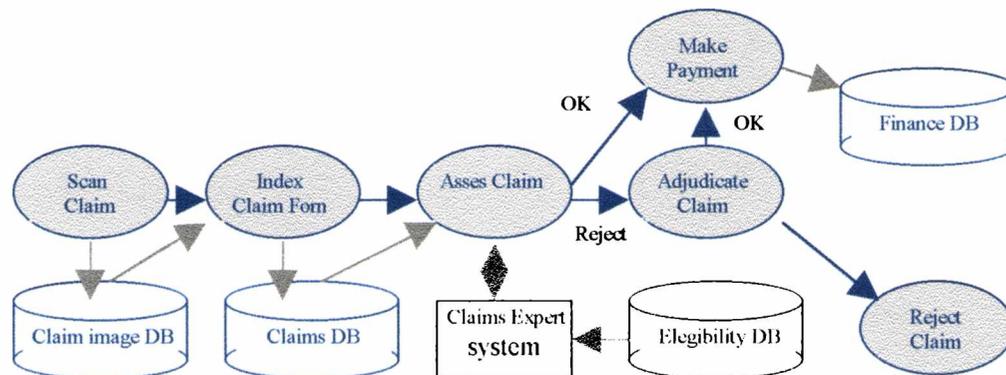


figura 3

Otra caracterización interesante es la que resulta de considerar si se involucra humanos o sistemas de software para realizar y coordinar las tareas. Los requerimientos de sistemas que manejan workflows en este ambiente son soportar la coordinación y colaboración de humanos y mejorar la comunicación entre ellos. Los humanos, sin embargo, deben asegurar la consistencia de los documentos y los resultados del workflow.

En el otro extremo, los workflows orientados a los sistemas, involucran sistemas de computación que realizan un gran número de operaciones de computación intensiva y tareas de software especializados. Además de estar altamente automatizados, estos sistemas acceden a sistemas de información HAD. Mientras que las implementaciones workflow orientadas a los humanos generalmente controlan y coordinan tareas humanas, las implementaciones de workflow orientadas a los sistemas controlan y coordinan tareas de software. En consecuencia, las implementaciones orientadas a los sistemas deben incluir software para control de concurrencia y técnicas de recuperación para asegurar consistencia y confiabilidad. Esto no es necesario ni tampoco se puede proveer en los workflows orientados a los humanos. Los workflow orientados a los humanos tienen procesos semánticos (tal como investigar donde mandar un documento) pero no tienen un conocimiento de la semántica de la información que está siendo procesada, por lo tanto los sistemas que soportan estos workflows están para asistir a las personas y no puede hacerse responsable de mantener la consistencia de los datos, desde que no tiene información semántica.

Por otro lado, los workflows orientados a sistemas tienen más conocimiento de la semántica de los datos involucrados. Entonces, a estos se les da una mayor responsabilidad para mantener la consistencia de la información.

Pasos para la construcción de un proceso workflow

Un proceso workflow se crea así:

1. Definir una actividad o tarea que el grupo de trabajo necesita realizar y las reglas de negocios que rigen dicha actividad.

2. Dividir la tarea en subtareas, también llamadas "etapas". Cada etapa representa una lista bien definida de cosas que las hará un individuo y las cuales están unidas lógicamente. Una tarea se puede dividir en subtareas de muchos modos distintos. Aquí es donde se requiere un criterio de negocios para decidir cómo dividir una tarea en sus etapas componentes.
3. Decidir el conjunto de especialistas requerido para realizar cada etapa: esto especificará las funciones del trabajo o los individuos que pueden ser llamados para realizar cada etapa.
4. Decidir la secuencia en la cual la etapa se realizará.
5. Si algunas etapas se realizan sólo si se cumple una condición, identificar estas etapas y definir las condiciones.
6. Definir un "mapa" del workflow, el cual identifica las etapas y la secuencia o "flow" en la cual las mismas se realizarán, luego asociar funciones o individuos con cada etapa.
7. Crear los formularios, documentos e instrucciones que usarán los individuos en cada etapa para realizar la subtarea.

De lo anterior, se puede decir que un sistema que se encargue del manejo de workflow debe proveer herramientas para:

- modelar procesos y especificar un workflow: requiere modelos de workflow y metodologías para capturar un proceso como una especificación workflow.
- Re-ingeniería de procesos: requiere metodologías para hacer re-ingeniería de los procesos optimizando su especificación workflow e
- implementación workflow: requiere metodología / tecnología para usar sistemas de información y personas para implementar una especificación workflow (para scheduling, ejecución y control de las tareas del workflow)

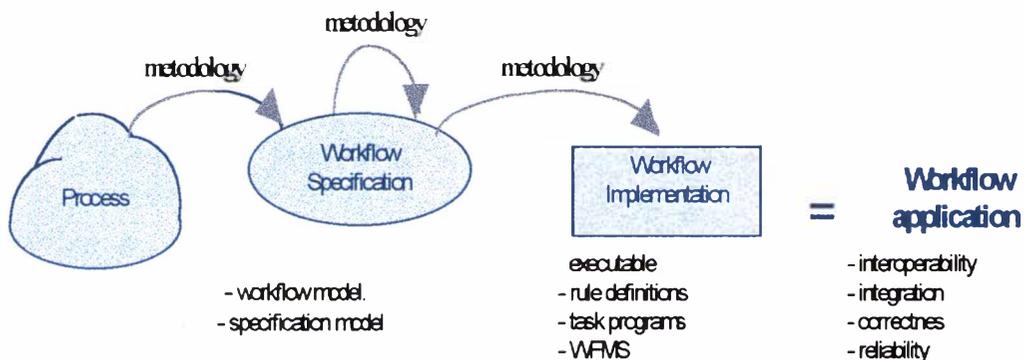


figura 4

Explicaremos cada una de estas cuestiones.

Modelar un proceso: antes de capturar un proceso primero necesitamos entenderlo. Esto usualmente involucra entrevistar personas con conocimiento sobre el proceso. Cuando hay suficiente conocimiento sobre el proceso, se realiza la especificación workflow para capturar el proceso.

Para realizar una especificación Workflow necesitamos un *modelo*. El modelo de Workflow generalmente incluye un conjunto de conceptos que se utilizan en la descripción de *procesos*, sus *tareas*, las *dependencias* entre las tareas y los *roles* requeridos (es decir, características de los individuos o sistemas de información) que pueden realizar las tareas especificadas. Para especificar el orden y sincronización de las tareas en él y los atributos para describir cada una y los roles para realizarlas, se pueden utilizar reglas, condiciones y/o constructores gráficos.

Re-ingeniería de un proceso: el objetivo de la re-ingeniería es optimizar el proceso de negocios. Las estrategias de optimización de un proceso dependen de los objetivos de la re-ingeniería (por ejemplo, incrementar la satisfacción del cliente, reducir el costo de hacer los negocios, introducir nuevos productos o servicios). A partir de la especificación de Workflow se obtiene una descripción de alto nivel la cual facilita el razonamiento sobre la eficiencia de los procesos de negocios.

Implementando y automatizando un Workflow: en la implementación hay que enfrentarse con las cuestiones asociadas con realizar un Workflow usando una computadora, software y/o sistemas de información. La automatización debe resolver el scheduling y control de ejecución de las tareas del Workflow.

No se requiere implementación o automatización de un Workflow cuando la única razón es capturar el proceso de negocios y razonar sobre su eficiencia. En otro caso, las especificaciones Workflow se usan para implementar y automatizar Workflows.

Mitos de la automatización Workflow

Workflow es una tecnología de software nueva, que puede mejorar la productividad de las organizaciones. Naturalmente, hay confusión y conceptos incorrectos entre los usuarios. Estos problemas se presentan por la gran cantidad de cambios en la tecnología, la ausencia de una terminología general, la gran cantidad de aplicaciones que se pueden automatizar con Workflow y la variedad de soluciones propuestas.

Algunos de estos conceptos equivocados son:

Workflow = Re-ingeniería

Workflow = Groupware

El primer mito es que la automatización Workflow y el proceso de re-ingeniería son uno sólo. Esto se debe a que casi la mayoría de los papers, conferencias o seminarios sobre Workflow, incluyen una discusión sobre re-ingeniería de los procesos de negocios. Las personas hablan de ellos como si fueran uno sólo. Los analistas incluyen la re-ingeniería en sus discusiones sobre automatización Workflow porque quieren mostrar que sus productos mejoran el modo en que sus clientes hacen los negocios, y hay muchos más libros sobre re-ingeniería que sobre automatización Workflow.

Re-ingeniería y automatización Workflow no significan lo mismo. Es muy importante entender esta distinción. Automatización Workflow es una tecnología de software que provee un modo de automatizar un proceso de negocios.

La re-ingeniería es el hecho de analizar los procesos de negocios de una compañía y cambiarlos con el objetivo de mejorarlos en algún modo.

Las organizaciones pueden automatizar el proceso de negocios usando automatización Workflow sin re-ingeniería. Del mismo modo es posible hacer re-ingeniería sin automatización Workflow alguna.

Se puede también realizar un proceso de re-ingeniería y como una parte del mismo, instalar una automatización Workflow. La automatización Workflow puede ayudar en un proceso de re-ingeniería y viceversa.

Este mito es uno de los factores que influye en la baja aceptación de la automatización Workflow. Automatización Workflow es una tecnología que mejora la productividad, como un procesador de texto y las planillas de cálculo. Sin embargo, el proceso de re-ingeniería involucra un cambio en el modo en que las organizaciones hacen los negocios, esto produce miedo, incertidumbre y resistencia al cambio. Por ello su aceptación es baja.

Cuando, explícita o implícitamente se iguala automatización Workflow con re-ingeniería, las reacciones en contra de esta última se vuelcan sobre la automatización Workflow.

Otro mito habla de Groupware y Workflow como tecnologías idénticas. Si bien, tanto el software de Workflow como el de Groupware u operaciones en grupo permiten a grupos de usuarios compartir información, en realidad no significan lo mismo.

El software básico de Workflow gracias a herramientas gráficas (como iconos y enlaces) y a un lenguaje de script, permite armar la estructura del sistema de enrutado a partir de un diseño de diagrama de flujo. Ya en funcionamiento, se ofrece la posibilidad del monitoreo del estado de las tareas en tren de realización, así como de las pendientes y las terminadas.

Por otro lado, Groupware es un concepto que no puede decirse que responda realmente a una tecnología o especificación técnica, aunque hay quienes piensan que cualquier aplicación que utiliza correo electrónico entra en la categoría de Groupware.

De cualquier modo, lo importante es que Groupware se ve hoy en día como una forma de facilitar el trabajo cooperativo y la interacción entre componentes de un grupo que trabajan con la misma información al mismo tiempo. Bajo este esquema, la administración del Workflow es una extensión lógica de esta filosofía, y de hecho, muchos productos de Groupware proveen una interfaz visual que permite la creación de ciertos tipos de Workflow.

Conclusiones

Los avances en la tecnología actual facilitan la comunicación dentro de las empresas, por medio de la utilización de bases de datos, correo electrónico y redes de área local.

En toda compañía existe, automatizado o no, un conjunto de tareas para cumplir con los objetivos de la misma. Por medio de analizar la situación se pueden identificar estas tareas y la información que fluye entre ellas.

Para aprovechar las ventajas de la tecnología se necesita algo más que la simple transferencia de papeles y el workflow manual como soporte de la estructura de tareas antes mencionada.

Se puede pensar que la tecnología de workflow sería útil para automatizar todo el proceso en el cual existe un conjunto de tareas a realizar, ya sean secuenciales o paralelas, las cuáles comparten información y existe una relación de consecutividad entre ellas. Además de la posibilidad de automatización y mejorar el control global de todas las tareas en el proceso, la tecnología workflow puede medir el proceso para obtener estadísticas que serían de mucha utilidad para optimizaciones futuras.

El relevamiento necesario para automatizar los procesos de la compañía utilizando la tecnología workflow, podría ser aprovechado para identificar problemas y optimizar procedimientos.

Mantenimiento

Desde la concepción de una idea para un sistema de software, hasta su implementación y entrega al cliente y aún después de esto, el sistema protagoniza un proceso gradual de desarrollo y evolución. Por ello, se dice que el software tiene un **ciclo de vida** compuesto de varias fases. Cada una de estas fases, resulta en el desarrollo de una parte del sistema o algo asociado con él, tal como un plan de pruebas o un manual del usuario.

El ciclo de vida es un tema muy estudiado en Ingeniería de Software, y existen diferentes versiones. Entre ellas podemos mencionar el modelo en cascada, la construcción de prototipos y el modelo en espiral.

En el desarrollo de sistemas reales es muy difícil encontrar que se cumpla algún modelo de ciclo de vida puro. En muchos casos se suele utilizar los beneficios de cada paradigma para obtener mejores resultados.

Independientemente de las diferentes filosofías de cada uno de los modelos, del proyecto o la organización de la empresa de desarrollo, se puede establecer en general, que el ciclo de vida de cualquier software presenta tres fases: (figura 1)

- definición,
- desarrollo,
- mantenimiento

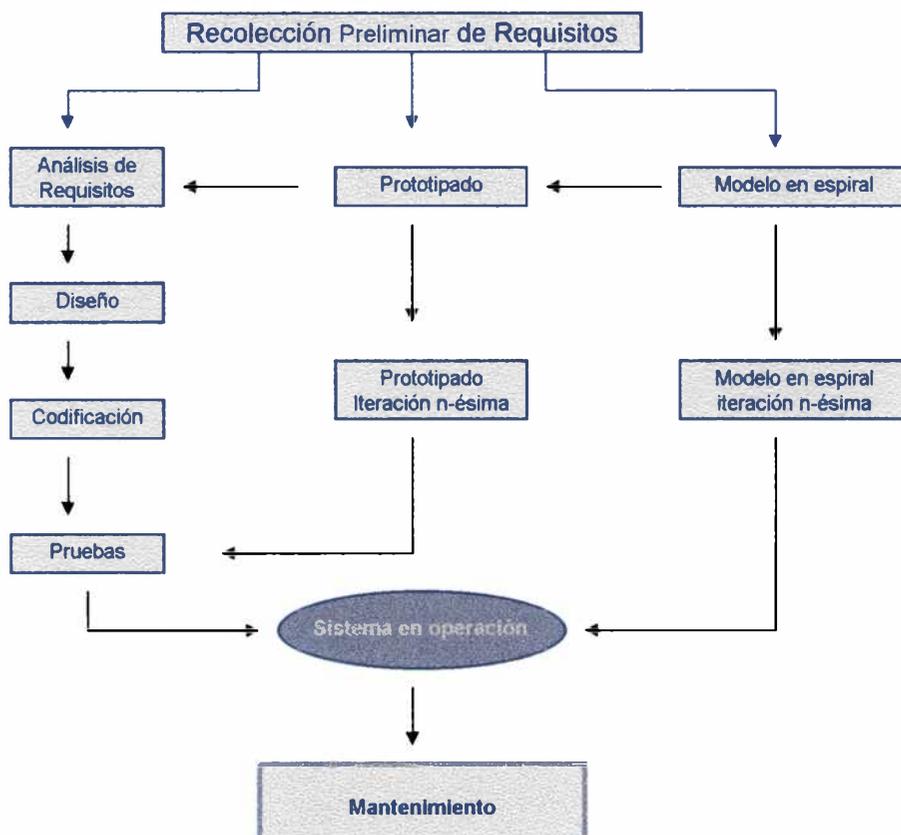


figura 1

En la fase de definición se identifican las principales funciones del software, cuáles son los datos de entrada al sistema y cuáles las salidas esperadas. Independientemente del paradigma de ingeniería de software aplicado, se necesitan tres pasos específicos:

- análisis del sistema,
- planificación del proyecto de software
- análisis de requisitos.

En la fase de desarrollo se realizan las tareas que construyen el software propiamente dicho. El enfoque se pone en diseñar los datos, descubrir los detalles procedimentales, realizar el diseño y traducir este diseño para obtener el código. Además se define la metodología para probar el software. En definitiva, independientemente de los métodos aplicados durante esta fase, se producirán en cierto modo, tres pasos concretos:

- diseño
- codificación
- prueba del software

El mantenimiento, la última etapa del ciclo de vida, es necesario por la naturaleza del software, que no es estático y se encuentra en constante evolución. Al mantenimiento lo podemos comparar con un salto de un río de alta montaña con el que se encuentra una persona que hace "rafting": el deportista sólo ve la espuma o el vapor pero no conoce la magnitud real del mismo, hasta que se enfrenta con él.

Es un hecho que, en esta fase, se invierte la mayor parte del presupuesto del proyecto. A medida que se desarrollan más sistemas, surge una tendencia molesta, la cantidad de esfuerzo y recursos dedicados al mantenimiento crece. Ocurre cada vez más a menudo que las organizaciones de software lleguen a ver el mantenimiento como una barrera, la cual impide a la empresa embarcarse en nuevos proyectos.

Esta última etapa se centra en el cambio del software, que va asociado a la corrección de errores, a las modificaciones requeridas por la evolución del entorno del software y a las modificaciones debidas a los cambios en los requisitos del cliente dirigidos a ampliar o reforzar el sistema. En la fase de mantenimiento se vuelven a aplicar los pasos de las fases de la definición y del desarrollo, pero sobre software ya existente. Desarrollándose así, una especie de sub-ciclos de vida, dentro del ciclo principal, uno por cada cambio solicitado para el proyecto.

Tipos de mantenimiento

Basándose en los tipos de cambio que puede sufrir el software se pueden decir que existen cuatro tipos de mantenimiento:

- *correctivo*: actúa para corregir los errores que no se descubrieron en la fase de desarrollo, antes de poner en uso el software,
- *adaptativo*: debido a los avances de la tecnología, el entorno externo del software cambia, y éste necesita modificarse para adaptarse a esos cambios.

- *perfectivo*: incorpora mejoras solicitadas por el usuario.
- *preventivo*: mejora la futura facilidad de mantenimiento y la fiabilidad como base para futuras mejoras.

El mantenimiento correctivo está relacionado con las deficiencias en el desarrollo del soft, primero con la construcción del software y segundo con el relevamiento inicial. El primer caso tiene que ver con errores de diseño o implementación que no fueron detectados en la etapa de prueba del sistema, y deben ser solucionados rápidamente. En el segundo, mucho más complicado, el producto de software no cumple las necesidades del usuario, los cambios a efectuar son mucho mayores.

El mantenimiento adaptativo o perfectivo está relacionado con cambios en las necesidades del usuario. Las modificaciones a realizar no son tan traumáticas como las del caso anterior, ya que no reportan errores no encontrados y que deban tener una rápida solución. Los cambios son realizados en común acuerdo con el usuario y pueden llevarse a cabo en forma metodológica, lo cual tiende a asegurar un cambio gradual que no influya sobre las operaciones existentes del software.

Existe una diferencia muy marcada entre el mantenimiento correctivo y el adaptativo o perfectivo. El mantenimiento correctivo debe resolverse rápidamente, puesto que, cuando el mismo se produce, el sistema no puede continuar operando. Los restantes, básicamente, deben comenzar por un estudio exhaustivo de su impacto sobre las especificaciones o requerimientos originales del sistema. Este estudio permite evaluar la viabilidad del cambio a producir dándole elementos al administrador del proyecto o al usuario final para decidir si las modificaciones solicitadas serán llevadas a la práctica.

Mantenimiento del software

Una de las principales características que debe cumplir un sistema es la facilidad de mantenimiento. El término mantenimiento de software comúnmente se utiliza para referirse a modificaciones que se realizan a los sistemas de software después de su versión inicial. Pero, la mayoría de las investigaciones han demostrado que, la mayor parte del tiempo gastado en el mantenimiento se utiliza para agregarle al producto nuevas características que no estaban presentes o estaban mal definidas en la especificación original.

Se puede decir que para que el software sea mantenible debe cumplir con dos requerimientos:

- facilidad de reparación
- estar preparado para evolucionar.

La facilidad de reparación se refiere a que permita identificar la causa de los errores y corregir sus defectos sin un gran esfuerzo. Por otro lado, el software está preparado para evolucionar si es capaz de aceptar las modificaciones necesarias para satisfacer los requerimientos actuales del usuario, ya sea cambiando funciones existentes o agregando nuevas.

La diferencia entre los dos conceptos anteriores no siempre es clara. Por ejemplo, si la especificación de requerimientos es vaga, puede que no sea posible distinguir entre corregir un error o satisfacer requerimientos nuevos.

Características del mantenimiento

El mantenimiento del software es uno de los puntos menos estudiados de la Ingeniería de Software. Hay muy pocas investigaciones sobre el tema y se han propuesto muy pocas metodologías para encararlo. Además, es un tema crítico en las organizaciones actuales, las cuáles se encuentran muy preocupadas por el alto costo del mantenimiento de sus sistemas.

Se analizarán las características del mantenimiento desde tres puntos de vista diferentes:

1. las ventajas y desventajas de estructurar las tareas necesarias para llevar adelante el mantenimiento.
2. los costos asociados al mantenimiento.
3. los problemas con los que se enfrenta el equipo encargado del mantenimiento.

Mantenimiento estructurado frente al no estructurado

Como cualquier otro producto de ingeniería, los productos del software se modifican con el tiempo para proveer nuevas funciones o modificar las existentes. Sin embargo hay una diferencia importante entre las modificaciones del software y la de otros productos de ingeniería. En el caso de las modificaciones en las otras ramas de la ingeniería, comienzan en el nivel de diseño y luego se realizan sobre la implementación del producto. Por ejemplo si se planea la ampliación de una casa, primero se estudia su viabilidad, luego se realizan las modificaciones en el plano y por último se comienza la construcción de la nueva parte.

En el caso del software, lamentablemente, casi nunca se realiza el mantenimiento por un camino organizado. En primer lugar se debe considerar la gran cantidad de software existente que no cuenta con documentación, sólo se dispone de los programas fuente, por lo tanto llevar a cabo el mantenimiento es una tarea muy ardua y complicada. Por otro lado, aún contando con la documentación y sin importar si el cambio pueda ser radical en la aplicación, generalmente se comienza con la implementación del cambio sin ningún estudio previo, ni siquiera un intento de modificación del diseño original. Es peor aún el caso si no se arma, una vez finalizado el trabajo, una documentación a posteriori, modificando la especificación del sistema para registrar el cambio. Esta forma de proceder conduce a un *mantenimiento no estructurado* y el equipo de desarrollo se enfrenta a los problemas asociados con todo software que no se desarrolló con una metodología bien definida.

