

## Guía 8 – Mantenimiento de Software

### Contenido

1. Defina mantenimiento correctivo. ....	2
2) Defina mantenimiento adaptativo .....	2
3) Defina los dos tipos de mantenimiento perfectivo .....	2
4) Enuncie puntos de vista para comprender las características del mantenimiento de software .....	2
5) Esquematice flujo de sucesos que se pueden dar como resultado de una petición de mantenimiento de software .....	2
6) Defina mantenimiento estructurado .....	3
7. Ejemplifique costes intangibles del mantenimiento del software .....	3
8) Plantee un modelo de esfuerzo del mantenimiento de software .....	3
9) Enuncie problemas asociados con el mantenimiento de software .....	3
10) Defina facilidad de mantenimiento de software .....	3
11) Enuncie factores de control relacionados con el entorno de desarrollo .....	4
12) Enuncie métricas de la facilidad de mantenimiento .....	4
13) De un esquema y describa el funcionamiento de una organización de mantenimiento .....	4
14) Defina formulario de petición de mantenimiento (informe de problemas de software) .....	4
15. Defina que informe de cambios de software y que se informa en él. ....	5
16) De un esquema del flujo de sucesos del mantenimiento .....	5
17) Enuncie preguntas de repaso de la situación pertinente una vez concluida la tarea de mantenimiento .....	5
18) Enuncie información a registrar concluida la tarea de mantenimiento .....	6
19) Enuncie medidas del rendimiento del mantenimiento .....	6
20) Enuncie cambios de código que tiene mayor probabilidad de inducir a un error .....	6
21) Enuncie cambios en los datos que tiene mayor probabilidad de inducir un error .....	6
21) Enuncie cambios en los datos que tiene mayor probabilidad de inducir un error .....	6
22) Enuncie acciones a desarrollar cuando se deba realizar mantenimiento de código ajeno ....	7
23) Defina ingeniería inversa.....	7
24) Defina reingeniería.....	7
25) Enuncie puntos a considerar antes de desarrollar un sistema software .....	7
26) Defina nivel de abstracción de un proceso de ingeniería inversa .....	7
27) Defina completitud de un proceso de ingeniería inversa e interactividad en su contexto .....	8

## 1. Defina mantenimiento correctivo.

Durante el uso se encuentran errores que son informados por el equipo de desarrollo. La primera actividad de mantenimiento se llama mantenimiento correctivo e incluye el diagnóstico y la corrección de uno o más errores

## 2) Defina mantenimiento adaptativo

La segunda actividad, el mantenimiento adaptativo, modifica el software para que interactúe adecuadamente con su entorno cambiante. Es tan necesaria como usual.

## 3) Defina los dos tipos de mantenimiento perfectivo

El mantenimiento perfectivo sirve para satisfacer recomendaciones de usuarios acerca de modificaciones, además de cambiar el software para una futura facilidad de mantenimiento o fiabilidad. Contabiliza la mayor cantidad de esfuerzo del mantenimiento, y se divide en ingeniería inversa y reingeniería

## 4) Enuncie puntos de vista para comprender las características del mantenimiento de software

- 1) Las actividades de las fases de mantenimiento y el impacto de la ingeniería de software sobre la eficacia de las actividades
- 2) Los costes asociados a la fase de mantenimiento
- 3) Los problemas encontrados cuando se lleva a cabo el mantenimiento

## 5) Esquematice flujo de sucesos que se pueden dar como resultado de una petición de mantenimiento de software

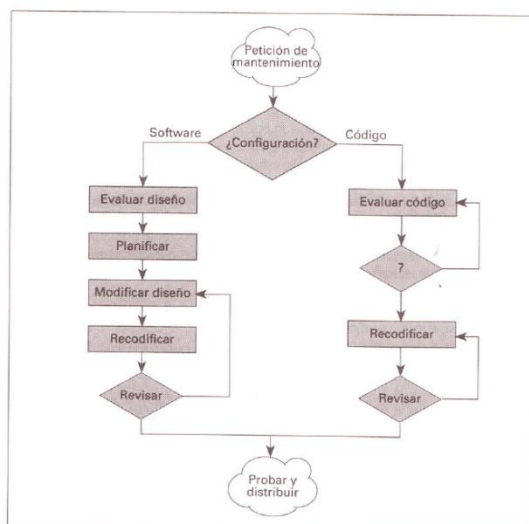


Figura 20.1. Mantenimiento estructurado frente al no estructurado.

