

## AUDITORÍA DEL MANTENIMIENTO

---

*Juan Carlos Granja Álvarez*

### 13.1. INTRODUCCIÓN A LA AUDITORÍA INFORMÁTICA DEL MANTENIMIENTO DEL SOFTWARE

Nunca se ha prestado demasiada atención al estudio de la Auditoría Informática en esta etapa, ni se le ha dedicado el esfuerzo necesario que por su alto nivel de coste merece. En ocasiones se ha hablado de una etapa en la que sólo se apercebían parte de los problemas y apenas se empleaba un mínimo esfuerzo en aplicar técnicas de auditoría con lo que surgía el efecto ICEBERG con el que algunos autores han denominado al hecho de que sólo se aprecia una pequeña parte de la problemática que encierra.

Varias investigaciones y experiencias revelan que la etapa de mantenimiento consume la mayor parte de los recursos empleados en un proyecto software. Por tanto, esta etapa debe ser especialmente considerada en los estudios de productividad y de la Auditoría Informática. La mantenibilidad, factor crítico de estudio en Auditoría Informática del Mantenimiento, es el factor de calidad que engloba todas aquellas características del software destinadas a hacer que el producto sea más fácilmente mantenible y, en consecuencia, a conseguir una mayor productividad durante la etapa de mantenimiento. En este capítulo se propone un modelo empírico de estimación de costes de mantenimiento centrado en este factor de calidad, así como el método para su implementación. Finalmente se consideran algunos casos prácticos que refuerzan la validez del modelo.

El control y evaluación de la Mantenibilidad puede ser uno de los factores determinantes en el estudio de la Auditoría Informática en la Etapa de Mantenimiento del Software.

Frecuentemente se olvida que los esfuerzos de auditoría en la etapa de Mantenimiento se plasman en las primeras etapas de desarrollo del software. En las especificaciones del software y en la llamada Ingeniería de Requisitos, se plasman los primeros pasos de los aspectos que van a determinar el esfuerzo o no dificultad de mantenimiento del software.

Podemos decir que la Mantenibilidad va a ser un factor determinante para la Auditoría Informática del Mantenimiento del Software. Vamos a centrar el estudio de este tema en todo lo que rodea la facilidad de mantenimiento del software y los aspectos a auditar.

Es frecuente que las empresas de software busquen la máxima productividad en el desarrollo de sus productos, dejando en un segundo lugar a la etapa de mantenimiento. Esto constituye un lamentable error ya que, como muchos estudios revelan, esta etapa es la que más recursos consume (más del 60% de los recursos empleados en todo el proyecto) [CANN72, WIEN84, HARR90]. Todo ello nos lleva a un profundo estudio de las técnicas de Auditoría en esta etapa.

Si la productividad en la etapa de mantenimiento es baja puede suceder, además de las evidentes implicaciones económicas, que el equipo humano que desarrolló el producto tenga que dedicarse a tiempo completo a su mantenimiento. Consecuentemente, si la empresa quiere abordar nuevos proyectos tendrá que incluir un nuevo equipo en su plantilla. Esto implica el desaprovechamiento, al menos parcial, de la experiencia adquirida por el equipo anterior, que sería de gran valor en los nuevos proyectos. Por otro lado se requiere una labor de formación del nuevo equipo hasta adquirir el conocimiento necesario sobre los métodos y herramientas utilizados por la empresa de software.

La productividad en la etapa de mantenimiento está directamente relacionada con la mantenibilidad del producto. La mantenibilidad es un factor de calidad que engloba todas aquellas características del software destinadas a hacer que el producto sea más fácilmente mantenible. Por tanto, va a ser un parámetro decisivo a la hora de auditar esta etapa.

Se propone un modelo de estimación del costo de mantenimiento que permite aprovechar las experiencias en proyectos previos. Se toma como punto de partida el conocido modelo de estimación de costes elaborado por Boehm (COCOMO), al que se incorporan unos índices que miden la mantenibilidad del producto y que afectan de manera importante al coste de mantenimiento.