Si no existe documentación alguna, las herramientas de re-ingeniería pueden ayudar en la tarea de evaluación del código. Pero no pueden suplantar la documentación de diseño, tan necesaria para la supervivencia del software.

En caso de existir una documentación completa del software, la tarea de mantenimiento debería comenzar con una evaluación del diseño. Luego se determinan cuáles son los requerimientos del cambio solicitado, se analiza el impacto del mismo y se traza un plan de tareas para llevarlo a cabo. Luego, se modifica el diseño y se desarrolla el nuevo código, cada uno de los pasos anteriores con sus respectivas revisiones. Por último se realizan todas las pruebas y se vuelve a lanzar el software.

A esta cadena de pasos se la puede llamar *mantenimiento estructurado* y sólo se puede conseguir si se cuenta con la documentación adecuada y el software se desarrolló con una metodología de Ingeniería de Software. Si bien una documentación

organizada no garantiza un mantenimiento libre de problemas, se reduce la cantidad de esfuerzo requerido y mejora la calidad del producto final.

Costos del mantenimiento

Consideremos ahora los costos del mantenimiento, que crecen a medida que se desarrollan nuevos sistemas y no se establece alguna metodología para organizarlo. No se habla solamente de los costos de dinero, existen otros costos intangibles que están relacionados por ejemplo, con el desarrollo de nuevos sistemas que se pospone o se pierde debido a que los recursos disponibles están dedicados al mantenimiento de sistemas viejos. También se produce insatisfacción del cliente cuando un pedido de cambio no se puede atender en un tiempo razonable. Otro costo intangible está constituido por los errores que introducen los cambios en el software mantenido, lo cual disminuye la calidad global del mismo.

Problemas del mantenimiento

Por último, se puede afirmar que la mayoría de los problemas con los que se encuentra el equipo de desarrollo encargado del mantenimiento se deben a las deficiencias al desarrollar y definir el software original. Es decir que, la falta de una metodología y de los controles necesarios en las tareas de desarrollo del software, la mayoría de las veces, se pagan en el mantenimiento.

Algunos de los problemas clásicos con el mantenimiento son los siguientes:

- es muy difícil el control de las diferentes versiones del software. Los cambios que las generaron no están adecuadamente documentados.
- generalmente es muy complicado, si no imposible seguir el proceso de construcción del software.
- cuando la documentación es pobre o nula, es muy difícil seguir un programa que fue desarrollado por otra persona que, a menudo, no está presente para prestar alguna ayuda o explicación.
- en la mayoría de los casos no existe una previsión del cambio en el momento de desarrollo del sistema. Así, las modificaciones serán difíciles y propensas a errores.
- el mantenimiento no se ve como un trabajo atractivo, debido a las frustraciones que se presentan cuando se trabaja en mantenimiento.

La responsabilidad de los problemas anteriormente expuestos se puede delegar en la gran cantidad de sistemas de software que se desarrollaron sin seguir patrón alguno. Sin tener en cuenta ninguna metodología de Ingeniería de Software. Si bien, no se solucionarán todos los problemas por utilizar un desarrollo estructurado, se pueden encontrar soluciones parciales a los problemas asociados con el mantenimiento.

Efectos laterales del mantenimiento

La modificación del software es muy riesgosa. Poniendo sumo cuidado en la documentación del diseño y en la prueba, puede que se disminuya el riesgo de errores, pero no se eliminarán los efectos secundarios del mantenimiento.

En el contexto de mantenimiento de software, el término efectos secundarios, implica un error o un comportamiento inesperado del software que surge de una modificación. Se definen tres grandes categorías de efectos secundarios:

- sobre el código
- sobre los datos
- sobre la documentación

Cualquier cambio que se aplique al *código*, por más pequeño que sea puede tener consecuencias impredecibles. Si lo que falla es el software de control de monitoreo de un paciente, el error puede ser fatal. Si bien, no todos los efectos secundarios serán tan dramáticos, siempre que hay un cambio se aumenta la posibilidad de error y el error trae aparejado problemas.

Debido a que la comunicación con la máquina se realiza por medio del lenguaje de programación, la probabilidad de que aparezcan efectos secundarios es muy alta. Algunos de los cambios que tienen mayor posibilidad de inducir a un error son:

1. .eliminación, agregación o modificación de un subprograma o identificador
2. . cambios para mejorar la eficiencia en ejecución
3. . modificación de apertura o cierre de archivos
4. . cambios del diseño que provoquen importantes cambios en el código

Los errores que se pueden incorporar van desde los más sencillos que se detectan y corrigen fácilmente, hasta problemas que pueden provocar la caída del sistema.

Durante la etapa de mantenimiento se puede modificar la *estructura de los datos* que maneja el software, lo cual puede provocar que el diseño del mismo no se complemente con la nueva estructura y de esta forma aparecerán los errores.

Los siguientes cambios tienen mayor posibilidad de inducir a errores:

1. redefinición de constantes locales o globales
2. redefinición de formatos de registros o de archivos
3. modificación del tamaño de los arreglos u otras estructuras de datos.
4. reorganización de los argumentos de los subprogramas.

Los efectos secundarios sobre los datos se pueden evitar con una documentación exhaustiva que describa cada una de las estructuras de datos y su relación con los módulos del software.

Por último, los efectos secundarios sobre la *documentación* se producen cuando no se reflejan los cambios del código fuente en la documentación del diseño y en los manuales del usuario.

Siempre que se haga un cambio sobre el flujo de datos, sobre la arquitectura del diseño, sobre los procedimientos o cualquier otra característica asociada al software, se debe actualizar la documentación técnica de soporte. Si la documentación no refleja fielmente el estado actual del software puede ser peor que si no existiese, porque lleva a una incorrecta interpretación de las características del software provocando así los efectos secundarios.

Para el usuario, la calidad del software se basa en qué tan buena es la documentación que describe su uso. Por lo tanto, si no se reflejan los cambios en la documentación del usuario, éste no estará conforme con el desempeño del sistema, ya que no sabrá como utilizarlo, provocándose así efectos secundarios que afectan la calidad del software producido.

Este último tipo de efectos secundarios se puede reducir substancialmente revisando toda la configuración antes de lanzar la nueva versión.

Cómo desarrollar el mantenimiento

Todo cambio a un sistema debe comenzar por un análisis pormenorizado del impacto que tendrá sobre las especificaciones o requerimientos del sistema. Este análisis es necesario siempre que exista una propuesta de modificación para el sistema, ya sea porque:

- el usuario decide que quiere añadir una nueva función al sistema actual, o no está conforme con alguna función actual y quiere cambiarla
- el personal que programa el mantenimiento decide recodificar un módulo para hacerlo más eficiente o para facilitar las futuras modificaciones
- se producen cambios en la organización de la empresa que desarrolló el software, como por ejemplo cambiar el lenguaje de programación
- cambia la organización interna de la empresa para la cual se desarrolló el software, por ejemplo, se agrega una nueva dependencia a la cual el sistema debe enviar alguna salida
- se descubren nuevos errores para ciertas combinaciones de datos de entrada que no fueron testeadas en la etapa de prueba

Cualquier cambio como éstos debe ilustrarse, documentarse y ser verificado con el usuario, haciendo al modelo del sistema los cambios pertinentes. Para realizar esta tarea se utiliza un formulario de solicitud de cambio del sistema. El cambio puede involucrar alguno o todos de los siguientes detalles:

- añadir terminadores nuevos al diagrama de contexto o eliminar anteriores
- añadir nuevos elementos a la lista de acontecimientos o eliminar otros
- si el cambio es substancial, puede modificarse la declaración de propósitos en el modelo ambiental

- cambios en:
 - ⇒ los modelos de flujos de datos,
 - ⇒ modelos de entidad - relación
 - ⇒ modelos de transición de estados
- modificación y/o refinamiento de las especificaciones de procesos y el diccionario de datos,
- cambios en la interface hombre - máquina

Ninguno de estos cambios se desarrollará por sí solo. Puede que algunos sean muy simples y las modificaciones necesarias en la especificación y los programas se realicen en unos pocos minutos. Sin embargo la persona o grupo que realiza los cambios debe escribir una declaración precisa y detallada del impacto que causarán los cambios en la especificación del sistema, llamada *declaración de impacto*. También es necesaria una *declaración de impacto económico*, es decir una comparación entre el costo del cambio y el beneficio que se obtendrá.

Como es de esperarse, hay cambios que no modificarán la especificación del sistema: una corrección de un error, algún cambio para mejorar la legibilidad o la eficiencia del código o un cambio de hardware o software existentes (por ejemplo, un nuevo compilador o sistema operativo). Aún en estos casos es necesaria la declaración de impacto económico para que queden claros los costos y beneficios asociados con dicho cambio.

En todos los casos la declaración de requerimientos es sin duda el documento más importante a actualizar, ya que será la base para los futuros cambios o modificaciones y sin este documento en perfectas condiciones se puede perder el control de todo el sistema.

Para que funcione la metodología propuesta anteriormente se necesita mucha disciplina dentro de la organización.

Mantenimiento y Calidad del Software

Toda metodología formal para el desarrollo de software persigue como fin último lograr software de calidad.

La calidad del soft se "construye" (o se "destruye") con la actividad de cada etapa del ciclo de vida del soft. Es un objetivo a cumplir en cada paso, desde la especificación del sistema, hasta la etapa de mantenimiento.

Intuitivamente sabemos qué es la calidad del soft, no obstante una definición formal puede revelar aspectos ignorados que resulten de utilidad a la hora del desarrollo. Una de las definiciones presentadas en la bibliografía consultada establece que la calidad del soft es:

La concondancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo soft desarrollado profesionalmente.

La definición anterior plantea tres aspectos sobre los cuales es necesario hacer hincapié:

- concordancia con los requisitos formales: si el soft no da los resultados que de él se esperan, la calidad falla. Esto es cuando el soft no cumple con la funcionalidad para la que fue desarrollado o no cumple con las restricciones establecidas para su funcionamiento.
- concordancia con los estándares de desarrollo explícitamente documentados: es el resultado de seguir una metodología formal de desarrollo de soft, que como tal, intenta no dejar nada librado al azar, cada paso dado es parte de un todo (la metodología) que ha sido concebido para dar como resultado soft de calidad. Si bien el hecho de seguir una metodología de desarrollo no garantiza de por sí la calidad del soft, la carencia de una metodología generalmente asegura la falta de calidad del soft resultante.
- concordancia con los requisitos implícitos que se esperan del soft desarrollado profesionalmente: entre estos requisitos implícitos se encuentra la posibilidad de un buen mantenimiento del soft obtenido, que permita mantener en el tiempo la calidad lograda con el desarrollo. Otros requisitos en esta categoría son la fiabilidad, la corrección, la eficiencia, etc.

Mc Call definió una serie de factores que hacen a la calidad del soft, y los agrupó en las tres categorías que muestra la siguiente figura:

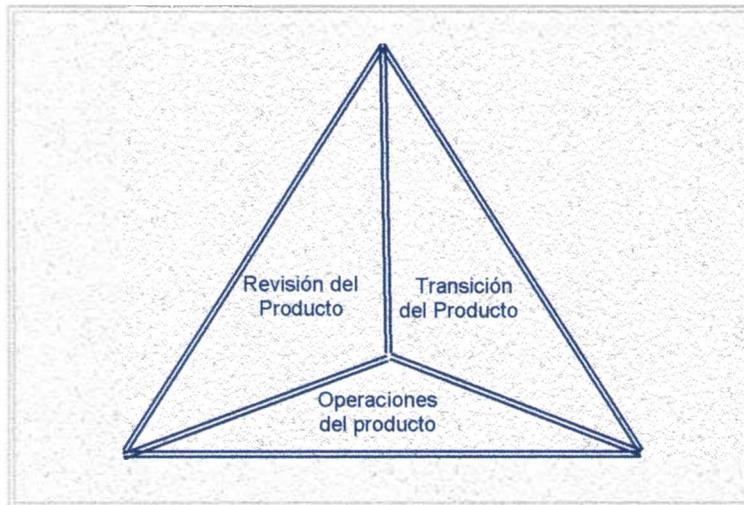


Figura 2

Cada uno de los conceptos incluidos en la figura puede asociarse con una pregunta, cuya respuesta da una pauta sobre la calidad del soft:

Características operativas

Corrección	¿Hace lo que se pidió?
Fiabilidad	¿Lo hace de forma fiable todo el tiempo?
Eficiencia	¿Cumplirá con las restricciones pedidas?
Integridad	¿Es seguro?
Facilidad de uso	¿Es amigable?

Capacidad de soportar los cambios:

Facilidad de mantenimiento	¿Puede ser corregido?
Flexibilidad	¿Puede ser cambiado?
Facilidad de prueba	¿Puede ser probado fácilmente?

Adaptabilidad a nuevos entornos:

Portabilidad	¿Puede ser usado en otra máquina?
Reusabilidad	¿Puede reusarse alguna de sus módulos?
Interoperabilidad	¿Puede interactuar con otros sistemas?

El segundo grupo de factores hace referencia a la etapa de mantenimiento del soft, a través de tres factores:

- facilidad de mantenimiento se refiere al esfuerzo que requiere llevar a cabo el mantenimiento correctivo, es decir ubicar un error en el sistema y corregirlo;
- flexibilidad hace referencia a la posibilidad de introducir modificaciones al soft (cambios adaptativos, perfectivos o preventivos) y
- facilidad de prueba hace referencia a la posibilidad de probar las partes del sistema y poder garantizar que realizan su función correctamente.

El hecho de que uno de los tres grupos apunte a la etapa de mantenimiento específicamente pone de manifiesto la relevancia de esta etapa en la calidad del soft.

El motivo es que una vez que se ha alcanzado cierto grado de calidad en el soft desarrollado, los cambios realizados en la etapa de mantenimiento constituyen una amenaza. Todo cambio tiene un fin altruista, el de mejorar el sistema, y por ende su calidad, pero en la práctica, una modificación en el soft puede generar efectos laterales que impacten como errores en otras partes del sistema.

Cuando un especialista realiza un cambio con mecanismos informales de gestión, centrando su atención en el cambio en sí mismo, y despreciando el contexto, es altamente probable que esté introduciendo, sin saberlo, más errores. El resultado: baja la calidad del software. La gravedad de esta situación se advierte si pensamos en el efecto acumulativo producto de sucesivas modificaciones al software

Conclusiones

Una de las etapas críticas en el ciclo de vida del software es el mantenimiento, debido a su alto costo, y a que sin un control adecuado puede llegar a provocar la decadencia de un sistema exitoso, hasta hacerlo totalmente inútil.

Controlar las modificaciones que se realizan sobre el software, involucra una serie de pasos que van desde manipular documentación realizada a partir del pedido

de cambio, hasta la instalación de la nueva versión de software al cliente. Para ello se necesita un conjunto de tareas que involucran los recursos humanos, de hardware y software existentes y además, actividades propias de la re-ingeniería.

El control de calidad no es ajeno a esta etapa. La forma en que se efectúa el gerenciamiento de los cambios es tomada en cuenta para determinar la calidad final de un producto.

De lo anterior, surge la necesidad de herramientas que automaticen la etapa de mantenimiento. Ellas deberían gestionar todo el proceso, desde el ingreso del pedido de cambio, el control de la modificación de todo lo necesario en la configuración del sistema, incluyendo documentación y software, y el manejo de las nuevas versiones del sistema.

Integración de herramientas.

La herramienta fue desarrollada teniendo en cuenta las pautas de construcción de un workflow automático. De este tipo de herramientas se espera que administren la interacción entre las aplicaciones que sean necesarias para facilitar la tarea del usuario. La aplicación principal administra el acceso a otras aplicaciones.

Con esta política se logra que sea invisible al usuario el intercambio de documentos y la comunicación requerida por el workflow para mantenimiento de soft, de modo que sus obligaciones se vean reducidas a completar su tarea específica. La herramienta provee al usuario del documento que necesita en el momento que lo necesita, se encarga de avisarle cuando tiene trabajos pendientes y de avisar a los encargados de los próximos pasos cuando el usuario termina su labor, la productividad de los actores aumenta.

Una de las características clave de Microsoft Windows, es su habilidad para compartir datos, en múltiples aplicaciones que se ejecutan simultáneamente. Los productos que corren sobre esta plataforma, en general, cuentan con los elementos para aprovechar esta facilidad, y por lo tanto pueden intercambiar datos entre ellos, de diferentes formas. Visual Basic es uno de estos productos.

Este lenguaje de programación permite diseñar aplicaciones orientadas a eventos, con la misma filosofía de interacción con el usuario que Microsoft Windows, y tiene la posibilidad de comunicarse con otros productos a través de controles especiales.

Fue elegido para implementar la solución de la herramienta de mantenimiento de soft, para la que fue necesario integrar un editor de textos potente, un sistema de correo electrónico y una herramienta de generación de reportes rápidos.

Como editor de textos se utilizó Microsoft Word, que provee una serie de características adecuadas a la solución.

El resultado de la labor en cada estadio de los workflows (de gestión y de desarrollo) se debe registrar en un documento. Estos documentos son diferentes para cada estadio. Word permite crear plantillas con el diseño de los documentos, de modo que el usuario tenga que completar campos con los datos necesarios, tal como si se tratara de un formulario en papel, y no generar un documento a partir de una "hoja" en blanco. Además, permite personalizar el ambiente con el que se encuentra el usuario cuando se abre un documento basado en esas plantillas.

Así la herramienta de mantenimiento de soft puede ofrecer el documento adecuado en el momento adecuado, se abre una sesión de Microsoft Word con un documento con el diseño esperado y con las posibilidades de trabajo en el editor, limitadas a las necesarias. Cuando el usuario deja el documento, en realidad cierra la sesión con el editor, y vuelve a la aplicación principal.

Como administrador de correo electrónico se eligió MS Mail. La herramienta para gestionar y desarrollar cambio en el soft, requiere de los servicios de correo electrónico para generar mensajes en forma automática y transparente para los usuarios, avisarles cuando tienen nuevos mensajes, permitirles enviar y contestar correos recibidos.

La mensajería automática es necesaria como parte de la automatización del workflow que sustenta a la solución. Cada vez que la acción de un usuario implique trabajo para alguno de los estadios que siguen en la secuencia, un mensaje automático debe acompañar al ruteo del flujo de información. Los encargados de los

estadios siguientes, deben ser informados del nuevo trabajo que tienen que atender, a la vez que este nuevo trabajo aparece en las listas de "trabajos para hacer".

Los actores de un workflow realizan diferentes pasos de una tarea común. Por lo tanto necesitan comunicarse. El correo electrónico provee una forma eficiente de administrar la mensajería entre los usuarios. Permite enviar, recibir y contestar mensajes sin sufrir demoras significativas, además de acumularlos o borrarlos según sea necesario.

Como generador de reportes rápidos se eligió Crystal Reports, que es una aplicación que provee Visual Basic desde su ambiente, por lo que resulta de fácil acceso durante el desarrollo del código, y sus resultados son buenos a la hora de generar reportes cuyo contenido se extrae de la base de datos del sistema.

Para integrar las herramientas mencionadas al sistema Visual Basic utiliza el control personalizado OLE y OLE Automation, controles personalizados MAPI y el control personalizado Crystal Reports.

Object Linking and Embedding (OLE) y OLE Automation.

OLE (vinculación e inserción de objetos) es un método que permite la interoperabilidad entre aplicaciones Windows.

Se considera objetos a las entidades de información (texto, gráficos, sonido, etc). El objeto encapsula los datos y la referencia a la aplicación que la generó. La aplicación receptora, no conoce ni necesita conocer, la información contenida en el objeto, ya que cualquier modificación o actualización será realizada por la aplicación fuente. Esto permite que una aplicación pueda recibir información de cualquier fuente y formato.

La aplicación principal (la herramienta para mantenimiento de soft en este caso) se denomina cliente (client) y la que provee, y eventualmente edita o modifica el objeto, se denomina servidor (server).

Ole Automation es una técnica que permite manipular objetos en forma remota, desde adentro de la aplicación cliente, invocando métodos que actúen sobre los objetos u obteniendo o estableciendo sus propiedades.

Visual Basic soporta OLE Automation y cuenta con los controles personalizados OLE para orquestar desde la aplicación cliente a un conjunto de otras aplicaciones servidoras.

En la implementación de la solución se utilizó un objeto OLE para interactuar con Microsoft Word y proveer acceso a los documentos de cada estadio del workflow. A través de la estructura de directorios que administra la herramienta, es posible establecer cuál es el camino del documento que se debe editar ante una solicitud de un usuario que cumple un determinado rol. Con este dato se establece la propiedad SourceDoc del objeto OLE, y luego se establece el valor de la propiedad Action para que Microsoft Word presente una ventana con el documento solicitado. Desde ese momento, el control es del procesador. Cuando el usuario cierra el procesador el control vuelve a la aplicación Visual Basic.

Controles MAPI

Los controles de la interfaz de programación de aplicaciones de mensajería (MAPI) permiten crear una aplicación MAPI en Visual Basic con el correo habilitado. Hay dos controles personalizados MAPI: MAPI session y MAPI messages.

Los controles personalizados MAPI son invisibles para el usuario. Para utilizarlos deben especificarse los métodos apropiados desde la aplicación cliente.

Para que estos controles funcionen tienen que estar presentes los servicios MAPI. Los servicios MAPI son suministrados de acuerdo con los sistemas de correo electrónico utilizados por Windows 3.0 o posteriores.

El control MAPI session establece una sesión MAPI. El control MAPI messages efectúa una variedad de funciones de sistema de mensajería después que se establece una sesión de mensajería con el control MAPI session.