## 6) Defina mantenimiento estructurado

La tarea de mantenimiento comienza con la evaluación de la documentación de diseño. Se determinan importantes características estructurales, de rendimiento y de interfaz, luego se estudia el impacto de las correcciones y se traza un plan de actuación. Esto se conoce como mantenimiento estructurado.

## 7. Ejemplifique costes intangibles del mantenimiento del software

- 1) Insatisfacción del cliente cuando una petición de reparación o modificación no se puede atender en un tiempo razonable
- 2) Disminución de la calidad del software por errores latentes introducidos en los cambios
- 3) Trastornos en otros esfuerzos de desarrollo al tener que poner a trabajar la plantilla de tareas de mantenimiento

## 8) Plantee un modelo de esfuerzo del mantenimiento de software

$M = p + Ke(e-f)$  donde: M = Esfuerzo total del mantenimiento

p = Esfuerzo productivo K = Constante

c = Medida de la complejidad por una falta de buen diseño y documentación

f = Familiaridad con el software

## 9) Enuncie problemas asociados con el mantenimiento de software

- 1) Es difícil o imposible seguir la evolución del software a través de varias versiones, porque no se documentan cambios
- 2) Es difícil o imposible seguir el proceso de construcción del software
- 3) Es difícil comprender un programa ajeno
- 4) Ese personaje ajeno no está cerca para explicar lo que hizo
- 5) No existe documentación apropiada o está mal preparada
- 6) La mayoría del software no ha sido diseñado previendo el cambio
- 7) El mantenimiento no se ve como un trabajo atractivo

## 10) Defina facilidad de mantenimiento de software

Se puede definir cualitativamente como la facilidad de comprender, corregir, adaptar o mejorar el software. Guía los pasos de cualquier metodología de software.

## 11) Enuncie factores de control relacionados con el entorno de desarrollo

- 1) Disponibilidad de una plantilla de software cualificada
- 2) Estructura del sistema comprensible
- 3) Facilidad de manejo del sistema
- 4) Uso de lenguajes de programación estandarizados
- 5) Uso de SO estandarizados
- 6) Estructura de documentación estandarizada
- 7) Disponibilidad de casos de prueba
- 8) Facilidad de depuración incorporada

## 12) Enuncie métricas de la facilidad de mantenimiento

- 1) Tiempo de reconocimiento del problema
- 2) Tiempo de retraso administrativo
- 3) Tiempo de recolección de herramientas de mantenimiento
- 4) Tiempo de análisis del problema
- 5) Tiempo de especificación de cambios
- 6) Tiempo activo de corrección o modificación
- 7) Tiempos de prueba local y global
- 8) Tiempos de revisión y tiempo total de recuperación

## 13) De un esquema y describa el funcionamiento de una organización de mantenimiento

Las peticiones se dirigen a un controlador de mantenimiento que remite cada petición o a un supervisor del sistema, quien es un miembro de la plantilla que asigna la responsabilidad de familiarizarse con un conjunto de programas. Luego de evaluación, una autoridad de control de cambios determina las acciones a llevar a cabo.

## 14) Defina formulario de petición de mantenimiento (informe de problemas de software)

Todas las peticiones de mantenimiento se presentan de forma estandarizada. El equipo de desarrollo genera un formulario de petición de mantenimiento, también llamado informe de problemas de software. Es un documento generado exteriormente que se usa como base para planificar las tareas de mantenimiento.