## 13.2. LISTAS DE COMPROBACIÓN EN AUDITORÍA INFORMÁTICA DEL MANTENIMIENTO

Siguiendo un enfoque clásico de la Auditoría Informática del Mantenimiento, nos encontramos con las técnicas de utilización de diferentes tipos de listas de comprobación.

Cara a la revisión del software en la etapa de mantenimiento, podríamos resaltar cinco grandes bloques o enfoques hacia los cuales poder orientar las preguntas:

1. ¿Se han tenido en cuenta las implicaciones laterales asociadas con el cambio?
2. ¿Se han tenido en cuenta los aspectos documentales en cuanto a evaluar y aprobar la petición de cambios?
3. ¿Se ha documentado el cambio, una vez realizado y procediéndose a dar información a todos los que se ven implicados en el proceso?
4. En cuanto a las revisiones técnicas formales, ¿se han realizado las adecuadas?
5. ¿Se ha hecho una revisión de aceptación final para asegurar que toda la arquitectura software, fue actualizada y probada y se procedió a los cambios adecuadamente?

La utilización de grandes bloques como los mencionados nos va a permitir centrar nuestro esfuerzo de auditoría informática, si bien vemos que la problemática persiste en buscar aquellos aspectos que con el menor esfuerzo de auditoría nos permitan llegar a auditar y conseguir la mayor cantidad de información que sea posible.

Surge así la necesidad de centrar el esfuerzo de auditoría en un factor que pueda ser determinante, tal como es la Mantenibilidad en la etapa de Mantenimiento del Software.

## 13.3. MODELIZACIÓN EN LA ETAPA DE MANTENIMIENTO

Podemos tomar como referente el COCOMO (COConstructive COSt MOdel), que es un modelo de estimación de costes de proyectos software creado por Boehm en 1981 a partir de datos recogidos de 63 proyectos [BOEH81]. El importante número de proyectos tratados y la esmerada elaboración del modelo hacen que su validez perdure

hasta la actualidad. Este modelo ofrece fórmulas empíricas de estimación de costes y esfuerzos software.

Tras aplicar la versión inicial del modelo a una amplia variedad de entornos se comprobó que no bastaba con un único modo de desarrollo, por lo que se plantearon tres modos (*orgánico*, *semidetached* y *embedded*) en función de varias características: tamaño, necesidades de comunicación, experiencia en proyectos similares, etc.

Por otro lado, se ofrecen tres versiones del modelo: básico, intermedio y detallado. El básico es adecuado para estimaciones rápidas, aunque sin una gran precisión. El intermedio considera 15 atributos del proyecto (fiabilidad requerida, tamaño de la base de datos, restricciones de memoria, tiempo de respuesta requerido, etc.) cuya valoración actúa como factor multiplicador en el modelo. La versión detallada considera las estimaciones en cada una de las etapas del ciclo de vida del proyecto.

La versión básica del modelo ofrece las siguientes fórmulas de cálculo del **esfuerzo de desarrollo** (medido en MM=month-man u hombre mes):

Modo orgánico	$MM_{DES} = 2.4 KS^{1.05}$
Modo semidetached	$MM_{DES} = 3.0 KS^{1.12}$
Modo embedded	$MM_{DES} = 3.6 KS^{1.20}$

Siendo KS = Estimación del tamaño del programa (en miles de líneas).