Con el control MAPI messages se puede:

- acceder a los mensajes actuales en la bandeja de entrada
- crear un nuevo mensaje
- agregar y eliminar destinatarios de mensajes y datos adjuntos
- enviar mensajes (con o sin interfaz de usuario compatible)
- mostrar el cuadro de diálogo libreta de direcciones
- mostrar el cuadro de diálogo detalles
- efectuar acciones de responder, de responder a todos y de reenviar mensajes

Controles Crystal Reports

El paquete Crystal Reports contiene tres elementos diferenciados: Crystal Reports, el motor de impresión de Crystal Reports y el control personalizado de Crystal. Al utilizarse juntos, estos tres elementos proporcionan la capacidad para diseñar informes que se pueden incluir en forma transparente en la aplicación. Usando estas herramientas la aplicación genera informes que el usuario final podrá imprimir pero no podrá editarlos de ninguna forma.

Crystal Reports es un poderoso redactor de informes de Windows que puede usarse para diseñar una variedad casi infinita de informes personalizados. El diseño puede incluir campos constantes y campos relacionados a campos de la base de datos. Además permite realizar selección de los registros que son relevantes para el informe.

El motor de impresión de Crystal Reports es una biblioteca de enlace dinámico (DLL) que la aplicación llamará cuando un usuario desee imprimir uno de los informes diseñados.

El control personalizado de Crystal está compuesto por un conjunto de herramientas que le facilita la conexión entre su aplicación y el motor de impresión. Seteando adecuadamente las propiedades de este control e invocando sus métodos,

la aplicación inicia la impresión de los reportes previamente diseñados a partir de los datos actuales de la base de datos.

Abstracción de la solución

La solución fue diseñada a partir de tres módulos relacionados por los datos que usan, pero funcionalmente independientes. Una primera abstracción del sistema revela un workflow con dos estadios, correspondientes a dos de los módulos antes mencionados, sustentados por los datos generados por el tercer módulo, el de administración. (figura 1)

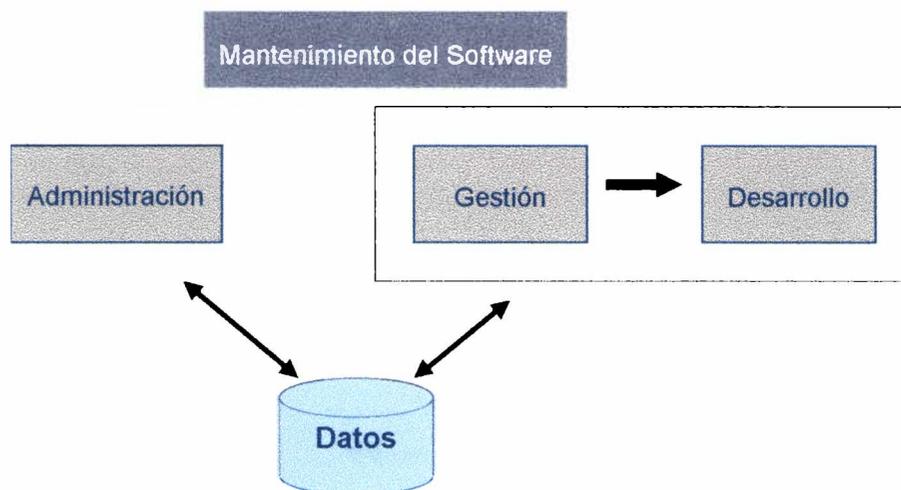


Figura 1

El flujo de información en este workflow comienza con la llegada de una solicitud de Pedido de Cambio al estadio **Gestión**, donde se evalúa la viabilidad del pedido en cuanto a costo económico, disponibilidad personal, relevancia de la solicitud, impacto en las distintas áreas de la empresa y demás cuestiones relacionadas. En este estadio se puede determinar que el pedido de cambio se rechaza o se puede determinar que se acepta el pedido, con lo cual se origina el flujo de datos hacia el estadio **Desarrollo**. La información que se transfiere está compuesta por los componentes y la documentación de la versión del software que se pretende modificar, más los datos de la solicitud del cambio pretendido, a fin de que en esta última parte se desarrolle el cambio del software. El módulo **Administración** provee al workflow de los datos necesarios para su funcionamiento.

Las dos primeras etapas de este workflow conforman, a su vez, cada uno un workflow, con sus propios estadios y flujo de información.

El módulo de Administración

La tarea del módulo de administración es incorporar, mantener y relacionar los datos con los que trabajan los otros dos módulos de la solución.

Las entidades con las que trabajan los módulos y que son administradas por este incluyen:

- los proyectos a ser mantenidos por la herramienta,
- el personal de la empresa que va a mantener esos proyectos,

- los pedidos de cambio recibidos para esos proyectos,
- los clientes que usan el software producido.

En este módulo se establece la restricción de acceso a la herramienta, determinando quiénes se consideran actores, quiénes pueden actuar como administradores, qué tareas podrán realizar en el módulo de gestión y con qué niveles de acceso, y quiénes pueden actuar en el módulo de desarrollo.

Sólo las personas que se incorporen como actores podrán usar la herramienta y sus actividades estarán acotadas en principio por el perfil que se les asigne. El perfil de un actor está formado por el conjunto de roles que puede desempeñar en el módulo de gestión, la habilitación o no para trabajar en el módulo de desarrollo, y por la habilitación o no a la administración de la herramienta.

Para el módulo de gestión de pedidos de cambio, los roles que pueden formar un perfil surgen de la relación muchos a muchos entre estos dos conjuntos (figura 2).

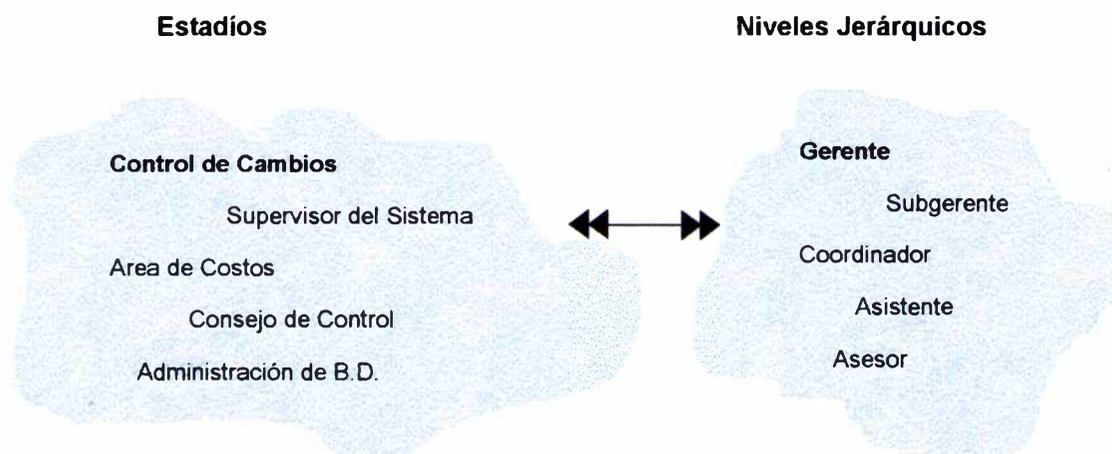


Figura 2

Además, se vincula a los actores con los proyectos que éstos van a atender, y con un rol particular, seleccionado entre los que figuran en su perfil.

Así, un actor queda ligado a un proyecto con uno o más roles (estos roles determinan estadio y nivel jerárquico). Su labor sobre el proyecto de marras, estará acotada a los permisos que le conceda el rol de mayor jerarquía que se le haya definido. Desde el punto de vista del proyecto, esta asignación define quiénes serán los encargados de la gestión de los pedidos de cambios que reciba.

Para trabajar como administrador de la herramienta, en cambio, basta con estar autorizado para ello, luego, el acceso es irrestricto dentro del módulo.

Con respecto a los permisos de acceso para el módulo de desarrollo, desde el módulo de administración sólo se determina si un actor es elegible o no para trabajar en desarrollo. Pero su verdadero accionar estará dado por su incorporación a un equipo de trabajo para el desarrollo de un cambio. Pero ésta última, no es una tarea del módulo de administración; es el Consejo de control el que establece qué actor se desempeñará en cada tarea. En cuanto a la jerarquía, sólo hay dos niveles determinados por la cualidad de líder o no que se le asigne a cada actor.

El módulo de gestión del cambio.

El objetivo de este módulo es realizar un estudio preliminar que permita definir la viabilidad del desarrollo del cambio. En el workflow que sustenta a este módulo se identifican cinco estadios, a través de los que fluye la información que se va generando a partir de un pedido de cambio que ingresa, para que finalmente se determine si se da curso al desarrollo del cambio o no.

A continuación, se definen los estadios y se muestran gráficamente en la figura 3, junto con el flujo de documentos entre ellos.

Controlador de cambios: los actores de este estadio son los responsables del primer análisis sobre lo pedido. En particular deben chequear que la información esté completa y acompañada de un documento de autorización de pedido válido. Asimismo deben archivar la solicitud y la autorización correspondiente. La solución establece que una vez que se determina que el pedido fue correctamente ingresado, la autoridad del estadio de control debe comunicar al grupo supervisor de cambios para el proyecto en cuestión, que hay un nuevo pedido para analizar, y facilitarles una copia de la solicitud.

Supervisión del sistema: este grupo conoce el proyecto que se pretende modificar, por eso es capaz de determinar a partir de la solicitud de cambio, análisis y proyecciones, cuáles son los componentes del sistema que se verán afectados y en qué medida. Por eso también pueden estimar la calidad y la cantidad de especialistas necesarios para resolver el cambio, y el tiempo que deberán dedicar a la tarea.

Basándose en estas estimaciones, el área de supervisión debe elaborar un documento informando al área de costos la cantidad de recursos humanos necesarios, y otro documento que recibirán el área de administración de base de datos y el consejo de control, en el que consta la magnitud y el impacto del cambio. Mediante estos documentos, se da participación en la gestión a los estadios Área de costos, Administración de base de datos y Consejo de control, los cuales pueden comenzar su evaluación.

Administración de bases de datos: este grupo trabaja teniendo en cuenta el documento que describe el impacto del cambio en los módulos componentes del sistema, para analizar y definir cuál es el impacto sobre las bases de datos. Medir el impacto sobre las bases de datos es fundamental, dado que las modificaciones en los datos podrían determinar la necesidad de cambios en otras aplicaciones que comparten esos datos. Como resultado del análisis en este estadio se generan dos documentos. Uno detallando los datos que se ven afectados, indicando qué otras aplicaciones los usan. Este documento se envía al Consejo de control. El otro documento, se dirige al Área de costos, e indica la cantidad y calidad de los recursos humanos necesarios, y el tiempo que se requiere de cada uno.

Área de costos: su análisis se nutre de los documentos generados por la Administración de bases de datos y el área de Supervisión del sistema, donde se detallan las necesidades de recursos humanos. El resultado de los cálculos es una estimación de la inversión necesaria en caso de que se apruebe el desarrollo del cambio. Genera un documento de impacto económico para informar al Consejo de control.

Consejo de control: es el responsable de tomar la decisión respecto de la suerte que correrá el pedido de cambio. Para evaluar la viabilidad del cambio, los encargados de este estadio cuentan con los documentos que detallan el impacto del cambio en los componentes del sistema, el impacto sobre los datos (y posiblemente sobre otras aplicaciones) y el impacto económico. Si el resultado de la evaluación

determina que se rechaza el pedido de cambio, el consejo genera un documento fundamentando tal determinación. Si, se determina que el cambio es viable, es el consejo de control el encargado de definir el equipo de desarrollo del cambio, los líderes de cada tarea, y el cronograma de tareas, indicando la fecha de comienzo y la duración de las tareas involucradas en el cambio.

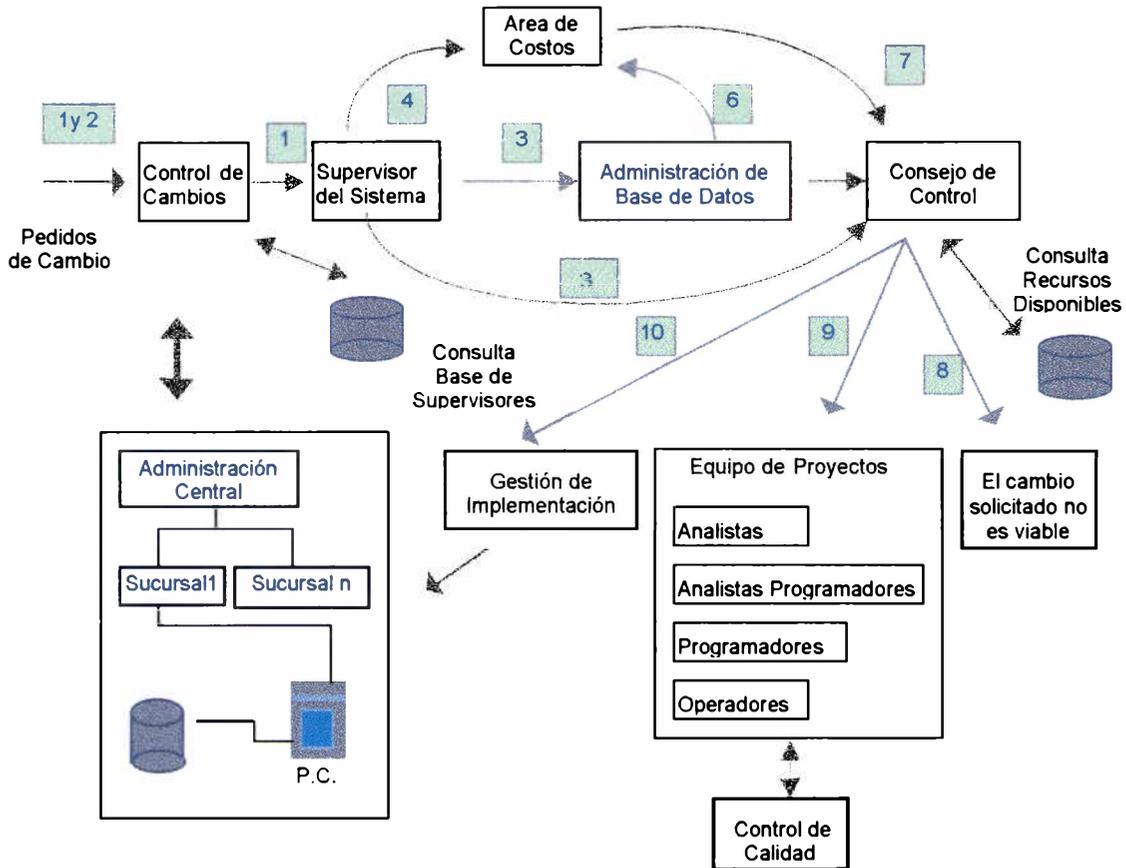


Figura 3

El grafo de estadios para este módulo es estático. Vale decir que todos los pedidos de cambio reciben el mismo tratamiento. No obstante, los actores de cada estadio se definen para cada proyecto. De esta manera en los extremos, puede haber tantos grupos de gestión de cambios como proyectos tenga la empresa, o puede definirse el mismo grupo de gestión para todos los proyectos. Esta flexibilidad permite que la herramienta se adapte a empresas o grupos de desarrollo de soft de distinta envergadura.

Se podría pensar en una asignación de personal de gestión a un nivel más bajo aún, esto es, por cada pedido de cambio armar el grupo de gestión, lo que genera un control descentralizado de los pedidos de cambio de un proyecto. Para la herramienta se eligió el enfoque que centraliza el control de los pedidos de cambio de un proyecto en sólo un grupo de personas, por considerar que de esta manera el análisis preliminar se hace más ágil por estar en manos de personas que conocen en profundidad el proyecto sobre el que se solicita el cambio, y además están al tanto de qué otros cambios se están desarrollando para el mismo proyecto.

Para cada estadio de la gestión de cambios de un proyecto, el administrador de la herramienta debe asociar como mínimo a un actor. De los actores asociados a cada estadio, debe haber por lo menos uno que cumpla el nivel jerárquico de gerente. Luego de respetar estas restricciones, el equipo de trabajo para cada estadio puede armarse como se considere necesario. El objeto de esta asignación es dejar en manos

de los gerentes la supervisión final del trabajo y la confirmación de la finalización de la tarea relativa al estado. No obstante el trabajo y las consultas necesarias para realizarlo pueden ser tarea de cualquiera de los integrantes del grupo.

El módulo de desarrollo

Metodología de Ingeniería de Software adoptada.

Un requisito para que la herramienta pueda ser utilizada con éxito, es que el software que se va a mantener haya sido desarrollado con una metodología de ingeniería que garantice que cada sistema cuenta con la documentación necesaria, como punto de partida en el desarrollo de un cambio.

Para diseñar la herramienta fue necesario definir un conjunto de tareas que conforman la metodología de desarrollo, ya que el mismo sirve de base para definir las tareas que se llevan a cabo durante el cambio.

El conjunto propuesto surge de reunir las tareas que son denominador común entre las tareas utilizadas por las metodologías analizadas. Vale decir, que todas ellas se utilizan, en mayor o menor grado de detalle, en las metodologías típicas de desarrollo de software. Si bien el grupo de tareas propuesto puede resultar el necesario y suficiente para los objetivos que persigue la herramienta, es cierto que se podría realizar una extensión para que la herramienta adopte el conjunto de tareas de la metodología de Ingeniería que utiliza la empresa que la va a usar para mantener su software.

Las tareas componentes son:

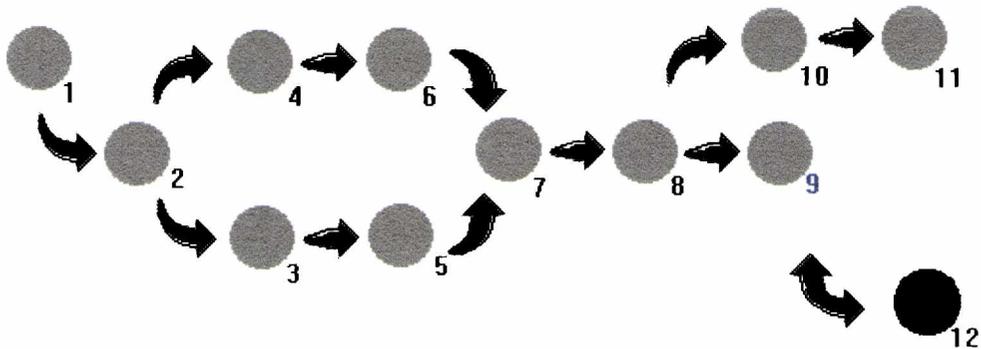
- especificación del sistema (que en la etapa de mantenimiento se traduce como especificación del cambio)
- especificación de requisitos del sistema (y su equivalente en mantenimiento: especificación de requisitos del cambio)
- diseño preliminar de datos
- diseño preliminar de estructura
- diseño detallado de datos
- diseño detallado de estructura
- codificación
- prueba
- desarrollo del manual del usuario
- desarrollo de manuales de operación e instalación
- instalación

Utilización de la metodología en el módulo de desarrollo

El grafo de tareas del workflow que sustenta a la etapa de desarrollo del cambio, tiene una estructura que se define como semidinámica. Si bien el consejo de control no tiene la libertad de elegir qué tareas de la metodología de Ingeniería de software adoptada serán necesarias para el desarrollo y cuáles no, es el tipo del cambio solicitado el que determina cuál es el grafo a seguir, entre tres posibles, predefinidos en la etapa de diseño de la herramienta.

Para los cambios de tipo adaptativo, perfectivo y preventivo, el grafo incluye un estadio para cada tarea del conjunto base definido anteriormente, el cuál se puede ver en la figura 4.

Estos tipos de cambio, en general generan en una nueva versión del soft. Por eso es que se involucran las mismas tareas que en el desarrollo del soft original. Es necesario garantizar que la nueva versión va a quedar perfectamente documentada.



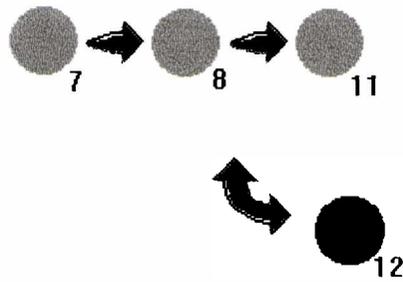
Referencias:	
1-Especificación del cambio	7-Codificación
2-Especificación de requisitos del cambio	8-Pueba
3-Diseño preliminar de datos	9-Generación de Manual del Usuario
4-Diseño preliminar de estructura	10-Generación de Manuales de Operación e Instalación
5-Diseño detallado de datos	11-Instalación
6-Diseño detallado de estructura	12-Supervisión del desarrollo

Figura 4

Para los cambios de tipo correctivo, el grafo incluye el conjunto de tareas estrictamente necesario para salvar rápidamente el "bug": una vez identificado el error, es necesario generar el código correcto, probarlo e instalarlo (figura 5).

Cuando el cambio implica sólo un cambio de versión del sistema en una o más áreas de la empresa cliente, basta con analizar las características que ofrece el sector y las características requeridas por la versión nueva (que constan en el manual de instalación), para determinar si es posible instalar o para determinar qué elementos faltan. Una vez definidos los lugares donde efectivamente se instalará y las versiones que se instalarán, se puede armar un cronograma que planifique la tarea. Para esto, sólo se necesita la intervención del grupo de instalación.(figura 6)

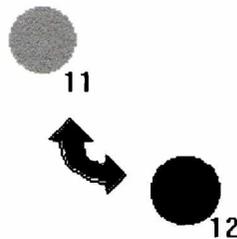
Esta tarea estará a cargo del responsable de mantenimiento definido al momento de introducir el proyecto en el módulo de administración. De esta manera hay sólo una persona a cargo del control de las versiones en uso de un sistema

**Referencias:**

- 7-Codificación**
- 8-Pueba**
- 11-Instalación**
- 12-Supervisión del desarrollo**

Figura 5

Cualquiera sea el grafo, es necesario que el consejo de control asigne un líder, un equipo de trabajo y una duración a todas y cada una de las tareas. El desarrollo del cambio no podrá saltar ninguno de los pasos previstos, a fin de asegurar la completitud del trabajo (el cambio en sí mismo más la documentación asociada).

**Referencias:**

- 11-Instalación**
- 12-Supervisión del desarrollo**

Figura 6

Cuando el cambio realizado al soft genera una nueva versión el resultado de este módulo es un conjunto de nuevos componentes (fuentes, ejecutables, manuales) para la versión nueva y la documentación que describe a la nueva versión.

