

# 24 MANTENIMIENTO DE LA ESPECIFICACION

Hasta ahora, el profesional clave de la computadora era alguien que podía aprender lo suficiente acerca de las necesidades de las organizaciones como para expresarlas en lenguaje de computadora. En el futuro, a medida que nuestra sociedad se vuelva irrevocablemente computarizada, el profesional clave será alguien que pueda aprender lo suficiente acerca de sistemas computarizados como para expresarlos en lenguaje humano. Sin ese alguien, habremos perdido el control de nuestra sociedad. Ese alguien es el ingeniero en reversa. Los que mantienen el software son los ingenieros en reversa de nuestra sociedad.

Nicholas Zvegintzov, editor  
*Software Maintenance News*

## En este capítulo se aprenderá:

1. Por qué es importante tener al día las especificaciones.
2. Qué tipo de cambios se necesitan hacer a una especificación.

Para muchos analistas, el proyecto termina cuando se termina la especificación estructurada y el usuario la acepta. En ese momento se entrega la especificación al equipo de implantación constituido por los diseñadores y programadores que construirán un sistema a partir de la especificación.

Desde luego, algunos analistas siguen colaborando con el proyecto a lo largo de las fases de diseño e implantación. A veces el analista sirve de administrador del proyecto, guiando y dirigiendo los esfuerzos del equipo de implantación. A veces sigue colaborando con la realización de análisis, es decir, sirviendo como intermediario entre el usuario y el equipo de implantación. También puede participar en el desarrollo de manuales para el usuario, datos de prueba de aceptación, planeación de la instalación y varias actividades complementarias que se hacen de manera concurrente con el proceso de implantación.

Sin embargo, casi todos los analistas dejan el proyecto en cuanto se completa el desarrollo y se pone en operación el nuevo sistema. Algunos programadores se quedan para actividades de mantenimiento, pero cuando se termina la fase de desarrollo se termina la fiesta, y la mayoría de los analistas, diseñadores y programadores se transfieren a otros proyectos nuevos (y a menudo a compañías nuevas, donde pueden percibir un salario mayor al actual).

Pero el trabajo hecho por el analista (todo el trabajo que se discutió a lo largo de este libro) sigue siendo importante. Así como los *programas* deben mantenerse durante los 5, 10 o 20 años de vida operacional del sistema, de igual manera debe mantenerse su especificación. O, por decirlo de otra manera, cambiarán diversos aspectos de la implantación del sistema durante su vida, y para cada uno de estos cambios debe haber uno correspondiente en la especificación.

Aunque el analista original pudiera no permanecer con el proyecto durante la vida operacional de éste, es importante que deje un legado que se *pueda* mantener. Este capítulo discute el mantenimiento de la especificación del sistema.

### 24.1 POR QUE ES IMPORTANTE

Hasta aquí podría estar un tanto confundido; después de todo, piensa, es perfectamente obvio que la especificación del sistema puede actualizarse. ¿Por qué no hacerlo? Desafortunadamente, la historia del campo de desarrollo de sistemas sugiere algo distinto: la gran mayoría, probablemente más del 80 por ciento, de los sistemas que están en operación actualmente no tienen una declaración precisa y actualizada de los requerimientos de usuario que realizan.

Este no es un fenómeno exclusivo del campo de la computación. ¿Cuántas casas de cien años de antigüedad tienen documentos actualizados que describen la instalación eléctrica, la tubería, la calefacción u otros detalles arquitectónicos? La verdad es que a menudo resulta más fácil hacerle una corrección, mejoría o cambio "rápido y sucio" a un sistema existente, que empezar a cambiar el documento de los requerimientos y luego propagar dicho cambio al documento de diseño y la implantación misma. Esto sucede sobre todo si se necesita hacer el cambio para arreglar un

problema inmediato, presionante y urgente.<sup>1</sup> "Ya cambiaremos los documentos más tarde", dice el encargado de mantenimiento, "pero primero tenemos que arreglar el problema mismo". La documentación es lo último que se quiere hacer, y muchas veces no se hace.

Los sistemas de información tienen una característica importante en cuanto al mantenimiento: duran más quienes los desarrollan originalmente. Esto también se da para las casas; ni el arquitecto ni el usuario final de una casa victoriana construida en 1880 están disponibles para consultarles hoy en día algo. También sucede para muchos sistemas de información; después de 10 o 20 años, el sistema está siendo empleado por usuarios de tercera generación (de los cuales muchos no tienen idea del por qué se desarrolló para empezar) y está siendo mantenido por programadores de mantenimiento de tercera generación (de los cuales algunos ni idea tienen de por qué quienes originalmente lo desarrollaron adoptaron esa estrategia de diseño en particular).<sup>2</sup> Esta es la razón por la cual Nicholas Zvegintzov describe a los programadores de mantenimiento como "ingenieros en reversa de la sociedad".