## 15. Defina que informe de cambios de software y que se informa en él.

Internamente, la organización genera un informe de cambios de software (ICS), indicando: 1) La magnitud de esfuerzo requerido para satisfacer el FPM 2) La naturaleza de las modificaciones requeridas 3) La prioridad de la petición 4) Otros datos sobre modificaciones. Luego, el ICS se envía a la autoridad de control de cambio.

## 16) De un esquema del flujo de sucesos del mantenimiento

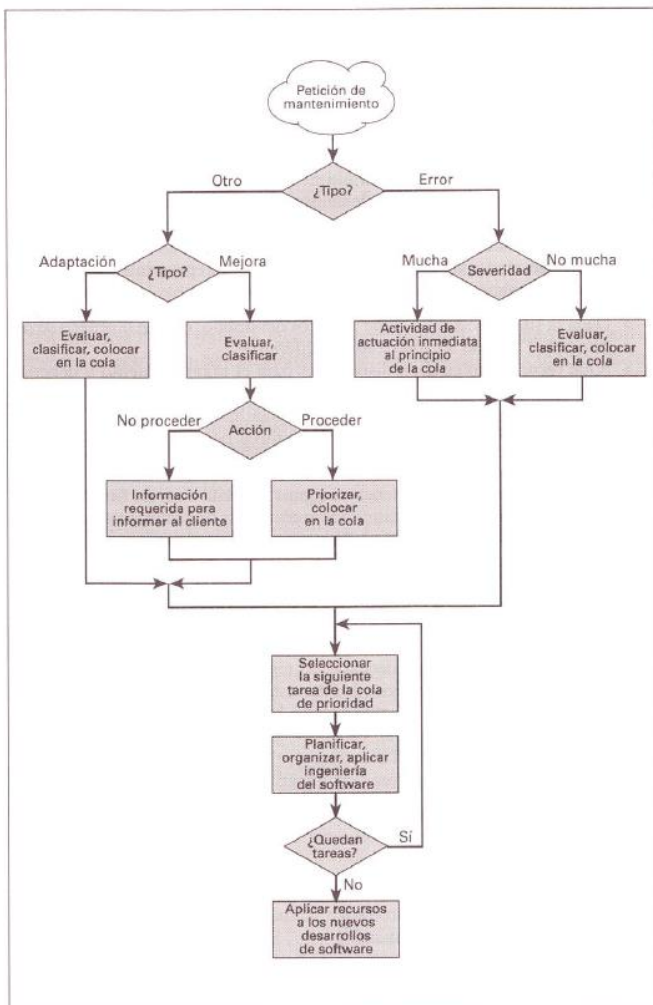


Figura 20.3. Flujo de sucesos del mantenimiento.

## 17) Enuncie preguntas de repaso de la situación pertinente una vez concluida la tarea de mantenimiento

- 1) En la actualidad, ¿Qué aspectos del diseño, código o prueba se podrían haber llevado a cabo de forma diferente?
- 2) ¿Qué recursos no estaban disponibles para el mantenimiento y deberían haber estado?
- 3) ¿Cuáles fueron las mayores dificultades del esfuerzo?
- 4) ¿Es adecuado un mantenimiento preventivo dados los tipos de solicitudes?

## 18) Enuncie información a registrar concluida la tarea de mantenimiento

- 1) Identificación del programa
- 2) Número de sentencias fuente
- 3) Numero de instrucción
- 4) Lenguaje de programación usado
- 5) Fecha de instalación del programa
- 6) Número de ejecuciones desde la instalación
- 7) Numero de fallos asociados al punto anterior
- 8) Número de personas-hora por cambio

## 19) Enuncie medidas del rendimiento del mantenimiento

- 1) Número medio de fallos de procesamiento por ejecución del programa
- 2) Total de personas-hora por cada categoría de mantenimiento
- 3) Número medio de cambios por programa, por lenguaje y por tipo de mantenimiento
- 4) Número medio de personas-hora por sentencia fuente añadida o eliminada
- 5) Medida de personas-hora por lenguaje