Implementación de la Herramienta

La herramienta tiene por objeto apoyar la fase final del desarrollo de software, el mantenimiento, facilitando la documentación de programas, el control de versionado, administrando la asignación de recursos y sobre todo automatizando la comunicación entre los encargados del mantenimiento, a fin de agilizar la tarea, para evitar demoras en la etapa del ciclo de vida del software más costosa para las empresas.

Así, la herramienta prevé la administración de los recursos que posee la empresa para el mantenimiento (proyectos, clientes y empleados) y por otro lado, el control detallado de las actividades de modificación del software que comienzan con la aprobación del cambio. Por lo cual, en ella se distinguen dos módulos principales:

- Administración de los recursos
- Seguimiento de los cambios a los proyectos

Administración de recursos

La información que maneja la empresa en el proceso de mantenimiento de los sistemas, está relacionada con los proyectos que deben mantenerse, los clientes que utilizan los sistemas y el “staff del cambio”, es decir los encargados de producir el cambio. Como se puede deducir, esta información es vital, por lo tanto debe ser administrada en un modo consistente y confiable, teniendo en cuenta los controles de seguridad necesarios para otorgarle el nivel confidencial que requiere.

Así, las tareas de este módulo, estarán a cargo del Administrador del Workflow de Mantenimiento, y sólo él podrá modificar dicha información.

Como se puede ver en la figura1, la tarea del administrador consiste en mantener actualizada toda la información referenciada en los párrafos anteriores, es decir, que podrá realizar altas, bajas y modificaciones. Los datos que se manejan en este módulo se dividen en tres opciones generales, a saber:

- proyectos protagonistas del cambio
- clientes que usan los proyectos
- staff encargado del mantenimiento.

Proyectos protagonistas del cambio

Cuando un proyecto ingresa al sistema, ya está listo para ser instalado en las oficinas del cliente, es justamente en ese momento cuando comienza la etapa del mantenimiento, y su vida dentro de la herramienta. Para dar de alta un proyecto es necesario ingresar el nombre, una breve descripción de sus objetivos, el responsable del desarrollo del mismo, es decir, la persona que tuvo a su cargo todas las etapas anteriores en el ciclo de vida, como así también el responsable de mantenimiento, la persona encargada de supervisar las futuras modificaciones. (figura 2)

Además es necesario, identificar los **documentos** de todas las etapas anteriores el ciclo de vida del proyecto para que estén disponibles cuando sean

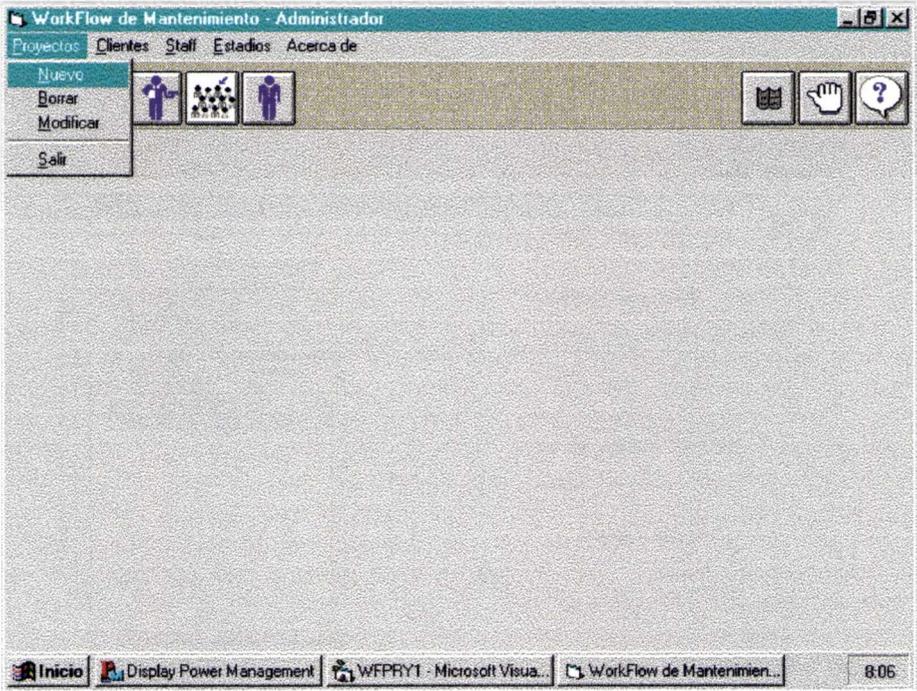


Figura 1

necesarios en el desarrollo de los cambios. Seleccionando el nombre del documento se conocerá su descripción, etapa del ciclo de vida a la que pertenece, además se debe ingresar el nombre del autor, la fecha de creación y la ubicación física del archivo.

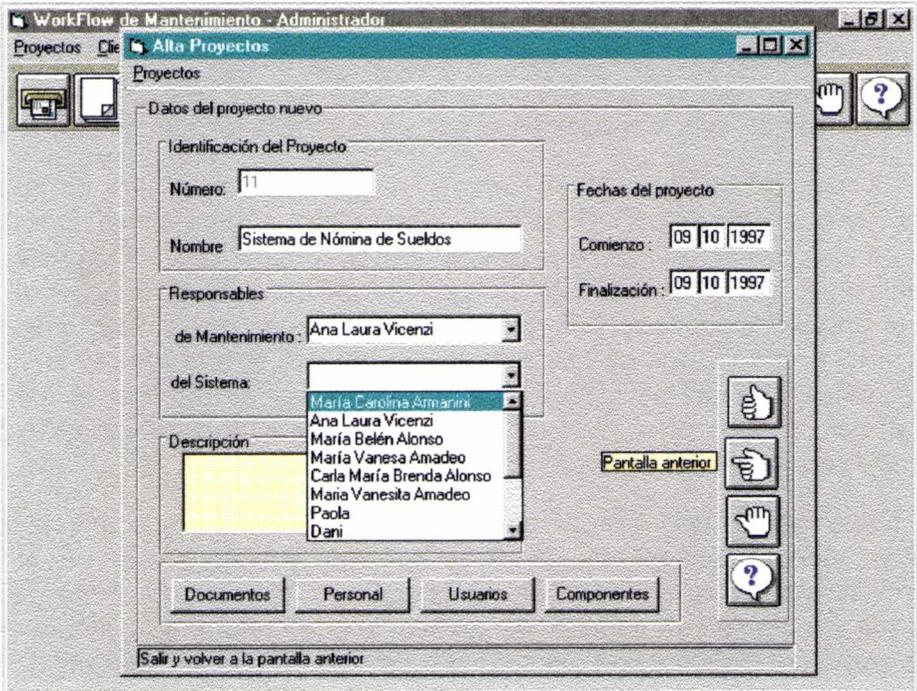


Figura 2

Es el Administrador también el encargado de asignar en este momento el personal que se encargará de gestionar los futuros pedidos de cambio. Debe seleccionar los participantes para cada uno de los estadios mencionados en el planteo de la solución: control de cambios, supervisión del sistema, administración de Bases de Datos, área de costos y consejo de control. Además, a cada uno de los

participantes se les define el nivel jerárquico (gerente, sub-gerente, asistente, asesor o coordinador), configurando así el rol que desempeñará en el estadio en cuestión. Esto será posible, siempre y cuando estén autorizados para dicho rol. (figura3)

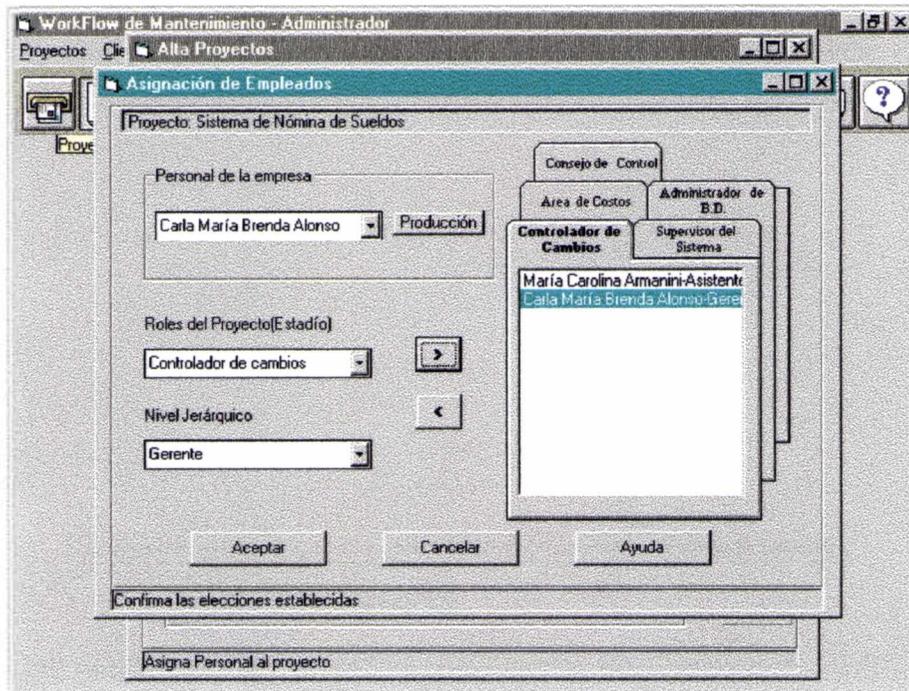


Figura 3

Cabe aclarar que la estructura del workflow de Gestión es fija, esto es, no se puede incorporar o eliminar etapas dependiendo de la empresa que instale el software, ni tampoco modificar la estructura dentro de una misma empresa, dependiendo del pedido de cambio del que se trate. Por esta razón, cada estadio debe contar con al menos una persona encargada del nivel más alto en la jerarquía.

La estructura fija no se ve como una restricción a la hora de la gestión del cambio, ya que por pequeña que sea la empresa, o por insignificante que sea el cambio, cada uno de los aspectos que se tratan en cada uno de los estadios deben ser analizados, si se pretende tomar una decisión con fundamentos al aceptar o rechazar un pedido de cambio. Por otra parte no hay ninguna restricción que impida que la misma persona realice tareas de más de una etapa de la herramienta. En un caso extremo, la misma persona podrá ser la encargada de todos los estadios y ella sola trabajar en cada uno. El sistema le servirá para tener el trabajo organizado y clasificado según se evolucione a través del workflow.

Cada estadio podrá organizarse jerárquicamente como la empresa considere necesario, independientemente de cómo se organizan los demás. Esta flexibilidad está prevista para que el sistema no imponga restricciones a la hora de formar los grupos para la gestión de pedidos de cambio, lo que generaría un inconveniente más en una etapa del ciclo de vida del software que es intrínsecamente conflictiva. Así, si la empresa considera que debido al caudal de trabajo le será suficiente contar con sólo una persona que controle los pedidos de cambio que llegan, podrá organizar el estadio de Control de Cambios con sólo un encargado. Si en cambio considera que el análisis del impacto sobre las bases de datos resulta crítico, podrá destinar más recursos en el estadio de Administración de Bases de Datos, con el organigrama que considere adecuado.

Por último, se solicita la información de los **clientes** que trabajan con el proyecto de marras. En este punto se requiere el nombre del cliente, que se puede seleccionar de la lista de clientes de la empresa, la oficina donde se instaló o se instalará el proyecto y el empleado del cliente encargado de manipular el sistema.

Toda la información anterior puede ser modificada por el Administrador, en el futuro siempre que no se infrinjan ciertas reglas como por ejemplo, desafectar un empleado que está desarrollando alguna tarea para ese proyecto.

Cientes que usan los proyectos

Se consideran clientes a aquellas empresas que utilizan en sus oficinas el software producido. Sus datos son necesarios para poder identificar quién solicita las modificaciones como así también para guardar un registro de los lugares donde están instaladas las distintas versiones de los proyectos.

Del mismo modo que en el punto anterior, el único que tiene acceso para modificar esta información es el Administrador.

Para dar el alta de un cliente es necesario ingresar su razón social, el domicilio, su número de identificación tributaria (C.U.I.T) y el nombre, cargo y documento del empleado responsable de mantener la comunicación entre el cliente y la empresa. Siendo indispensable el ingreso de la razón social y el número de identificación tributaria.

En cualquier momento se pueden consultar todos estos datos seleccionando el nombre de una lista de clientes y se mostrará una pantalla como la de la figura 4.

The screenshot shows a software window titled "WorkFlow de Mantenimiento - Administrador" with a tab labeled "Cientes". The main area contains a dropdown menu with "Gruelin Martinez & Asociados" selected. Below this, there are several input fields organized into sections:

- Domicilio:** Dirección: "Virrey del Pino 12365", Teléfono: "(0954) 23- 5100", Fax: "23- 5100".
- Responsable ante la empresa:** Nombre: "Esteban Dolero", Documento: "dni: 22.132.68", Cargo: "Gerente de Marketing".
- Identificación Tributaria:** C.U.I.T: "12.323.234", Responsable: "No inscripto".

On the right side, there is a "Sistemas" button and three icons: a hand pointing to the right, a hand pointing to the left, and a question mark. At the bottom, there is a label "Los sistemas que utiliza el cliente".

Figura 4

Para la modificación de los datos de un cliente también se pide el ingreso del nombre del cliente y a partir de esto, el administrador tendrá plena libertad para efectuar cambios sobre la información.

El dar de baja un cliente, siempre por medio de su nombre, desencadena un conjunto de acciones tendientes a eliminar todos los sistemas que tenía instalados el cliente. Se puede ver que es importante estar seguros de que el cliente deja de estar conectado totalmente con la empresa, y no es que sólo quiere dar de baja sólo algún sistema, conservando otros. Este último caso, debería verse reflejado en la modificación de proyectos - clientes y no en la baja del cliente.

Staff encargado del mantenimiento

Todos los empleados de la empresa que se dedican a alguna de las tareas asociadas al mantenimiento, deben estar incluidos en este grupo, tanto los de gestión de los cambios como los encargados de la implementación. Por lo tanto hay mucha información en juego en este punto y sólo debería ser modificada por la persona adecuada, el Administrador.

Para dar de alta un actor del equipo de mantenimiento, es necesario ingresar el nombre, el tipo y número de documento, el sexo, domicilio, su categoría, para su posterior liquidación y el título que posee como se puede observar en la figura 5. Además de estos datos, es necesario e indispensable asignarle un nombre y una clave de entrada a la herramienta, junto con el grupo en el cuál se va a poder desempeñar, a saber, Administrador, Gestión del cambio o Desarrollo del cambio.

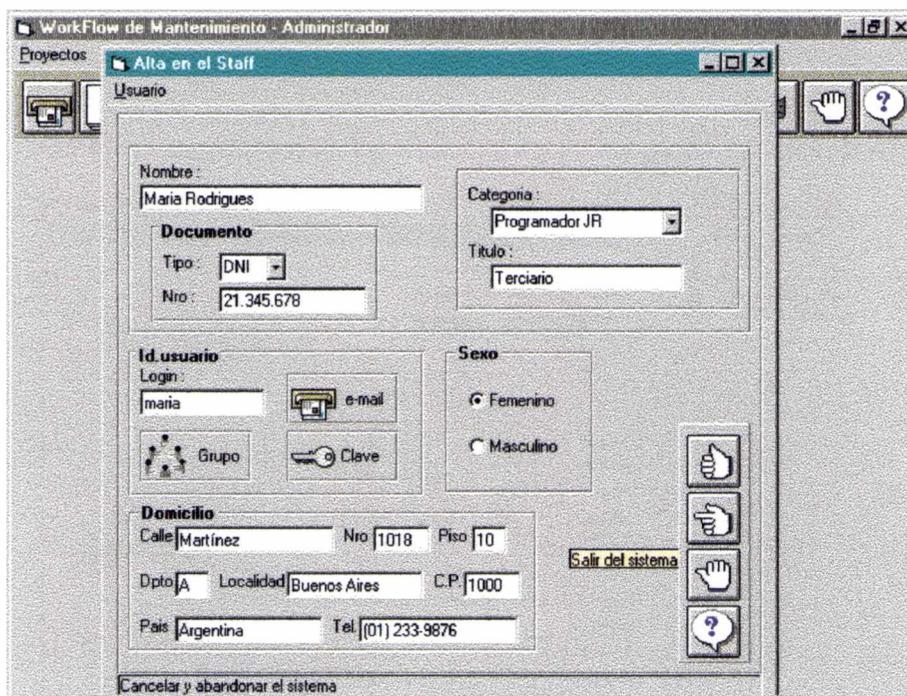


Figura 5

Dependiendo del grupo que se seleccione, hay un conjunto de pasos a seguir, si es Administrador o Desarrollo, la tarea concluye en ese momento. En cambio, si se selecciona gestión del cambio, se pasa a una pantalla en la cual se realiza la asignación de roles que puede desempeñar el empleado en cualquier proyecto que se encare. Al decir roles nos referimos a la etapa en el proceso más el nivel jerárquico dentro de éste. (figura 6).

Si el empleado se va a desempeñar en el desarrollo del cambio, no se registra en este momento en qué tareas lo hará, ya que esta asignación se realizará en el consejo de control cuando se defina el equipo de desarrollo para el cambio.

A fin de que el actor esté comunicado con el resto, para recibir, por ejemplo, los avisos informando sobre trabajos pendientes o tareas atrasadas en este punto, se define también una cuenta en el correo electrónico para el empleado preferentemente con el mismo nombre (login) y clave de entrada (password) que se le defina en el sistema para que el manejo del correo sea transparente.

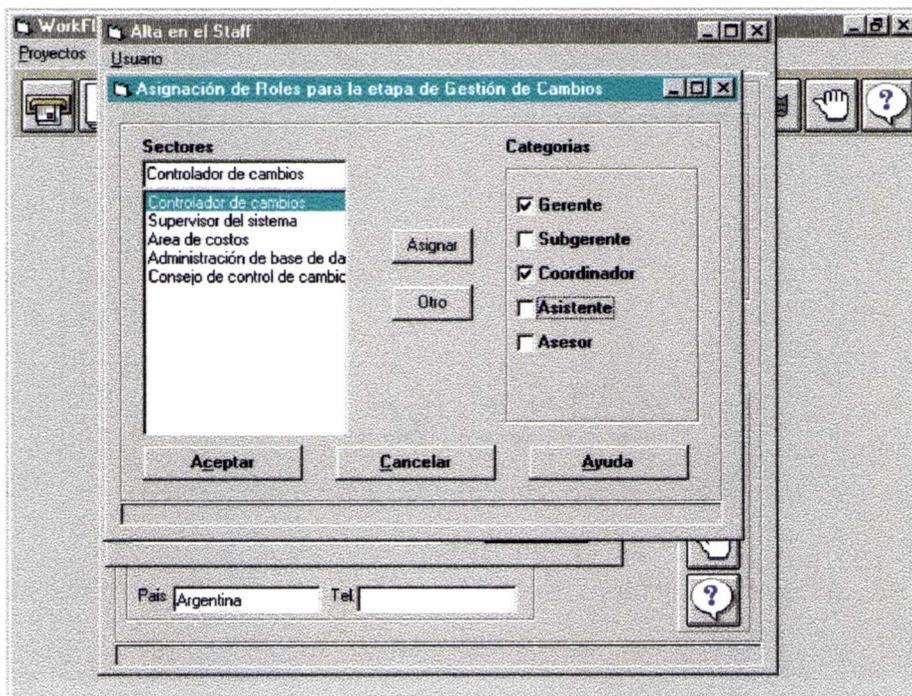


Figura 6

Por otro lado, se puede acceder a la lista completa de empleados como una forma de consulta (figura 7). En ella se muestran todos los actores agrupados por categoría y si se desea se pueden conocer los datos personales de cada uno en una pantalla similar a la de consulta individual (figura 5). Además, se pueden obtener los roles que puede desempeñar en la gestión del cambio, los datos de sistemas que tuvo asignados en el pasado, es decir su experiencia laboral en la empresa, como así también datos sobre los sistemas que tiene actualmente a su cargo.

En esta última opción también se pueden visualizar los empleados que están disponibles, es decir que no están ocupados en algún cambio de los proyectos a su cargo en este momento, en cada categoría para su futura asignación en los proyectos nuevos.

Seguimiento de los cambios a los proyectos.

Un pedido de cambio puede solicitar una modificación en el software del sistema o simplemente un cambio de versión en alguna o todas las oficinas del cliente.

Para cada pedido de cambio ingresado, la herramienta arma el soporte para dos sub-workflows, uno que soporta la gestión y otro el desarrollo del cambio. Los estadios del sub-workflow de desarrollo dependerán del tipo de cambio del que se trate.

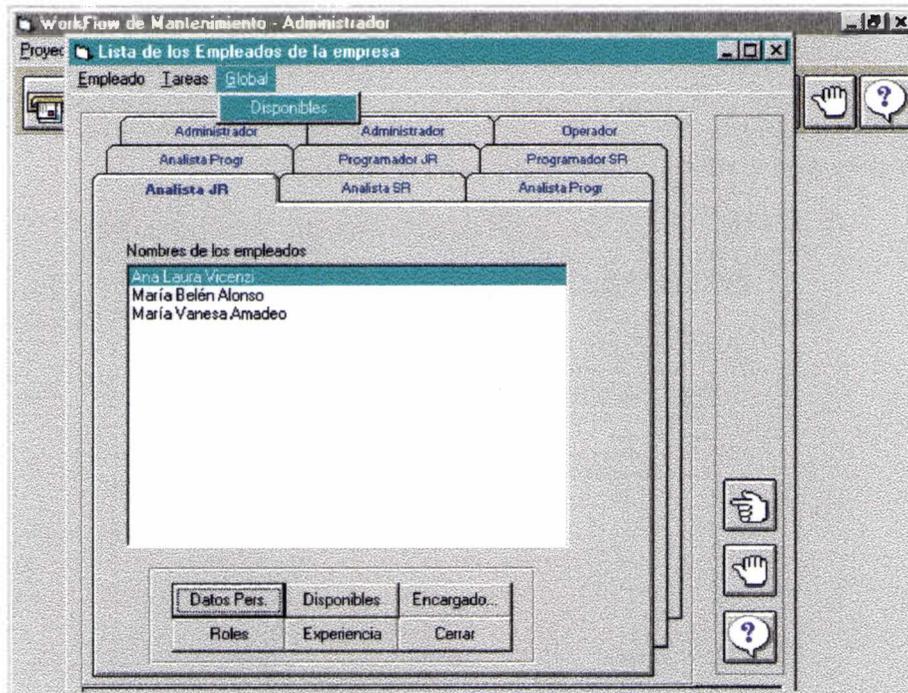


Figura 7

Así, para el seguimiento de los cambios se pueden considerar tres grandes módulos:

- gestión de cambios
- desarrollo de cambios

Gestión de cambios

La herramienta plantea el conjunto de estadíos mostrado en la figura 3 del capítulo anterior para soportar la incorporación de un pedido de cambio a un proyecto, su análisis y posterior aceptación o rechazo. A través de los distintos estadíos fluyen documentos a fin de informar y dejar debidamente documentado el fundamento de aceptación o rechazo del pedido.