Hay otra cosa importante sobre los sistemas de información: tienden a ser complejos desde el principio, y se vuelven cada vez más complejos al pasar años de mantenimiento. Si el sistema fuera sencillo (por ejemplo, unas 250 instrucciones de Pascal), entonces se mantendría fácilmente aún si no tuviera documentación. Pero un sistema típico tiene por lo menos 100,000 instrucciones; muchos de los más grandes que se mantienen en la actualidad tienen más de 500,000, y algunos tienen más de un millón de instrucciones. Ningún individuo puede entender la complejidad de un sistema tal, *sobre todo* si 1) no estuvo involucrado en el desarrollo del sistema original y, 2) no se documentaron los requerimientos y el diseño originales. Y sin embargo eso es precisamente lo que pedimos de la mayoría de los programadores de mantenimiento.<sup>3</sup>

Existen docenas, si es que no cientos, de ejemplos de organizaciones con problemas severos de mantenimiento del tipo descrito anteriormente. Casi cualquier organización importante que empezó a computarizarse hace 20 años ahora se enfrenta a sistemas de 20 años de antigüedad cuya implantación es un misterio y, peor aún, cuyos requerimientos de usuario son un misterio.

1 Una encuesta en [Lientz y Swanson, 1980] mostró que aproximadamente un 9% de todo trabajo de mantenimiento consiste en "reparaciones de emergencia de programa".

2 Un estudio hecho por el fabricante británico de computadoras ICL en los años 70 reveló que a un sistema típico lo mantienen *siete generaciones* de programadores de mantenimiento antes de ser desechado finalmente. Esto sugiere que mucho de lo que llamamos programación de mantenimiento podría describirse más precisamente como arqueología.

3 Uno de los ejemplos más extremos de un sistema grande y complejo con requerimientos continuos de mantenimiento es el proyecto de Estación Espacial que actualmente desarrolla la NASA. Su propósito es colonizar e industrializar la porción cercana del sistema solar. Está programado para dentro de 30 años, y requerirá mantenimiento permanente.

La única solución a esta crisis en el futuro es mantener documentación precisa y actualizada por la duración del sistema mismo. ¿Pero, cómo hacerlo?

## 24.2 PRÉRREQUISITOS NECESARIOS

No se puede mantener actualizados un sistema y su documentación asociada a menos que ésta sea precisa. Este es un punto de partida: debe asegurarse que cuando un nuevo sistema se ponga en operación todos los documentos relacionados estén completos y sean consistentes, actualizados y precisos.

A lo largo de este libro hemos discutido las características de un modelo preciso de los requerimientos del usuario, además de las reglas a seguir para asegurar que el modelo del sistema sea completo e internamente consistente. Para que pueda mantenerse con éxito, deben obligatoriamente seguirse estas reglas, y la persona independiente o grupo que lo haga debe certificar que los documentos sean precisos antes de poner el sistema en operación.

Además de certificar que los documentos mismos sean precisos, debe asegurarse que exista un mecanismo para hacerles cambios posteriores. De nada servirá que la especificación estructurada se haya inscrito en tablas de piedra como registro permanente para generaciones futuras; la especificación debe verse como un documento vivo, sujeto a cambios continuos, aunque controlados.

## 24.3 COMO HACERLO

La primera y más fundamental de las reglas para el mantenimiento de sistemas es la siguiente: *cualquier* cambio propuesto al sistema operacional existente debe, en todos los casos, empezar con un examen de su impacto sobre las especificaciones o requerimientos del sistema. Esto debe hacerse en todos los casos que se mencionan a continuación, y con cualquier otro cambio propuesto al sistema:

- El usuario decide que quisiera añadir una nueva función al sistema actual.
- El usuario no está contento con la forma en la que se realiza alguna función actual y quiere cambiarla.
- El usuario quiere un nuevo reporte de salida además de los que ya tiene.
- El usuario quiere modificar el formato u organización de un reporte de salida existente.
- Los programadores de mantenimiento desean recodificar un módulo para hacerlo más eficiente.
- El departamento de operaciones ha anunciado que planea mejorar los sistemas de cómputo actuales de la organización y se necesitarán algunos cambios de programación.

- El usuario se queja de que el sistema produce salidas incorrectas para ciertas combinaciones de entradas.
- La organización de desarrollo de sistemas ha decidido que Ada se adopte como nuevo lenguaje de programación. Se hacen planes para convertir todo el software existente a Ada.
- Se requiere que el sistema mande salidas a una nueva dependencia gubernamental, que no existía cuando se desarrolló originalmente.