## 20) Enuncie cambios de código que tiene mayor probabilidad de inducir a un error

- 1) Un subprograma eliminado o cambiado
- 2) Eliminación o modificación de una sentencia de etiqueta
- 3) Eliminación o modificación de un identificador
- 4) Cambios para mejorar rendimiento en ejecución
- 5) Modificación de apertura o cierre de archivos
- 6) Modificación de operadores lógicos

## 21) Enuncie cambios en los datos que tiene mayor probabilidad de inducir un error

- 1) Redefinición de constantes locales o globales
- 2) Redefinición de formatos o registros de archivos
- 3) Aumento o disminución del tamaño de arrays o estructuras de datos

## 21) Enuncie cambios en los datos que tiene mayor probabilidad de inducir un error

- 1) Redefinición de constantes locales o globales
- 2) Redefinición de formatos o registros de archivos
- 3) Aumento o disminución del tamaño de arrays o estructuras de datos
- 4) Modificación de datos globales
- 5) Re inicialización de indicadores de control
- 6) Reorganización de argumentos de E/S

## 22) Enuncie acciones a desarrollar cuando se deba realizar mantenimiento de código ajeno

- 1) Estudiar el programa antes de entrar en modo emergencia
- 2) Tratar de familiarizarse con el flujo de control general del programa, ignorando los detalles de la codificación
- 3) Evaluar las características de la documentación, insertar comentarios si se cree que serán de ayuda
- 4) Hacer buen uso de los listados de referencias cruzadas
- 5) No eliminar código hasta estar totalmente seguro
- 6) Insertar comprobación de errores
- 7)

## 23) Defina ingeniería inversa

En el caso del hardware, una compañía desensambla un producto de la competencia para encontrar los secretos del diseño. En el caso del software, se trata de un producto propio al que no se le realizó una especificación, por lo que ingeniería inversa es el proceso de analizar el programa en un esfuerzo por crear una representación de mayor nivel de abstracción que el código fuente. Es un proceso de recuperación de diseño.

## 24) Defina reingeniería

No solo recupera la información de diseño, sino que usa esa información para alterar o reconstruir el sistema, para mejorar la calidad general. El software tiene la misma función, pero se añaden nuevas funciones o mejora el rendimiento

## 25) Enuncie puntos a considerar antes de desarrollar un sistema software

- 1) El coste de mantener una línea de código es de 20 a 40 veces el coste de desarrollo inicial de esa línea
- 2) Rediseñar la arquitectura con metodologías modernas pueden facilitar el mantenimiento
- 3) Debido a la existencia de un prototipo del software, la productividad puede ser mucho mayor que la media
- 4) El usuario ya tiene experiencia con el software, es más fácil descubrir nuevos requisitos
- 5) Las herramientas CASE pueden automatizar parte del trabajo

## 26) Defina nivel de abstracción de un proceso de ingeniería inversa

Es la sofisticación de la información de diseño que se pueda extraer del código fuente. Idealmente, el nivel de abstracción debería ser el mayor posible, es decir que se deben poder obtener representaciones de diseño procedimental, estructura, y modelos de flujo de datos.

## 27) Defina completitud de un proceso de ingeniería inversa e interactividad en su contexto

Es el nivel de detalle que proporcione en el nivel de abstracción. En la mayoría de los casos, la completitud decrece cuando aumenta la abstracción. La completitud mejora cuantos más análisis realice la persona que realiza la ingeniería inversa. La interactividad se refiere al grado en que la persona está integrada con las herramientas automáticas para crear un proceso efectivo de ingeniería inversa. Cuando aumenta la abstracción, aumenta la interactividad para que no sufra la completitud.

## 28) Defina direccionalidad de un proceso de ingeniería inversa

Si la direccionalidad tiene un solo sentido, toda la información extraída del código fuente va al ingeniero para que la use en el mantenimiento. Si es de dos sentidos, la información aumenta una herramienta de reingeniería que intenta reestructurar o regenerar el programa antiguo