En general, se puede decir que a cada uno de los estadíos se les asignan encargados (en el módulo de administración), para que realicen las tareas requeridas en él. Los encargados de cada tarea se pueden organizar con cualquier estructura jerárquica, con distinta cantidad de niveles, pero sólo los del nivel superior pueden aceptar o rechazar un pedido de cambio o dar por finalizada una tarea. Los de niveles inferiores podrán realizar el trabajo que requiere el estadío, (análisis y generación de documentos) pero no podrán confirmar la finalización del mismo.

En el comienzo de una sesión de este módulo hay una cierta similitud entre todos los roles, es decir que, sin importar el rol elegido, todos accederán a la misma pantalla inicial. Figura 8

En ella se listarán todos los proyectos que el empleado tiene asignados, con sus versiones, clasificados en:

- *proyectos en curso*: son aquellas versiones de proyectos que el usuario que está trabajando con la herramienta tiene a su

cargo (con cualquier nivel jerárquico) y cuentan con pedidos de cambio que están siendo analizados actualmente.

- *proyectos terminados*: son los proyectos con sus versiones que están a cargo del usuario para los que hubo pedidos de cambio que ya fueron analizados, es decir, que ya se desarrollaron todas las tareas de este estadio .

Desde aquí, el empleado puede obtener un conjunto de datos sobre el proyecto seleccionado, a saber: su identificación, nombre, descripción, fechas de inicio y fin del desarrollo, nombre y nombre de entrada al sistema del responsable de desarrollo del proyecto, y nombre y nombre de entrada al sistema del responsable del mantenimiento.

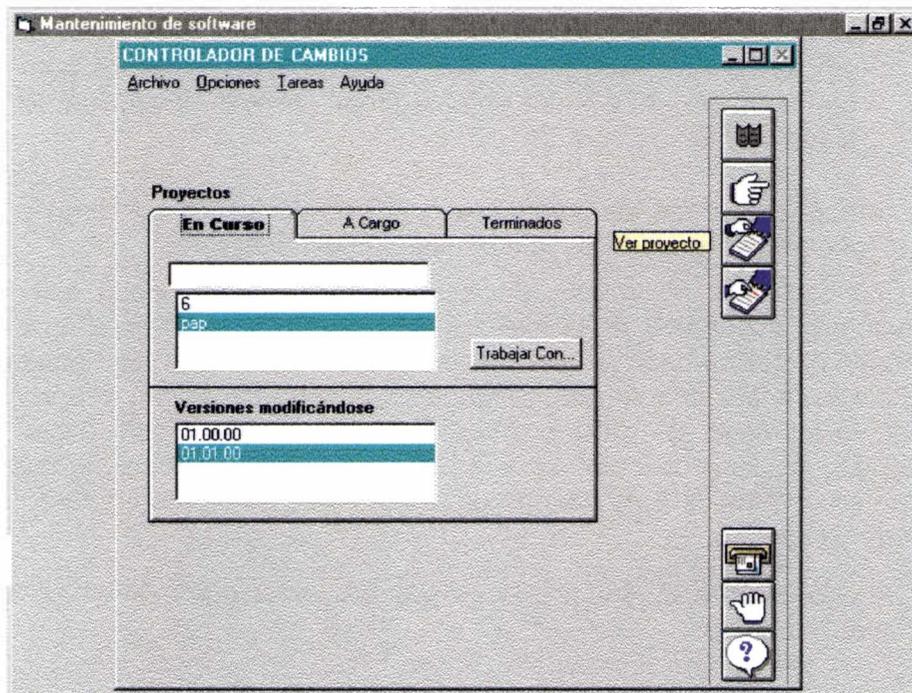


Figura 8

Por otro lado existe la posibilidad de obtener datos sobre la versión seleccionada tales como identificación de la versión, versión que le dio origen, número de cambio que la generó y descripción de la versión actual.

Además, están presentes las siguientes opciones, comunes a toda la herramienta:

- *salir del sistema*, que permite abandonar el sistema desde cualquier punto del mismo,
- *cambiar de perfil*, que permite abandonar la tarea actual y elegir otro rol para el mismo módulo o para otro, sin necesidad de salir del sistema y entrar nuevamente, y
- *mensajería*, por medio del cual el usuario puede acceder al Correo Electrónico basado en Microsoft Mail para ver sus mensajes, responderlos y enviar otros.

El botón de cambiar de perfil es útil cuando pensamos en una empresa de software de poca envergadura, en la cual seguramente, una misma persona deberá desarrollar más de una tarea en la etapa de mantenimiento.

Cuando el usuario selecciona un proyecto y su versión, tendrá acceso a los pedidos de cambio que le fueron asignados. Estos pedidos se agrupan en dos categorías para todos los estadios, salvo para los de control de cambios y consejo de control, que incorporan una tercera categoría, dado que sólo en ellos existe la posibilidad de rechazar pedidos:

- *pendientes*, son los pedidos que el usuario tiene a su cargo pero todavía no atendió o no culminó todas las acciones necesarias para cumplir con el.
- *terminados*, todos los pedidos que fueron analizados por el empleado y que su registro puede ser útil para revisarlos ante un nuevo pedido y poder obtener ideas sobre la resolución de algún problema.
- *rechazados* son los pedidos que el usuario tuvo a su cargo y que fueron rechazados como resultado del análisis.

Una vez que el trabajo de esta etapa ha terminado, quien tenga la autorización de encargado de la misma podrá confirmar la finalización de la etapa, momento en el cual se habilita a las siguientes en el workflow y sus encargados serán informados automáticamente por medio de un mail de tal hecho, a la vez que verán aparecer en las listas de trabajos nuevos, el pedido de cambio actual junto a la disponibilidad de los documentos necesarios para cumplimentar las tareas que se requieren en el estadio. Esta cadena de acciones es transparente para los usuarios de la herramienta.

Se analizarán a continuación los aspectos en los que difieren los estadios de la gestión del cambio.

El primer estadio del workflow fue denominado **Control de Cambios** (figura 9). En este punto se reciben los pedidos de cambio desde los distintos clientes con la correspondiente autorización. En el apéndice A se ve el diseño de los dos formularios para solicitud de modificaciones, el "Pedido de Cambio" y la "Autorización del Cambio".

Por medio de la opción Pedido Nuevo y el primer documento se incorporan al sistema los datos del pedido de cambio, para que ingrese al workflow. El segundo documento permite determinar la validez o no del pedido. Es necesario dar de alta en el sistema ambos documentos para posteriores consultas, utilizando la opción Documento Nuevo.

En esta etapa se evaluará la consistencia y completitud de los datos en ambos formularios, para ello se cuenta con la opción de Abrir Documento, que una vez que este fue cargado, me permitirá abrirlo una y otra vez, hasta que la tarea esté concluida, por medio de un enlace con el procesador de texto Word. En caso de no cumplir con los requisitos mencionados anteriormente, se rechaza el pedido de cambio y no continúa su flujo, hacia los estadios siguientes.

Una vez que el pedido es rechazado o aceptado e incorporado al sistema, se dará por finalizada esta etapa. En caso de ser aceptado, se habilita la tarea de Supervisión del Sistema como se explicó con anterioridad.

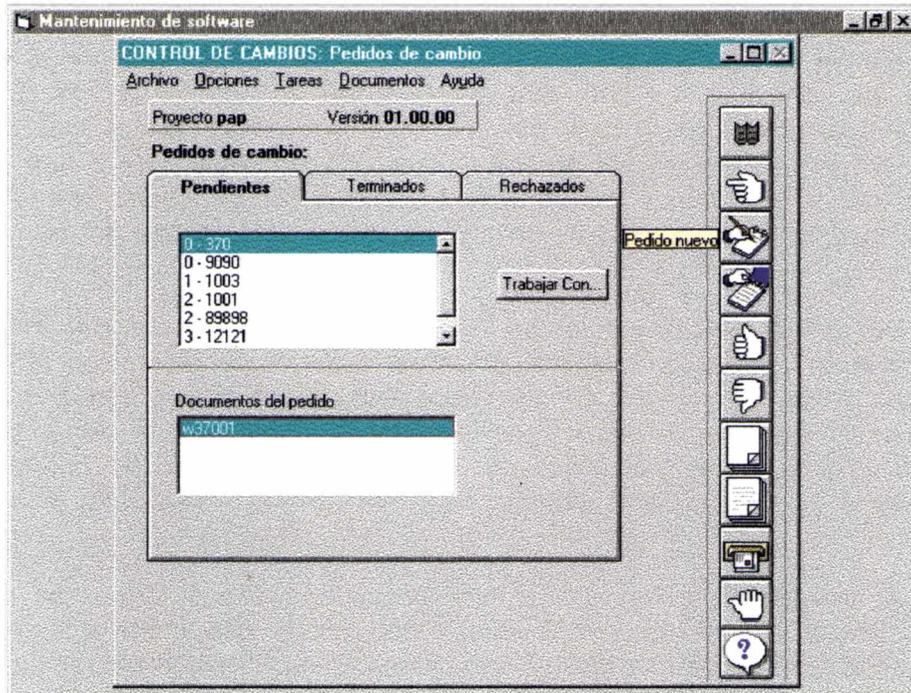


Figura 9

Tanto para la Supervisión del Sistema como para Administración de Bases de Datos y Area de Costos, la automatización de las tareas es similar debido a sus semejanzas en los pasos a seguir. Todos ellos necesitan abrir documentos creados en la etapa anterior (Abrir Documento) y generar nuevos (Documento Nuevo) que se enviarán a las etapas sucesivas para su posterior análisis (figura 10).

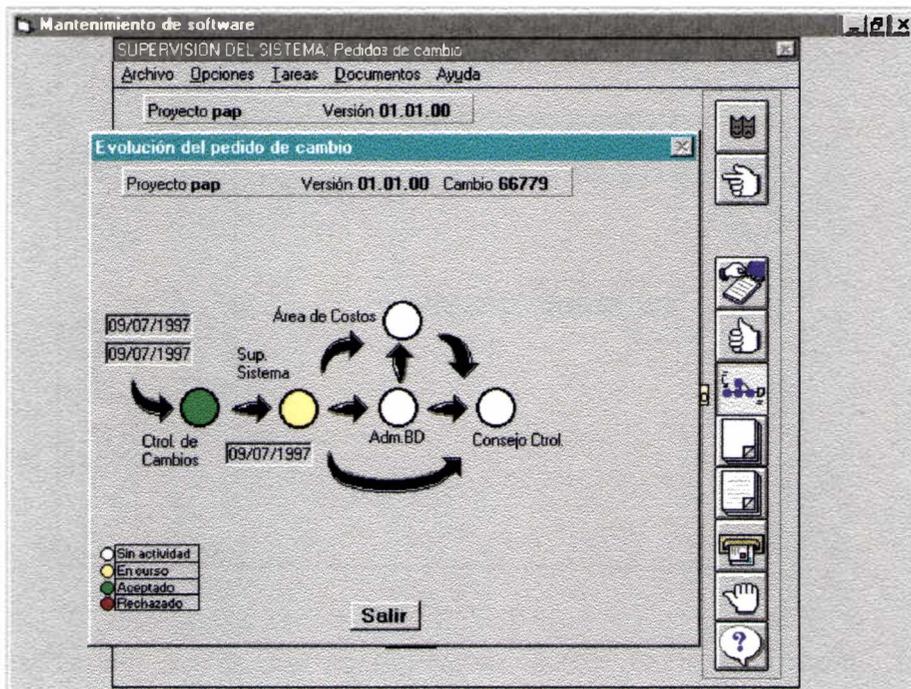


Figura 10

Hay, sin embargo un elemento distintivo en la Supervisión del Sistema. En esta etapa, se permite acceder a una pantalla para ver el estado del desarrollo de la gestión del cambio, a través de un mapa de tareas. Las tareas se verán de diferentes colores, acorde a su estado de evolución (sin actividad, en curso, etc.)

En la Supervisión del Sistema se accede al documento de Pedido de Cambio por medio de la opción Abrir Documento, del mismo modo que en el estadio de Control de Cambios. Por otro lado, se generan el tercer y cuarto documentos, "Informe técnico por pedido de cambio" e "Informe Técnico para el Area de Costos", respectivamente. Para ello se provee la opción Documento Nuevo que permite, la creación de los documentos a partir de las plantillas que se puede ver en el apéndice A, para facilitar la tarea del usuario y mantener la uniformidad del formato de los documentos.

Del mismo modo se manejan los documentos en las otra dos etapas. En la Administración de Bases de Datos, se abre el tercer documento y se generan el quinto y el sexto, "Informe del Administrador de Base de Datos por pedido de cambio" e "Informe del Administrador de Base de Datos para el Area de costos" respectivamente. Y, en el Area de Costos, se consultan el cuarto y quinto documentos y se genera el séptimo, "Informe del Area de Costos por pedido de cambio".

A diferencia de las etapas anteriores del workflow que eran meramente administrativas y se basaban en análisis y creación de documentos, los encargados de la etapa **Consejo de Control** tienen otros objetivos por alcanzar (figura 11). A partir de los informes generados en los demás estadios, se debe evaluar si el pedido de cambio que llega será aceptado o rechazado. En esta última etapa de la gestión, el pedido de cambio se evalúa desde un punto de vista global, esta decisión determina si el pedido prospera o se documenta su rechazo.

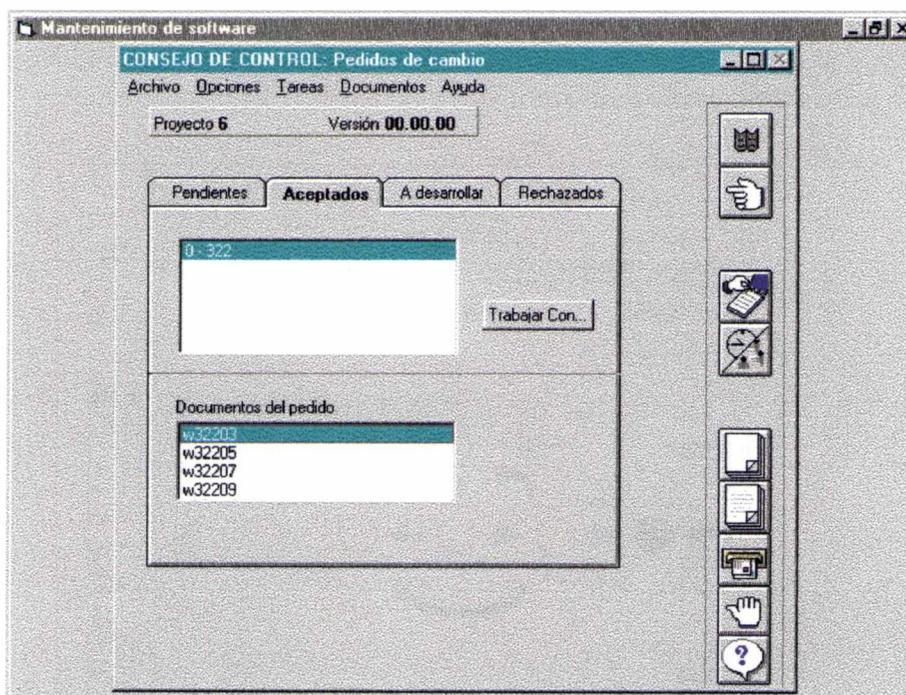


Figura 11

Cuando se acepta un pedido de cambio, en este paso se debe definir con qué recursos se desarrollará el cambio y en qué plazo de tiempo. El administrador de recursos permite efectuar estas asignaciones, todas sus funciones se explicarán en detalle más adelante.

Con la etapa del Consejo de Control se completarían todos los pasos de la Gestión de pedido de cambio. A partir de ahora, cada pedido aceptado ingresa al

workflow de desarrollo del mismo, donde un equipo de profesionales se encargarán de todas sus etapas y documentos. Este es el tema de la sección siguiente.

Desarrollo de cambios

Para la implementación de la herramienta se consideró, además de los cuatro tipos de mantenimiento ya definidos, el control de versiones como una función más de esta etapa, dando lugar a una forma más de mantenimiento:

- mantenimiento por cambio de versiones

Los cambios que originan a cada uno de los tipos de mantenimiento mencionados se diferencian por una serie de características: los motivos que causan el pedido de cambio, los efectos que produce el cambio al ser desarrollado y la urgencia o no en el desarrollo.

Cambios del mantenimiento correctivo:

se originan por un error en el soft que no fue detectado en las pruebas. Este tipo de cambios tiene por objeto corregir un defecto, y por lo tanto su solución debe ser rápida (a veces el programa cancela al llegar al error !!). Por otro lado no generan nuevas características en el soft, sólo hacen que el soft funcione como dice la especificación del sistema. Es por ello que este tipo de cambios no genera una nueva versión del soft.

Cambios del mantenimiento perfectivo:

pueden ser impulsados por los mismos programadores, que desean mejorar una parte del soft que, una vez terminada no los satisfizo. En general no son urgentes, ya que sólo reemplazan una parte del soft para que resulte más amigable para el usuario o para que resulte más fácil de mantener en el futuro. Así es que pueden generar nueva versión (si el cambio es visible para el usuario: cambian las pantallas, la forma de trabajo) o no (cuando la funcionalidad ofrecida y la interface al usuario no se modifican, sólo se modifica el código).

Cambios del mantenimiento adaptativo:

son cambios que el medio o las circunstancias obligan a realizar. Cuando las características del medio en el que el sistema está inmerso cambian, el sistema debe adaptarse a ellas. La urgencia estará dada por lo crítico o no que resulte seguir usando el sistema con las viejas características. Generalmente generan nuevas versiones, dado que los cambios pueden ser necesarios para un sector de una empresa que use el soft y para otro sector no, por ejemplo si los sectores pertenecen a distintas ciudades, con diferentes características.

Cambios del mantenimiento preventivo:

surgen cuando se advierte que alguno de los requerimientos que debía cumplir el sistema en un principio se advierte (a través de proyecciones futuras por

ejemplo) que se van a ver superados tempranamente. En estos casos se organiza un cambio preventivo, para evitar tener que realizar correcciones urgentes más adelante. El tiempo de desarrollo en general no es crítico y genera nuevas versiones de soft.

Cambios del mantenimiento por cambios de versión:

cuando se tienen disponibles varias versiones del soft, cada sector de la empresa que lo usa tendrá instalada la versión que se ajuste a las características de hard del sector y a las necesidades del sector. Cuando un sector tiene una versión "vieja" del soft, y se equipa con el hard necesario para poder correr una versión "más moderna"(por ejemplo), puede solicitar que se le cambie la versión. Nuestra empresa de desarrollo de soft sólo deberá planificar la instalación (teniendo en cuenta los manuales de operación e instalación), sin que se vea involucrada ninguna otra tarea.

Estos cambios no generan nuevas versiones de soft (sólo hacen que se instale una versión en lugar de otra), y es deseable que el cambio sea rápido.

En la práctica estos pedidos se realizan sobre la versión 00.00.00 (la versión original del proyecto).

Para este tipo de pedidos de cambio se incluye un párrafo especial más abajo.

Surge de las definiciones anteriores que el tratamiento de los cambios preventivos, perfectivos y adaptativos es similar. Mientras que los cambios correctivos y los cambios de versión tienen características distintivas. Así, se distinguen tres mapas para el workflow de desarrollo. Es decir, que los estadios de este segundo workflow no son fijos y dependen del tipo de cambio para el que se construya.

Se establece un conjunto de tareas a seguir para el desarrollo de los cambios. Esta selección se basó en el ciclo de vida de la mayoría de los proyectos, suponiendo que todos los cambios se implementarán por medio de una revisión y modificación de cada una de las etapas anteriores, quedando supeditada al proyecto y cambio en cuestión la duración de cada una de ellas.

Cuando un actor comienza su tarea en este workflow,(figura 12) se encuentra con una pantalla que presenta los proyectos que tiene a su cargo, clasificados según los pedidos de cambio existentes en las siguientes categorías:

- *nuevos*, los pedidos de cambio que arribaron al sistema y están listos para ser atendidos pero aún no los abrió ningún actor en este estadio.
- *a cargo*, la primera vez que un actor atiende un pedido pasa a este estado y permanece aquí hasta que se haya terminado.
- *terminados*, todos los pedidos que concluyeron su ciclo en esta etapa del workflow, pasando al siguiente estadio.
- *en revisión*, caen aquí ante un pedido de revisión de este estadio o de alguno anterior.