Cualquier cambio como éstos debe ilustrarse, documentarse y ser verificado con el usuario, haciendo al modelo del sistema los cambios pertinentes. Esto usualmente se hace llenando una forma conocida como solicitud de cambio del sistema. El cambio de mantenimiento puede involucrar alguno, o todos, los siguientes detalles:

- Añadir terminadores nuevos al diagrama de contexto, o eliminar anteriores. Los flujos de datos entre el sistema y sus terminadores podrían añadirse, eliminarse o cambiarse. Las funciones que previamente desempeñaban los terminadores podrían efectuarse ahora dentro del sistema; de manera inversa, ciertas funciones que el sistema hacía podrían considerarse ahora fuera de él y dentro de los dominios de un terminador.
- Puede ser necesario añadir nuevos eventos a la lista, o eliminar otros.
- Si el cambio es substancial, puede modificarse la declaración de propósitos en el modelo ambiental.
- Los modelos de flujo de datos, modelos de entidad-relación o modelos de transición de estados pueden requerir cambios.
- Las especificaciones de proceso y el diccionario de datos pueden necesitar modificarse o refinarse.
- Varios aspectos del modelo de implantación del usuario pueden requerir cambios que involucren la interfase humano-máquina o las restricciones de implantación que se refieren al tiempo de respuesta, etc.

Ningún cambio de éstos vendrá gratis. Es posible que algunos sean mínimos y sólo requieran unos cuantos minutos de trabajo para ser incorporados, es decir, sólo tomaría minutos hacer los cambios necesarios a la especificación y a los programas existentes. Sin embargo, la persona o grupo que realiza los cambios tiene la obligación de escribir una *declaración de impacto*: esto es, una declaración precisa y detallada de los cambios necesarios en la especificación del sistema para poder implantar el cambio propuesto. Además, debe existir una declaración de impacto económico: es decir, una declaración del costo del cambio y el beneficio que se estima que traerá. Es sobre todo importante si la actividad de mantenimiento cambiará el enfoque del sistema.

Desde luego, habrá algunos cambios que no causen impacto en la especificación del sistema: una corrección de programación para arreglar un error, un cambio de codificación para mejorar la legibilidad o la eficiencia del sistema existente, o un cambio del hardware o software existentes (compilador, sistema operativo, sistema de administración de bases de datos, etc.). Sin embargo, incluso en estos casos debe generarse una declaración de impacto económico para que el usuario y la organización de desarrollo de sistemas entiendan los costos y beneficios asociados con dicho cambio.

Cualquier cambio del sistema comúnmente resultará en un cambio del software y/o hardware; también puede resultar en el cambio de los manuales del usuario, procedimientos de operación y varios otros componentes del sistema. *Pero el documento más importante de actualizar es definitivamente la declaración de requerimientos del usuario.* Sin él, los cambios o modificaciones futuros se volverán cada vez más difíciles de hacer; y el cambio a un sistema totalmente nuevo será infinitamente más caro, tardado y doloroso de lo que debería.

No hay duda de que un analista veterano con 20 años de experiencia vería esta petición de especificación de sistema actualizado con ojos enfermos. Después de todo, el proceso de análisis y la tarea de crear una especificación precisa han sido tan difíciles durante tantos años, que la idea de mantenerla permanentemente actualizada parece casi risible.

La respuesta es, a la larga, la automatización. Las estaciones de trabajo automatizadas de análisis de sistemas del tipo descrito en el Apéndice A están disponibles a costos accesibles, y representan una dramática mejoría sobre la tecnología usada por la mayoría de los analistas hoy, como los sistemas de procesamiento de palabras representan una dramática mejoría sobre la máquina de escribir eléctrica de los años 60. Hay planes más ambiciosos para desarrollar ambientes de ingeniería de software integrados que abarquen todo y que sirvan de depósito central para todos los documentos asociados con el desarrollo de un sistema. Sin embargo, tal tecnología avanzada probablemente no se desarrolle por completo hasta mediados de los años 90.

Sin embargo, queda mucho que hacer aun con la tecnología disponible actualmente. Simplemente no hay excusa para hacer cambios a un sistema existente sin hacer el cambio correspondiente a su especificación. Sin embargo, para que esto funcione se requiere una administración fuerte y disciplinada dentro de la organización.

#### 24.4 RESUMEN

Existe una cantidad creciente de libros sobre el tema del mantenimiento de software, además de por lo menos una sociedad profesional (la Asociación de Mantenimiento de Software en los Estados Unidos) que se ocupa de cuestiones de mantenimiento. El énfasis actual es sobre la administración y refinamiento de programas