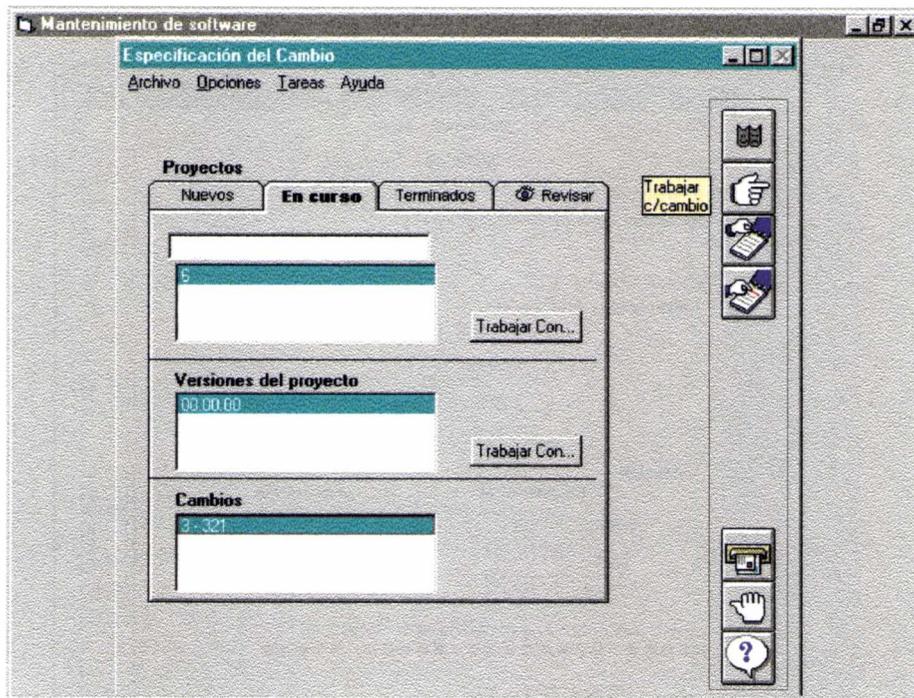


Figura 12

Antes de entrar a trabajar con un cambio, se pueden obtener datos sobre el proyecto y la versión que se encuentran seleccionados tales como: la identificación del proyecto, su nombre, descripción, fechas de inicio y fin del desarrollo, nombre y login del responsable de desarrollo del proyecto, y nombre y login del responsable del mantenimiento, y versión origen, número de cambio que la generó y descripción de la versión actual.

Además, están presentes las siguientes opciones, comunes a toda la herramienta:

- *salir del sistema*, que permite abandonar el sistema desde cualquier punto del mismo,
- *cambiar de perfil*, que permite abandonar la tarea actual y elegir otro rol para el mismo módulo o para otro, sin necesidad de salir del sistema y entrar nuevamente, y
- *mensajería*, por medio del cual el participante puede acceder al Correo Electrónico basado en Microsoft Mail para ver sus mensajes, responderlos y enviar otros.

Al seleccionar el cambio objetivo de la sesión, el actor accede a distintos módulos de trabajo, acorde con el estadio en el que desee trabajar o el perfil con el que ingresó a la herramienta, o ambos.

La mayoría de los estadios del Workflow de Desarrollo, cuenta con un formulario (figura 13) en el que se informa las fechas del estadio, es decir, las fechas tempranas, tardías y reales de inicio y fin del mismo. Las fechas de inicio y fin de cada tarea tanto tempranas como tardías, son calculadas por la herramienta en el PERT que se construye para cada cambio a partir de los datos proporcionados por el Consejo de Control en el workflow anterior y al cual tiene acceso el Supervisor del desarrollo.

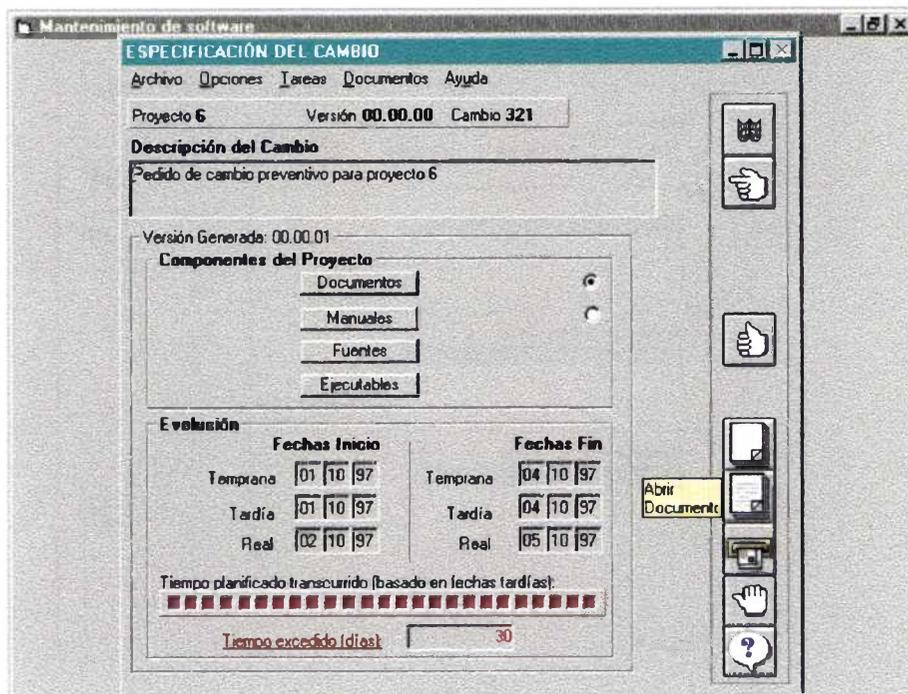


Figura 13

Por otro lado se informa la fecha de notificación, es decir, la fecha en que la herramienta comunica al actor sobre la existencia de la tarea pendiente. No necesariamente, el primer día de trabajo en ella si el actor estaba ocupado en otras actividades. Se muestra también el tiempo transcurrido desde el inicio, basándose en las fechas tardías, por medio de una barra estado. En dicha barra, se representa cuánto tiempo pasó y cuánto le resta en el desarrollo. En caso de haberse acabado el tiempo asignado, se pintará toda de rojo para llamar aún más la atención del actor y ponerlo sobre aviso del problema, indicando además la cantidad de días en que se superó el tiempo planificado.

Por supuesto, además de la información anterior, se tiene la posibilidad de abrir los documentos correspondientes a cada etapa y crear el documento resultante de cada una de ellas del mismo modo que en el workflow de Gestión.

Una vez finalizada la etapa, el encargado de ella podrá confirmar su finalización. A partir de este momento son habilitados los siguientes estadios en el workflow, informando a sus encargados automáticamente por medio de un mail que tienen una tarea pendiente. También, la herramienta administra los documentos de tal forma que cuando aparece un trabajo nuevo en las listas de pedidos pendientes, los actores tendrán a su alcance los documentos que necesita para cumplimentar las tareas que se requieren en el estadio.

Como última tarea del ciclo de vida del desarrollo del cambio, aparece la tarea de instalación del soft modificado. El líder de esta tarea es el Responsable de Mantenimiento, el cual después de un análisis de la situación de cada una de las oficinas del cliente y teniendo en cuenta los recursos de software y humanos, toma la decisión de instalar o no la nueva versión en cada una de ellas. Con dicha información, el encargado de la instalación debe completar un formulario como el que aparece en la figura 14 configurando así, el plan de instalación para el cliente.

Además, hay una tarea propia del mantenimiento, la tarea de **Supervisión del desarrollo**. Su líder o responsable es el empleado que tiene a su cargo el

mantenimiento del proyecto. El actor que ingresa al sistema como supervisor de desarrollo tendrá un amplio control sobre todo lo que sucede con los cambios de los proyecto que tiene a su cargo. Manejará información sobre el estado de las tareas, cuáles están atrasadas, cuáles son críticas y qué empleados están afectados a las mismas.

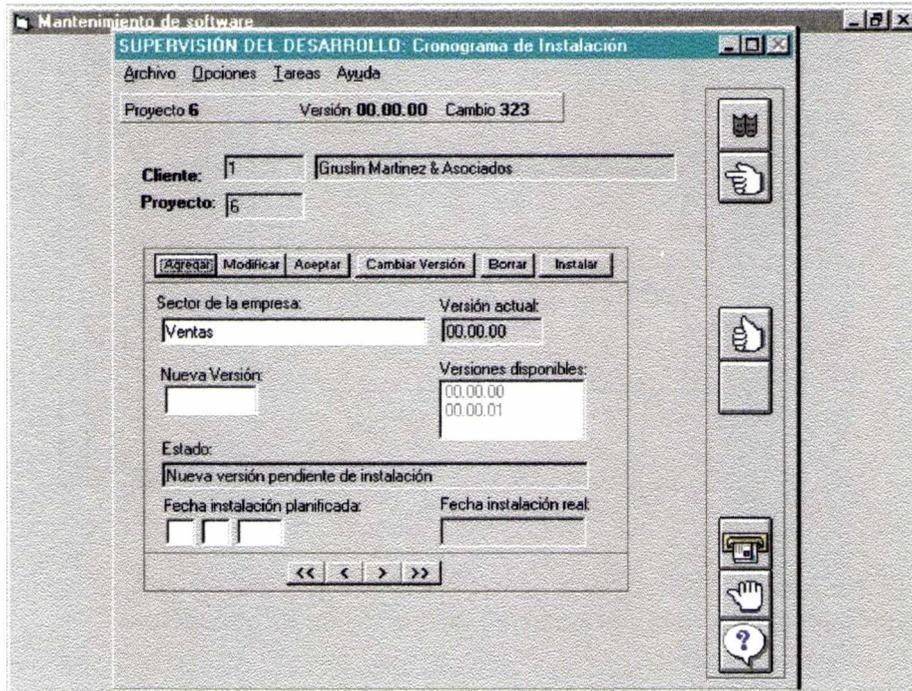


Figura 14

La selección de proyecto, versión y cambio sobre el que se va a trabajar se realiza en un modo similar a las tareas anteriores (figura 15), pero cuenta con la posibilidad de seleccionar de la lista de tareas aquellas que cumplan determinada condición por medio del botón de selección. Para realizar la clasificación, combina el tipo de tarea:

- críticas
- no críticas
- críticas / no críticas (todas)

junto con el estado del desarrollo de las mismas, que puede ser:

- atrasado
- sin atrasos.

Por medio de esta opción, el supervisor tiene la posibilidad de ver de un sólo golpe de vista, entre otras cosas, las tareas que necesitan mayor control y en cuáles tiene que apurar el desarrollo.

Una vez seleccionado el proyecto, versión y cambio objetivos de la sesión, el actor de este estadio contará con un diagrama, que representa la red de tareas del desarrollo de ese cambio (figura 16).

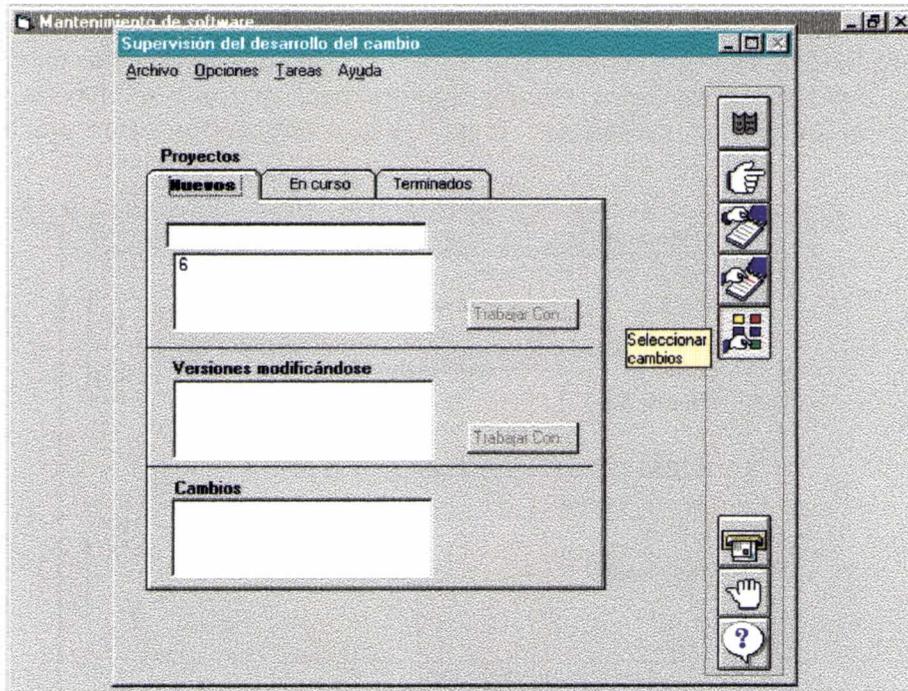
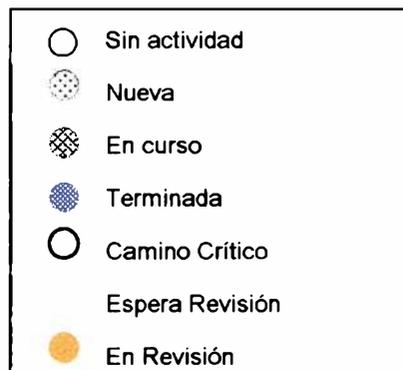


Figura 15

Dicha red de tareas permite ver en forma gráfica el estado de cada una de ellas distinguiendo los estados posibles utilizando diferentes colores.



Desde el rol de supervisor también se puede obtener un "zoom" de la tarea, para conocer datos tales como las fechas de inicio y fin, planificadas y reales, sus encargados y el estado de la misma.

Una de las tareas más relevantes del supervisor del desarrollo del cambio, es administrar las revisiones de las tareas terminadas. Estas revisiones son necesarias debido a que el desarrollo de soft es un proceso iterativo, que va depurando los resultados de las etapas a medida que avanza.

Cualquiera sea el grafo de tareas a seguir en el desarrollo de un cambio, las mismas se van habilitando conforme sus antecesoras terminan.

Cuando una tarea termina, su documentación no puede ser modificada, para que el/los sucesores trabajen a partir de elementos ciertos. Pero como se mencionó antes, los encargados de una tarea pueden advertir que los elementos provistos por las tareas previas les son insuficientes (por ejemplo porque hay errores, porque falta definir elementos o porque se advierte tardíamente una mejor forma de solución).

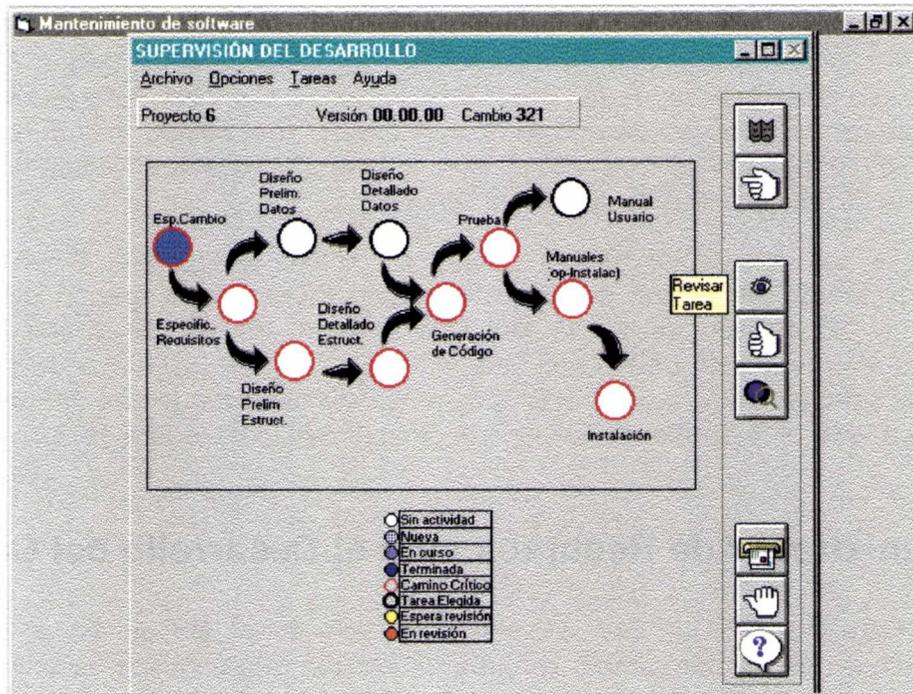


Figura 16

En esos casos, los encargados de la tarea corriente no pueden seguir con su trabajo hasta tanto no se solucione el problema con el que se encontraron. Pero los encargados de la tarea previa ya no tienen acceso a modificar la documentación (la tarea previa ya terminó). Entonces interviene el supervisor del desarrollo.

El equipo de trabajo que se encuentre con una dificultad como la mencionada, deberá comunicarse con el supervisor de desarrollo, para juntos determinar desde qué tarea previa es necesario rever el trabajo. Una vez determinado el lugar desde donde se debe revisar el trabajo, el supervisor marcará a la tarea como "en revisión" y enviará un mail a sus encargados. Esto implica que la tarea que había terminado, aparecerá en la columna de trabajos "a revisar" cuando se logonee algún miembro del equipo de trabajo.

Cuando una tarea se marca para revisar, todas las tareas que la tengan como predecesora (directa o indirecta) se verán afectadas: si habían terminado o si estaban en curso, también pasarán a revisión (pero no se habilitarán hasta tanto no termina la tarea previa).

El supervisor podrá marcar "para revisar" sólo tareas que ya hayan terminado, y sólo cuando las tareas que se vean afectadas por la revisión no estén ya involucradas en otra revisión previa.

Cuando una tarea revisada termina, se habilita el desarrollo o la revisión de las sucesoras (dependiendo de cuál haya sido su estado en el momento de marcar la revisión).

Control de Versiones

En esta sección de la herramienta, se tratarán aquellos pedidos de cambio que solicitan sólo un cambio de versión en el lugar de instalación. Es decir, que por ejemplo, el cliente compró equipo nuevo para alguna de sus oficinas y ya puede instalar la nueva versión.

Para desarrollar este tipo de cambio, no es necesaria ninguna modificación de los fuentes ni tampoco de los documentos del ciclo de vida del software. Se requiere sólo un profundo análisis de las nuevas condiciones planteadas por el cliente y del costo de instalar la nueva versión. En caso de aceptar el pedido, hay que desarrollar un nuevo cronograma de instalación y asignar personal para que ejecute las acciones tendientes a llevar a cabo el cambio de versión.

Así, un usuario que ingresa al sistema con este perfil, puede tener a su cargo sólo dos tareas de las del desarrollo completo, a saber, Instalación de la Nueva Versión y Supervisión del Desarrollo. Básicamente, trabaja del mismo modo que personal del desarrollo de los cambios (figuras 12 y 13). Dependiendo de la tarea específica tiene acceso a crear el Cronograma de Instalación (figura 14), en el primer caso, o a la Red de Tareas del Desarrollo en el otro estadio (figura 16).

Administrador de Recursos para el Desarrollo del Cambio

En el workflow de Gestión, más precisamente en el Consejo de Control, al aceptarse un pedido de cambio se debe definir con qué recursos se desarrollará el cambio solicitado y cuáles son los plazos de tiempo que se manejarán. El administrador de recursos permite efectuar estas asignaciones. (figura 17)

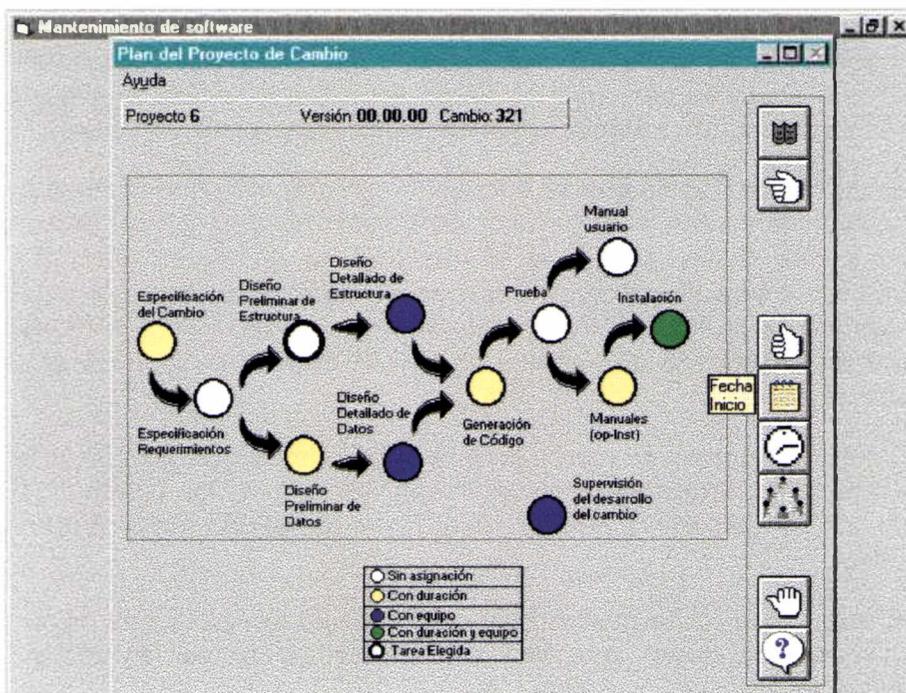


Figura 17

Como se puede ver en la figura, se muestra la red de tareas asociadas al desarrollo del cambio, donde cada tarea se relaciona con sus antecesoras y sucesoras. La relación de antecesoras y sucesoras se establece teniendo en cuenta la necesidad de resultados de otras tareas para ejecutar la actual y los resultados que brinda dicha tarea para que se ejecuten otras, respectivamente.

Al seleccionar una tarea se puede elegir entre los integrantes del staff, quiénes van a conformar el equipo de desarrollo para ella (figura 18). La asignación se puede realizar por medio de los botones o la técnica de "drag and drop", en la cual se debe seleccionar el integrante elegido y desplazarlo con el mouse hasta el lugar

deseado. También existe la posibilidad de deshacer estas asignaciones utilizando los botones o la técnica antes mencionada.

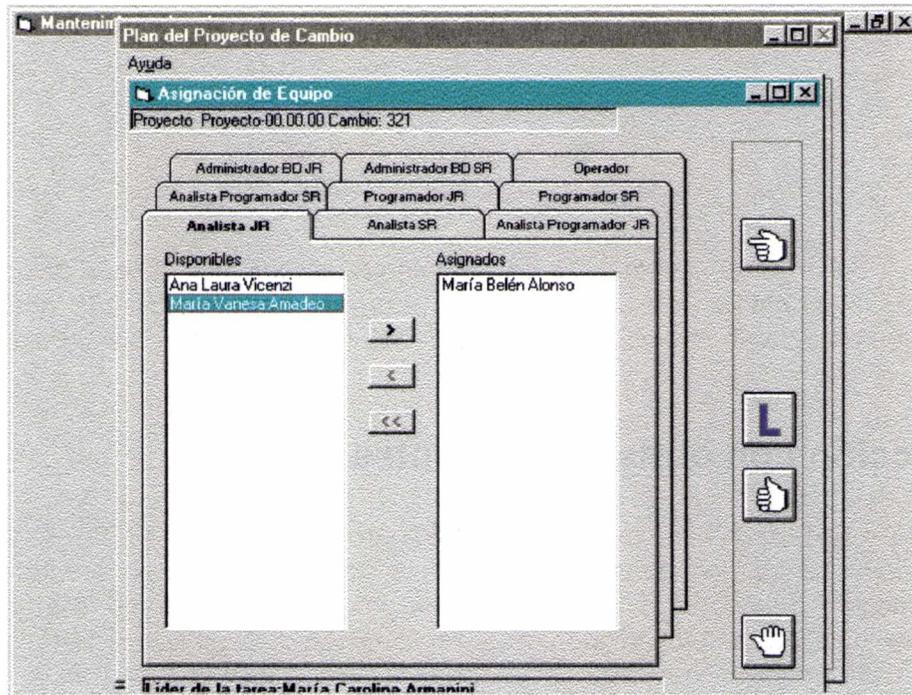


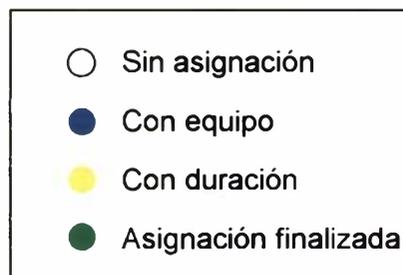
Figura 18

Cada tarea debe tener asignado un líder o encargado que tendrá a su cargo la coordinación de las acciones que en ella se desarrollan y la confirmación de finalización de la misma. Dicha asignación se desarrolla en la misma pantalla de la figura anterior.

Por otro lado, es necesario asignar la duración de la tarea y definir la fecha de comienzo programada para el desarrollo del cambio, a fin de que el sistema calcule el Plan de tareas (similar a un PERT) que lo guiará, determinando cuáles serán las tareas críticas y las fechas límite de comienzo y fin de cada tarea. A partir de este plan, los desarrolladores tienen una referencia del tiempo en que pueden y deben trabajar en cada etapa del cambio.

Así, cada tarea irá pasando por distintos estados en la asignación. En un primer momento, todas las tareas de desarrollo que aparecen en el diagrama, que dependerán del tipo de cambio que se esté analizando como se explicó en el módulo de desarrollo, estarán en estado de "sin asignación". Al asignar la duración o el equipo de una tarea, se moverán a los respectivos estados. Cuando todas las tareas de desarrollo cuentan con equipo y duración, la labor estará finalizada.

La asignación de equipo y duración se realiza por medio de los botones que aparecen en la pantalla. La herramienta permite ver en forma gráfica el estado de la asignación distinguiendo por medio de diferentes colores los estados descritos anteriormente.



Así, de un solo vistazo, el o los actores encargados de la asignación pueden ver cuál es el estado de su trabajo y cuándo se puede dar por finalizada la asignación con la seguridad de no equivocarse o que queden tareas sin equipo o duración.

Interacción con otras herramientas

Un requerimiento fundamental de la solución planteada es la necesidad de comunicación entre los participantes del workflow, ya sea por medio de la documentación que debe fluir entre los diferentes estadios como así también la posibilidad de intercambio de mensajes sin necesidad de salir del sistema. Por lo cual, en casi todos los estadios de la solución, existe la necesidad de resolver los siguientes problemas:

- manipulación de documentos y
- correo electrónico.

Manipulación de documentos

El resultado de la mayoría de las etapas en la Gestión del Cambio es uno o dos documentos con cierto formato preestablecido que se pasan a alguna o todas las etapas subsecuentes. Por otro lado, para llevar adelante las tareas del Desarrollo del Cambio también se consultarán, modificarán y crearán documentos que serán consultados por otras personas. Es necesario, entonces, que la herramienta administre el manejo de los documentos necesarios para el cambio.

Es conveniente generar una nueva versión de la documentación ante cada solicitud de Cambio del software, para evidenciar la evolución de la misma a medida que evoluciona el software. La herramienta prevé la administración del versionado de los documentos, presentando la versión que corresponda dependiendo del proyecto, versión y número de cambio seleccionado. Una vez seleccionado un documento y ya en el ambiente del procesador de texto, éste será el documento activo, y sólo se podrá operar sobre él.

Dependiendo de la tarea en que se encuentre el usuario y el estado de la misma el acceso a los documentos puede ser de sólo lectura o lectura - escritura. Para las tareas terminadas, el acceso es de sólo lectura. En cambio, si la tarea está activa o en revisión, el acceso a los documentos puede ser de sólo lectura para los que son de consulta, o de lectura - escritura para los generados en dicha etapa.

Como ya se dijo en la exposición de las secciones anteriores, el procesador de texto utilizado es *Microsoft Word*, el cuál fue adaptado a los requerimientos de la herramienta. La mayoría de los cambio afectaron al menú Archivo (figura 19).

La opción *Nuevo* fue eliminada porque sólo se permite trabajar sobre los documentos generados por la herramienta, es decir, aquellos elegidos por el usuario dentro de la misma y por lo tanto estarán totalmente controlados.

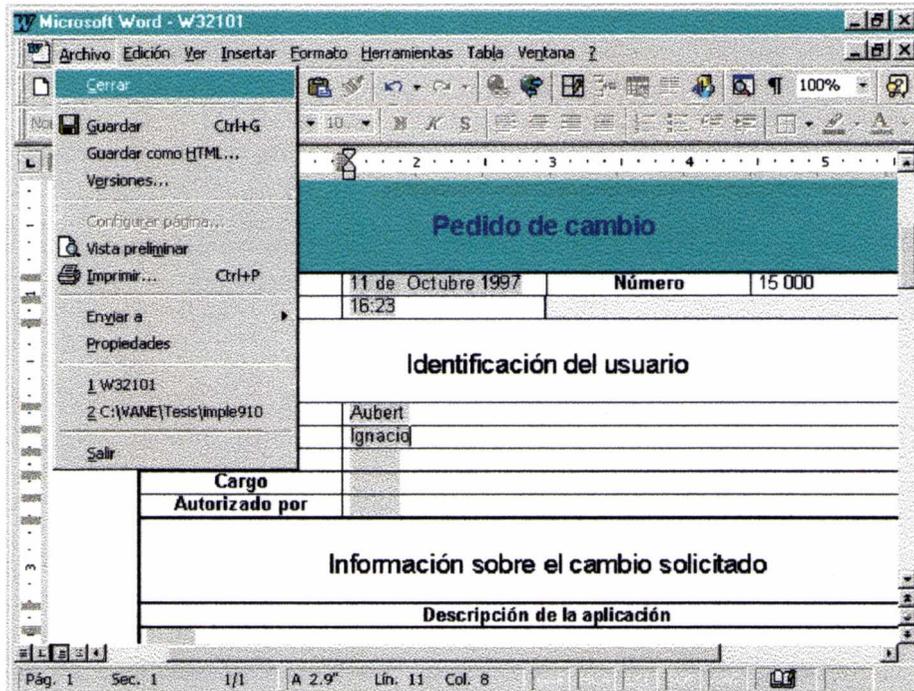


Figura 19

La opción *Guardar*, se comporta de la forma habitual pero fue modificada para que no llame a la opción *Guardar Como* cuando se modifica un documento de sólo lectura. Esto se debe al comportamiento del procesador de texto utilizado, ya que al permitir la modificación de un documento de sólo lectura activa la opción *Guardar Como*, dándole la posibilidad al usuario de grabarlo con el nombre que desee. Pero no sólo la opción *Guardar* desencadena este proceso, ya que al seleccionar *Cerrar* o *Salir*, también se llama a *Guardar Como*, es por ello que debieron ser modificadas para cambiar su comportamiento.

De este modo, los documentos estarán protegidos de toda modificación no autorizada, resguardando así información vital para el sistema de mantenimiento como lo es la contenida en sus documentos.

Correo electrónico

La comunicación entre los participantes es una piedra angular en la herramienta. Tanto para intercambio de información entre ellos como así también para los mensajes de aviso del sistema informando sobre los trabajos pendientes que tiene cada uno.

Para afrontar la problemática anterior, en la herramienta se provee la posibilidad de establecer comunicación entre los participantes por medio del Correo Electrónico. Permitiendo el acceso a éste desde la herramienta misma, sin necesidad de salir del sistema y abandonar la tarea que estaba desarrollando.

A pesar de sus desventajas, como por ejemplo que disminuye la interacción entre las personas, el Correo Electrónico brinda ventajas muy interesantes. En las organizaciones que se manejan con Correo Electrónico se puede observar:

- un aumento de la eficiencia de los empleados.
- una reducción de la necesidad de conversaciones telefónicas.
- desaparición de la burocracia de los papeles que debían ser llevados por personas a otras oficinas, corriendo el riesgo de perderse y los correspondientes retrasos.
- comunicación inmediata con cualquier sitio de la empresa, aunque se encuentre en algún lugar remoto.

El participante que está trabajando con la herramienta puede acceder al correo de dos formas, en el momento de ingresar al sistema se le avisará que tiene mensajes pendientes y si desea atenderlos y luego, en el ambiente de trabajo por medio de la opción de mensajería.

Una vez dentro del correo se permite consultar todos los mensajes recibidos, que aparecen en una lista, marcando con una flecha los mensajes que aún no se leyeron. Además, se cuenta con la posibilidad de actualizar esta lista para tener acceso exactamente a todos los mensajes recibidos hasta el momento. (figura 20)

También en esta pantalla se tiene la opción de imprimir o eliminar un mensaje, desde el botón o por medio de la técnica “drag and drop” explicada anteriormente.

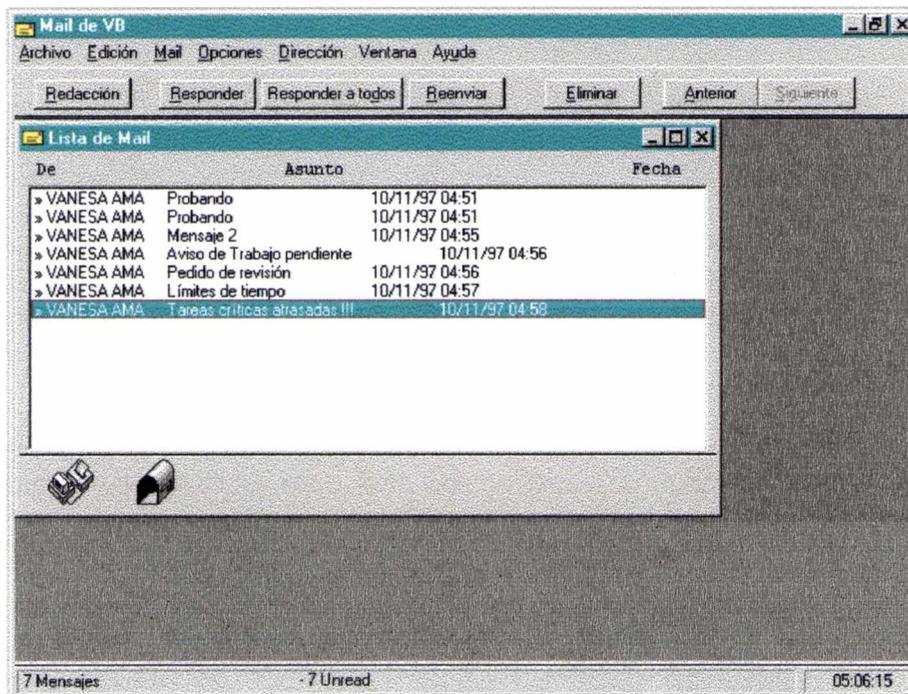


Figura 20

El servicio de Correo que brinda la herramienta nos permite:

1. redactar un mensaje,
2. leer un mensaje recibido,
3. enviar un mensaje
4. almacenar los mensajes enviados
5. responder a los mensajes recibidos,

6. eliminar los mensajes que ya no interesen
7. imprimir los mensajes y
8. enviar copias de cortesía

Para componer un mensaje se debe elegir la opción redactar, en el campo Para poner la dirección del receptor del mensaje y en Cc colocar las direcciones a las cuáles le quiere enviar una copia del mensaje. Luego, en el cuadro resumen colocar la frase que explique el motivo del mensaje y que es la que aparecerá en la pantalla del receptor. Por último, redactar el texto del mensaje y elegir el botón de enviar.

Para contestar el mensaje seleccionado, se debe elegir el botón Responder y se accede nuevamente a la pantalla de composición de mensajes mencionada anteriormente.

Otro servicio que brinda es la libreta de direcciones, una especie de agenda que se mantiene con todas las direcciones accedidas por el usuario. Para buscar un nombre se provee la opción buscar (figura 21). De allí se puede seleccionar el o los receptores del mensaje a componer, es decir los datos que se colocarán en los cuadros Para y Cc.

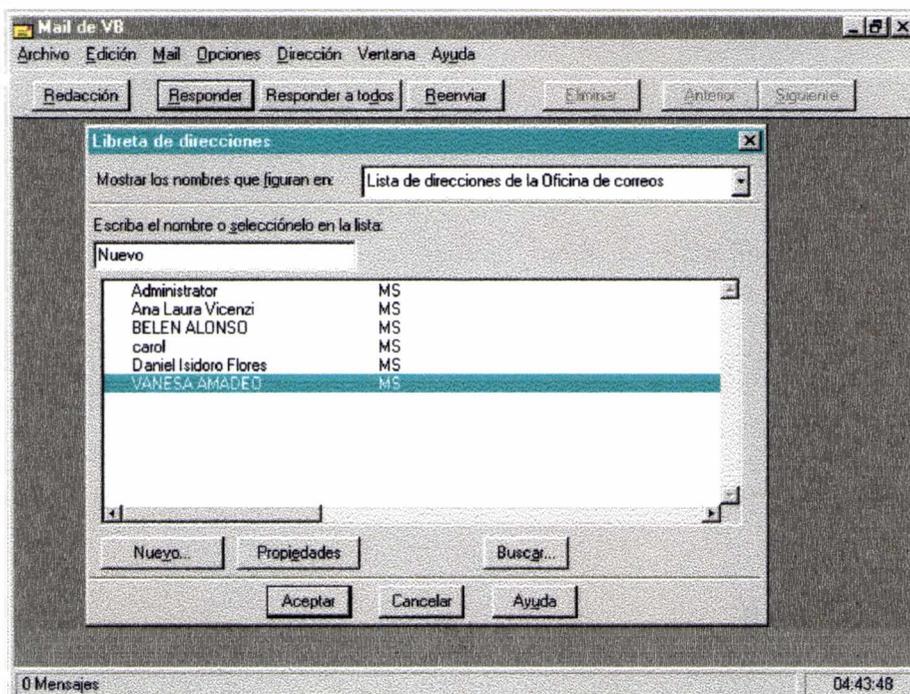


Figura 21

Ingreso al sistema

Para ingresar al sistema es necesario identificarse en una pantalla de login que solicita, en primer lugar <nombre de entrada> y <contraseña> para identificar al usuario, y luego permite elegir el módulo con el que desea trabajar (figura 22). Una vez elegido el módulo, se presenta al usuario la lista de las tareas que está habilitado a desarrollar, entre las que componen los distintos estadios del workflow subyacente, en el caso de los módulos de **Gestión** y **Desarrollo**, o la única tarea posible en el caso de que el módulo elegido sea **Administración**.

Cuando el sistema pudo determinar quién ingresa y con qué rol, se habilita el botón de ingreso al sistema. Permitiendo el ingreso a la tarea elegida por el usuario. Si el usuario tiene mensajes nuevos, la herramienta le informa este hecho y le da la posibilidad de acceder a ellos.

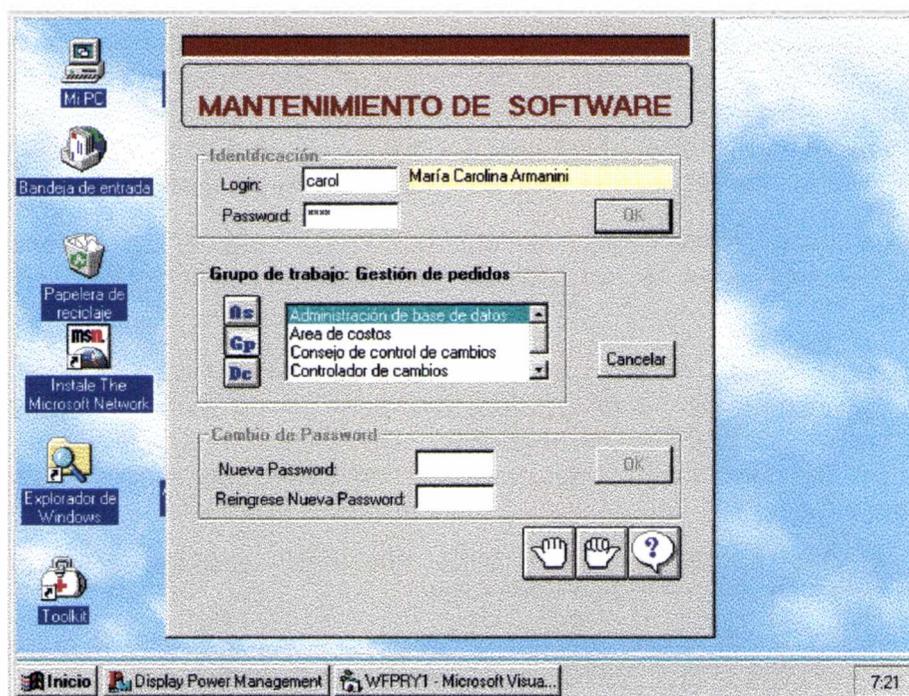


Figura 22

Además, en esta pantalla, el usuario tiene la posibilidad de cambiar su contraseña. Para ello se debe ingresar una contraseña nueva y reingresarla para evitar errores u omisiones. En caso de hacerlo, además debería cambiar su contraseña en el sistema de correo, ya que esta actualización no se realiza automáticamente. Tiene la posibilidad de acceder directamente al correo y cambiarla o informar del cambio al Administrador y solicitarle su cambio de contraseña en el sistema de correo. El hecho de actualizar la contraseña del correo es de suma importancia para mantener la transparencia en el manejo de su correspondencia.

Conclusiones

El sistema tiene por objeto apoyar a una empresa de desarrollo de software en la etapa de mantenimiento del mismo, y provee un número de características que pueden ser de mucha ayuda en las organizaciones facilitando la documentación de programas, el control de versionado, administrando la asignación de recursos, proveyendo diferentes niveles de seguridad y sobre todo, automatizando la comunicación entre los encargados del mantenimiento, a fin de agilizar la tarea, para evitar demoras en la etapa del ciclo de vida del software más cara para las empresas.

Con respecto al primer punto mencionado, la documentación, es vital en una empresa que se encarga del mantenimiento del software, la herramienta provee una organización y versionado de la misma de tal forma que :

- el acceso a ella esté restringido sólo a las personas autorizadas,

- se dispone de la documentación generada anteriormente en un modo organizado.
- siempre se encuentra disponible la versión del documento que se necesita en cada etapa. Sin necesidad de búsquedas extenuantes.

El control de versiones es primordial en este tipo de cuestiones, por medio de la herramienta se obtiene :

- información sobre qué versión del sistema estamos modificando actualmente, como también cuál es la cadena de versiones anteriores,
- información sobre qué versiones del sistema tiene instalado cada cliente y así conocer que versión se debe modificar ante un pedido de cambio
- como ya se ha mencionado, el control del versionado de la documentación del proyecto, conociendo en cada versión del mismo cuáles son los documentos correspondientes.

También se provee un soporte para la asignación de recursos, con el cual se permite :

- determinar los integrantes de cada etapa de un pedido de cambio y las responsabilidades asignadas a cada uno.
- controlar el estado del desarrollo de las etapas del cambio, para conocer los posibles atrasos y poner sobre aviso a sus encargados.
- conocer en todo momento la productividad real de cada uno de los empleados de la organización.

Puede adaptarse a empresas con distintos organigramas, debido a que asume que la etapa de mantenimiento será desarrollada por una jerarquía que responderá a las necesidades propias de la etapa de mantenimiento y tomará recursos de los distintos sectores del organigrama principal. Cada uno de los grupos definidos en este organigrama paralelo tendrá los niveles que la empresa considere necesarios, dependiendo de la cantidad de personal que se destine a mantener el software generado.

En el proceso de Mantenimiento se necesita un cierto nivel de protección de documentos y acciones que, en la mayoría de los casos, deben ser confidenciales. Para ello la herramienta provee diferentes niveles de seguridad:

- por cada proyecto. Sólo los encargados del mismo tendrán acceso al desarrollo de las tareas y la información de la misma de acuerdo al rol que tenga asignado.
- dentro de la organización. Sólo el administrador de la herramienta tendrá acceso a la información global de la organización. El resto de los participantes tendrán el acceso reducido a los proyectos que tiene a su cargo.

La comunicación entre los participantes se establece de dos modos:

- por correo electrónico entre los participantes del sistema,
- por la activación de tareas en el circuito, por medio del aviso de finalización de una tarea a todas las posteriores.

El último punto a destacar, y si se quiere el más importante, es la automatización de la comunicación entre los encargados del mantenimiento, permitiendo que el proceso de mantenimiento, una de las etapas más críticas en la vida del software, se pueda desarrollar en un modo organizado y controlado. Desde que se plantean un conjunto de pasos a seguir, en cada uno el participante conoce cuál será la próxima tarea a realizar en cada proyecto.

Apéndice A

Documentos de la Gestión del Cambio

- I. Pedido de Cambio
- II. Autorización del Cambio
- III. Informe Técnico por Pedido de Cambio
- IV. Informe Técnico para el Area de Costos
- V. Informe de Administración de Base de Datos por Pedido de Cambio
- VI. Informe de Administración de Base de Datos para el Area Costos
- VII. Informe del Area Costos por Pedido de Cambio
- VIII. Informe por Pedido de Cambio Rechazado
- IX. Plan de Trabajo de la Ejecución del Cambio

I. Documento del Pedido de Cambios.

Pedido de cambio			
Fecha		Número	
Hora			
Identificación del usuario			
Apellido			
Nombre			
Sector			
Cargo			
Autorizado			
Información sobre el cambio solicitado			
Descripción de la aplicación			
Descripción del cambio solicitado			
Tipo de cambio			
Error	<input type="checkbox"/>	Cancela	<input type="checkbox"/>
		No Cancela	<input type="checkbox"/>
Mejora	<input type="checkbox"/>		
Adaptación	<input type="checkbox"/>		
Prioridad (Indique un valor de 1 a 4)			
Plazo	(dd-mm-aa)		

II. Documento de Autorización del Pedido de Cambio.

Autorización de Pedidos de cambio			
Fecha		Número de Pedido	
Hora			
Identificación de la autoridad			
Apellido			
Nombre			
Sector			
Cargo			
FIRMA			

III. Documento del Informe Técnico por Pedido de Cambio.

Informe técnico por pedido de cambio			
Fecha		Número de	
Hora			
Identificación del Supervisor			
Apellido			
Nombre			
Información técnica sobre el cambio solicitado			
Elementos de configuración		Afectado(s/n)	
Especificación del sistema			<input type="checkbox"/>
Especificación de requisitos			<input type="checkbox"/>
Especificación de Diseño			<input type="checkbox"/>
Código Fuente			<input type="checkbox"/>
Manuales de Operación e instalación			<input type="checkbox"/>
Programas ejecutables			<input type="checkbox"/>
Descripción de la base de datos			<input type="checkbox"/>
Manual de usuario final			<input type="checkbox"/>
Tipo de Cambio			
Error	<input type="checkbox"/>	Cancela	<input type="checkbox"/>
		No cancela	<input type="checkbox"/>
Mejora	<input type="checkbox"/>		
Adaptación	<input type="checkbox"/>		

IV. Documento del Informe Técnico por Pedido de Cambio para el Area Costos

Informe Técnico por pedido de cambio para Area Costos			
Fecha		Número de Pedido	
Hora			
Identificación del Supervisor			
Apellido			
Nombre			
Evaluación de Recursos			
Descripción			Cantidad de Horas
Analista Senior			
Analista Junior			
Analista Programador Senior			
Analista Programador Junior			
Programador Senior			
Programador Junior			
Operador			
Documentación			

VI. Informe del Administrador de Base de Datos para el Area Costos

Informe Adm. Base de Datos para Area Costos			
Fecha		Número de Pedido	
Hora			
Identificación del Técnico			
Apellido			
Nombre			
Evaluación de Recursos			
Descripción			Cantidad de Horas
A.D.B.M.S. Senior			
A.D.B.M.S. Junior			
Programador Senior			
Programador Junior			
Operador			
Documentación			

VII. Informe del Area Costos por Pedido de Cambio

Costos para el Cambio			
Fecha		Número de Pedido	
Hora			
Identificación del Técnico			
Apellido			
Nombre			
Información técnica sobre el cambio solicitado			
Descripción	Cantidad de Horas	Valor Hora	Total
Analista Senior			
Analista Junior			
Analista Programador Senior			
Analista Programador Junior			
Programador Senior			
Programador Junior			
Operador			
A.D.B.M.S Senior			
A.D.B.M.S Junior			
Programador Senior			
Programador Junior			
Operador			
Documentación			
UTILIZACIÓN DE RECURSOS DE HARDWARE Y SOFTWARE			
GASTOS GENERALES			
GASTOS IMPOSITIVOS			
TOTAL			

IX. Plan de Trabajo para la Ejecución del Cambio

1. Documento Preliminar
2. Plan de Ejecución de Tareas
3. Equipo de Trabajo

1. Documento Preliminar

Plan de Trabajo															
Tareas	Responsable	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Especificación del Cambio															
Especificación de requisitos del cambio															
Diseño Preliminar de Datos															
Diseño Preliminar de Estructura															
Diseño Detallado de Datos															
Diseño Detallado de Estructura															
Codificación															
Prueba del Producto															
Generación de Manuales del Usuario															
Generación de Manuales de Ope/Ins.															
Instalación															
Supervisión del Desarrollo															

2. Plan de Ejecución de Tareas

<h2 style="margin: 0;">Plan de Trabajo</h2> <h3 style="margin: 0;">Tareas</h3>							
Identificación del Pedido de Cambio Proyecto: Versión que se modifica: Nro. de Pedido de Cambio:	Versión que se genera					Fecha:	Página:
Tareas	Duración	Crítica	Inicio Tempr.	Inicio Tardía	Fin Tempr.	Fin Tardía	Estado
1 Especificación del Cambio							
2 Especificación de requisitos del cambio							
3 Diseño Preliminar de Datos							
4 Diseño Preliminar de Estructura							
5 Diseño Detallado de Datos							
6 Diseño Detallado de Estructura							
7 Codificación							
8 Prueba del Producto							
9 Generación de Manuales del Usuario							
10 Generación de Manuales de Ope/Ins.							
11 Instalación							
12 Supervisión del Desarrollo							

Apéndice B

Documentos del Desarrollo del Cambio

- I. Documento de Especificación del Sistema
- II. Documento de Especificación de Requerimientos del Sistema
- III. Documento del Diseño Preliminar de Datos
- IV. Documento del Diseño Preliminar de Estructura
- V. Documento del Diseño Detallado de Datos
- VI. Documento del Diseño Detallado de Estructura
- VII. Documento de la Codificación
- VIII. Documento de la Prueba del Soft
- IX. Manual del Usuario
- X. Manuales de operación e Instalación
- XI. Documentos de la Instalación del Sistema

DOCUMENTO DE ESPECIFICACIÓN DEL SISTEMA

1-IDENTIFICACIÓN DEL SISTEMA

a) Nombre:

[Nombre del sistema]

b) Descripción breve:

[Reseña del sistema]

c) Autor:

[Nombre del autor del documento]

2-DEFINICIÓN DEL PROBLEMA

[Situación actual de la empresa en torno al problema]

3-JUSTIFICACIÓN DEL SISTEMA

[Alternativas de solución]

4-OBJETIVOS DEL SISTEMA

[Funcionalidad a cumplir]

5-RESTRICCIONES DE SISTEMA

[Precisiones de performance. Descripción del hard]

6-FUNCIONES

[Funciones de soft, de hard y del personal]

7-CARACTERÍSTICAS DEL USUARIO

[Organigrama de la empresa. Sectores usuarios del sistema]

8-AMBIENTES

[Ambientes de desarrollo, de operación y de mantenimiento]

9-ESTRATEGIAS DE SOLUCIÓN

[Conjunto de soluciones posibles. Justificación de la elección]

10-CARACTERÍSTICAS PRIORITARIAS

[Requisitos principales a cumplir]

11-CRITERIOS DE ACEPTACIÓN

[Elementos que evaluará el usuario]

12-TÉRMINOS ESPECÍFICOS

[Aclaración de terminología técnica usada]

ESPECIFICACIÓN DE REQUERIMIENTOS DEL SISTEMA

1-IDENTIFICACIÓN DEL SISTEMA

a) Nombre:

[Nombre del sistema]

b) Descripción breve:

[Reseña del sistema]

c) Autor de este documento:

[Nombre del autor del documento]

2-DESCRIPCIÓN DE LA INFORMACIÓN

a) Representación del flujo de la información

[Flujo de datos]

[Flujo de control]

b) Descripción del contenido de la información

[Descripción detallada del problema a resolver]

c) Descripción de la interfaz del sistema

[Detalle de las interfaces entre los módulos y con elementos externos]

3-DESCRIPCIÓN FUNCIONAL

[Descripción de cada función]

[Narrativa de procesamiento]

[Restricciones]

[Diagramas de soporte]

4-DESCRIPCIÓN DEL COMPORTAMIENTO

[Funcionamiento del soft como consecuencia de eventos externos]

[Estados del sistema]

[Sucesos y acciones]

5-CRITERIOS DE VALIDACIÓN

[Límites de rendimientos]

[Clases de pruebas exigidas para validar funciones, rendimiento y restricciones]

[Respuesta esperada del sistema]

DISEÑO PRELIMINAR DE DATOS

1-IDENTIFICACIÓN DEL SISTEMA

a) Nombre:

[Nombre del sistema]

b) Descripción breve:

[Reseña del sistema]

c) Autor de este documento:

[Nombre del autor del documento]

2-DESCRIPCIÓN DE LOS DATOS

[Revisión del flujo de datos]

3-MÓDULOS

[Por cada módulo: texto explicativo]

[Por cada módulo: organización de los datos]

4-DATOS GLOBALES

[Datos externos a las bases de datos]

5-REFERENCIAS CRUZADAS PARA LOS REQUISITOS

[Asegurar que los requisitos se satisfacen con el diseño preliminar de datos]

[Indicar elementos críticos]

DISEÑO PRELIMINAR DE ESTRUCTURA

1-IDENTIFICACIÓN DEL SISTEMA

a) Nombre:

[Nombre del sistema]

b) Descripción breve:

[Reseña del sistema]

c) Autor de este documento:

[Nombre del autor del documento]

2-ÁMBITO

[Objetivos del sistema]

[Principales funciones del software]

[Principales restricciones y limitaciones del diseño]

3-DESCRIPCIÓN DEL DISEÑO

[Estructura de programa]

[Interfaces dentro de la estructura]

4-MÓDULOS (para cada módulo)

[Texto explicativo]

[Descripción de la interfaz]

[Módulos que usa]

[Comentarios]

DISEÑO DETALLADO DE DATOS

1-IDENTIFICACIÓN DEL SISTEMA

a) Nombre:

[Nombre del sistema]

b) Descripción breve:

[Reseña del sistema]

c) Autor de este documento:

[Nombre del autor del documento]

2-DESCRIPCIÓN DE LOS DATOS

[Revisión del flujo de datos]

3-ESTRUCTURA DE ARCHIVOS Y DATOS GLOBALES

[Estructura lógica de archivos externos]

[Descripción lógica de registros]

[Método de acceso]

[Estructura de datos globales]

4-REFERENCIAS CRUZADAS PARA LOS REQUISITOS

[Asegurar que los requisitos se satisfacen con el diseño preliminar de datos]

[Indicar elementos críticos]

5-PREVISIONES DE PRUEBA

[Directrices de prueba]

[Estrategia de integración]

[Consideraciones especiales]

DISEÑO DETALLADO DE ESTRUCTURA

1-IDENTIFICACIÓN DEL SISTEMA

a) Nombre:

[Nombre del sistema]

b) Descripción breve:

[Reseña del sistema]

c) Autor de este documento:

[Nombre del autor del documento]

2-ÁMBITO

[Objetivos del sistema]

[Principales funciones del software]

[Principales restricciones y limitaciones del diseño]

3-DESCRIPCIÓN DEL DISEÑO

[Estructura de programa]

[Interfaces dentro de la estructura]

4-MÓDULOS (para cada módulo)

[Descripción en lenguaje de diseño]

[Comentarios]

5-PROVISIONES DE PRUEBA

[Directrices de prueba]

[Estrategia de integración]

[Consideraciones especiales]

DOCUMENTO ASOCIADO A LA CODIFICACIÓN

1-IDENTIFICACIÓN DEL SISTEMA

a) Nombre:

[Nombre del sistema]

b) Descripción breve:

[Reseña del sistema]

c) Autor de este documento:

[Nombre del autor del documento]

2-LENGUAJES DISPONIBLES

[Enumeración de los lenguajes disponibles para la codificación y características que los hacen elegibles]

3- LENGUAJES UTILIZADOS

[Justificación de la elección de los lenguajes]

[Justificaciones para descartar lenguajes]

4-MÓDULOS COMPONENTES

[Enumeración de los módulos de código que conforman el soft y su localización]

[Comentarios]

DOCUMENTO DE PRUEBA DEL SOFT

1-IDENTIFICACIÓN DEL SISTEMA

a) Nombre:

[Nombre del sistema]

b) Descripción breve:

[Reseña del sistema]

c) Autor de este documento:

[Nombre del autor del documento]

2-ALCANCE DE LA PRUEBA

[Características funcionales, de rendimiento y de diseño a ser probadas]

[Límites del esfuerzo de prueba]

3-PLAN DE PRUEBA

a) Grupo de prueba

[Identificación de los encargados de llevar a cabo las pruebas]

[Personal de la empresa cliente que debe estar presente]

b) Ámbito de prueba

[Determinación del lugar físico donde se desarrollará la prueba]

c) Requerimientos

[Agenda]

[Recursos de hard y de soft necesarios]

d) Fases de prueba

[Enumeración de las fases en que se divide la prueba de acuerdo a la funcionalidad que se evalúa en cada una]

4-PROCEDIMIENTO DE PRUEBA (para cada fase)

a) Orden de integración

[Propósito]

[Módulos a ser probados]

b) Pruebas de unidad, para cada módulo de la fase

[Técnicas utilizadas: caja blanca, caja negra, camino básico, etc]

[Resultados esperados]

c) Datos de los casos de prueba

[Datos de entrada para probar el soft]

6-RESULTADO DE LAS PRUEBAS

[Resultados obtenidos al someter al soft a las pruebas planificadas con los datos planificados]

[Conclusiones a partir de los resultados]

MANUAL DEL USUARIO

1-IDENTIFICACIÓN DEL SISTEMA

a) Nombre:

[Nombre del sistema]

b) Descripción breve:

[Reseña del sistema]

c) Autor de este documento:

[Nombre del autor del documento]

2-OBJETIVOS DEL SISTEMA

[Descripción del sistema]

[Área de incumbencia]

3-MÓDULOS COMPONENTES

[Enumeración de los principales módulos del sistema]

4-DESCRIPCIÓN DE COMPONENTES (para cada módulo)

[Los siguientes puntos se desarrollarán a distintos niveles de abstracción, para cada módulo o submódulo en que se divida el sistema]

a) Funcionalidad

[Principales funciones que realiza el módulo]

b) Menú / Descripción de pantallas principales

[Orientación gráfica de lo que se verá del sistema en operación]

c) Explicación de cada opción de menú / Función

[Instrucciones para el uso]

5-SOLUCIONES A PROBLEMAS

[Guía rápida para solucionar problemas en la operatoria habitual]

6- MENSAJES DE ERROR

[Apéndice de códigos de error que retorna el sistema]

MANUAL DE OPERACIÓN E INSTALACIÓN

1-IDENTIFICACIÓN DEL SISTEMA

a) Nombre:

[Nombre del sistema]

b) Descripción breve:

[Reseña del sistema]

c) Autor de este documento:

[Nombre del autor del documento]

2-VERSIONES DEL SISTEMA

[Enumeración de las versiones y breve descripción de cada una]

[Requerimientos de hard y de soft previos a la instalación de cada versión]

3-PROCEDIMIENTO DE INSTALACIÓN

[Por cada versión detallar los pasos para lograr su instalación correcta]

4-PROCEDIMIENTO DE OPERACIÓN

[Descripción de procesos complementarios, por ejemplo procesos batch, horario de corrida, procesos previos, etc]

[Comentarios]

DOCUMENTOS DE LA INSTALACIÓN DEL SISTEMA

- **DESCRIPCIÓN DE LA INSTALACIÓN**
- **CRONOGRAMA DE INSTALACIÓN**

DESCRIPCIÓN DE LA INSTALACIÓN

1-IDENTIFICACIÓN DEL SISTEMA

a) Nombre:

[Nombre del sistema]

b) Descripción breve:

[Reseña del sistema]

c) Autor de este documento:

[Nombre del autor del documento]

2-ÁMBITO DE INSTALACIÓN

[Nombre de la empresa cliente]

[Dirección]

[Teléfono]

3-RESPONSABLES DE INSTALACIÓN EN LA EMPRESA CLIENTE

[Nombre del encargado]

[Teléfono]

[Horario de trabajo]

4-HORARIOS PARA LA INSTALACIÓN

[Enumeración de restricciones horarias impuestas por el cliente, si las hay]

[Comentarios]

5-ELEMENTOS ADICIONALES

[Enumeración de elementos de hard, de soft, herramientas, etc. necesarios para realizar la instalación]

6-OBSERVACIONES

[Notas para recordar]

CRONOGRAMA DE INSTALACIÓN

Identificación del Pedido de Cambio Proyecto: Versión que se modifica: Nro. de Pedido de Cambio:	Lugar de Instalación Cliente:
--	---

Fecha: Página:

Sector	Versión Actual	Versión Nueva	Fecha de Instalación	Fecha Instalado	Estado de Instalación

- Bibliografía -

Administración de Problemas y Cambios en sistemas complejos (experiencia en Aerolíneas Argentinas)

Lucas, G.

Anales Congreso y Exposición Internacional de Informática y Tel telecomunicaciones. INFOCOM 95. La Nueva Percepción

Administrator's Guide. Microsoft Mail

Microsoft Corporation. 1993

Análisis Estructurado Moderno

Yourdon, E.

Prentice Hall 1993.

Case Study: The role of IS in Reengineering

Eckerson, W.

Open Information System, Patricia Seybold Group, Vol9, No. 2. Febrero 1994.

Conozca las Herramientas de Flujo de Trabajo.

May, T.

BYTE, Julio 94, Mc. Graw Hill.

Customizing Transaction Models and Mechanisms in a Programmable Environment Supporting Reliable Workflow Automation.

Georgakopoulos, D.

IEEE Transactions on Knowledge and Data Engineering. Abril 1996.

Design Principles for Cooperative Office Support System in Distributed Process Management

Schael, T, and Zeller, B.

Support Functionality in the Office Environment . Editorial Verrinjn – Stuart 1991

Fitting Pieces to the Maintenance Puzzle

Osborne, W. And Chikofsky.

IEEE Software, enero 1990. Pp10-11.

Fundamental Concepts for Workflow Automation in Practice. Centre for Telematics and Information Technology,

Joosten, S. and Brinkkemper, S.
University of Twente, The Netherlands 1995.

Fundamentals of Software Engineering.

Ghezzi, C.; Jazayeri, M.; Mandrioli, D.
Prentice Hall 1991.

Groupware: Some Issues and Experiences

Ellis, C.; Gibbs, J. and Rein, G.
Communications of the ACM, Vol 34, Nro.1. 1991

Handbook of workthroughs, inspections, and technical reviews

Freedman, D.P. and Weinberg
Dorset House, 1990.

Herramienta para la Administración de problemas y cambios

Bertone, R.; Ramón, H.; Russo, C.; Gorga, G.; Madoz, C.
Encuentro de Ingeniería de Software Santiago de Estero 1996.

Ingeniería de Software. Un enfoque práctico

Pressman, R.
Mc. Graw Hill 1993.

Maintenance Management

Bennet, L. and Swanson, B.
Reading, Mass.: Addison Wesley, 1980.

Managing Software Development and Maintenance

Mc. Clure.
Van Nostrand Reinhold, 1981.

Managing the System Life Cycle

Yourdon, E.
Yourdon Press 1988.

Programmer's Guide. Microsoft Visual Basic

Microsoft Corporation. 1993

Realizing the Workflow Paradigm in Products, Methodologies, Infraestructure, and Current Research.

Georgakopoulos, D.

Workflow Tutorial. GTE Laboratories Technical Communication. Dic.1994.

Software Maintenance Management

Bennet, L. and Swanson, B.

Reading, Mass.: Addison Wesley, 1980.

The Maintenance "Iceberg".

Canning , R.

EDP Analyzer, vol 10, no. 10, Octubre 1972.

The state of software maintenance"

Schemeidewind, H.

IEEE Trans. Software Engeeniering, vol. Sec. 13, nro3, mar. 1987, pág 300-310

The Use of Software Complexity Metrics in Software Maintenance .

Kafura, D. and Reddy, G.

IEEE Trans. on Software Engineering, vol SE-13, no. 3, mar. 1987. Pp 335- 343

The Visual Basic 3 for Windows Handbook.

Gary Cornell.

Osborne- Mc. Graw Hill. 1993

There is more than one kind of Workflow Software.

Mc Cready, S.

Computerworld. Noviembre 1992

Ultimus, Bussines Workflow Automation,

[http: //www.ultimus1.com/utintro/ultintro.](http://www.ultimus1.com/utintro/ultintro)

Una propuesta de Herramienta para la Administración y Gerenciamiento del Mantenimiento de Proyectos de Software

Amadeo. M.; Alonso, B.; Bertone,R.

Anales de CACIC 97. Facultad de Ciencias Exactas. Dpto. Informática UNLP Septiembre 1996.



Una propuesta para la especificación de documentos para el proceso de Ingeniería de requerimientos.

Balda, M.; Vicenzi, A., Bertone, R.

Anales de ICIE 97. Facultad de Ingeniería.UBA Buenos Aires 1997.

Understanding Maintenance Work

Bendifallah, S. and Scacchi, W.

IEEE Transactions on Software Engineering, vol SE-13, no. 3, marzo 1987.

WorkFlow

Fernández, Jorge Ben,

LAN & WAN, Abril 95

Workflow. Apuntes de Master de Automatización de Oficinas.

Bianchi, A.

Dpto. de Informática. Facultad de Ciencias Exactas UNLP 1997.

Workflow Management.

World Wide Web server of the Workflow Management Project at the departament of Computer Science of the University of Twente in the Neteherlands.

<http://www-is.cs.utwente.nl:8080/~joosten/workflow.html>.

Workflow para la Etapa de Mantenimiento del Ciclo de Vida Clásico.

Ramón, H.; Bertone, R.; Russo, C.; Gorga, G.; Madoz, C.

Anales del IV Encuentro Chileno de Computación. Noviembre 1996.

WorkFlow Reengineering

Poysic, G. and Hannaford, S.

Adobe Press. 1996

Workflow Software: A Layman' s Handbook, Part1.

Black, D.

INFORM Abril 1994.

Workflow Software Automates Processes

Korzeniowski, P.

Software Magazine. Febrero 1993.

DONACION.....	TES
Fecha... 29-8-05	97/2
Inv. E.....Inv. B... 1962	

TES
97/2
DIF-01962
SALA



UNIVERSIDAD NACIONAL DE LA PLATA
Biblioteca
50 Y 120 La Plata
catalogo:info.unlp.edu.ar
biblioteca@info.unlp.edu.ar



DIF-01962