Para la estimación del **esfuerzo de mantenimiento** se necesita un nuevo parámetro: el Tráfico de Cambio Anual (TCA), que consiste en la proporción de instrucciones fuente que sufren algún cambio durante un año, bien sea por adición o por modificación.

$$TCA = \frac{NLN + NLM}{NLI}$$

NLN	= Número de líneas nuevas
NLM	= Número de líneas modificadas
NLI	= Número de líneas inicial

Así, el esfuerzo en la etapa de mantenimiento, según el modelo COCOMO, viene dado como producto del esfuerzo de desarrollo y el tráfico de cambio anual.

$$MM_{MANT} = TCA MM_{DES}$$

## 13.4. MODELO DE ESTIMACIÓN EN EL MANTENIMIENTO

La mantenibilidad es, sin duda, el factor de calidad del software con mayor influencia en la etapa de mantenimiento y, por tanto, elemento decisivo de referencia en los estudios de Auditoría Informática del Mantenimiento. Un estudio realizado por W. Itzfeld en Alemania, recogido por Wallmüller en [WALL94] presenta un *ranking*

de utilización de métricas de calidad en el cual las métricas de mantenibilidad se encuentran en primer lugar, empleadas por un 67% de los encuestados.

Boehm [BOEH79] reconocía la importancia de la mantenibilidad. Uno de sus estudios indicaba que el esfuerzo de mantenimiento de un software de baja mantenibilidad puede estar en relación de 40 a 1 con respecto al esfuerzo de nuevos desarrollos. Es decir, existe una relación de dependencia entre las características de mantenibilidad del software desarrollado y el esfuerzo de mantenimiento, lo cual es bastante evidente.

Por tanto, para el cálculo del coste estimado  $d$ : mantenimiento hemos de considerar un factor que indique el grado de mantenibilidad o facilidad de mantenimiento del producto. Tomando como punto de partida la fórmula de estimación del esfuerzo de mantenimiento del modelo COCOMO de Boehm, se va a incluir en ella dicho factor que denominamos **índice de mantenibilidad**, y que va a ser función de algunas medidas del software desarrollado:

$$MM_{MANT} = TCA \cdot MM_{DES} \cdot I_{MANT}$$

$$I_{MANT} = f(X_1, X_2, \dots, X_n)$$

Este índice va a mostrar el grado de mantenibilidad o facilidad de mantenimiento del producto de forma que valores grandes expresan baja mantenibilidad mientras que los valores bajos indican alta mantenibilidad. Al mismo tiempo va a ser un buen indicador de la productividad en la etapa de mantenimiento.

Así pues, nuestro principal objetivo consiste en determinar qué forma ha de tener este índice. Dicho de otro modo, se trata de obtener la relación que existe entre el esfuerzo estimado de mantenimiento y aquellas características que hacen que el producto sea más o menos mantenible.

Dos son, pues, los pasos a seguir para la normalización del modelo:

- Establecimiento de las métricas de mantenibilidad.
- Obtención de las funciones de mantenibilidad que relacionan las métricas establecidas con el índice de mantenibilidad.

Previamente a abordar estos dos puntos y teniendo en cuenta las tres actividades que conforman una acción de mantenimiento, el índice de mantenibilidad se va a descomponer a su vez en tres índices: índice de comprensibilidad, índice de modificabilidad e índice de testeabilidad.

### 13.4.1. Elementos de la mantenibilidad

Una acción de mantenimiento se puede descomponer en tres actividades:

- **Comprensión** del cambio a realizar.
- **Modificación** o realización del cambio.
- **Prueba** de colección del cambio realizado.

Son tres tareas claramente diferenciadas que se realizan una tras otra, por lo que el esfuerzo de mantenimiento se puede considerar como suma de los tres esfuerzos: comprensión, modificación y prueba.

$$MM_{MANT} = MM_C + MM_M + MM_T$$

Así pues, vamos a tener tres índices de mantenibilidad,  $I_C$ ,  $I_M$  e  $I_T$  que relacionan los parámetros del proyecto, TCA y  $MM_{DES}$  con los tres componentes del esfuerzo de mantenimiento:  $MM_C$ ,  $MM_M$  y  $MM_T$

$$MM_C = TCA \cdot MM_{DES} \cdot I_C$$

$$MM_M = TCA \cdot MM_{DES} \cdot I_M$$

$$MM_T = TCA \cdot MM_{DES} \cdot I_T$$

En consecuencia, el índice de mantenibilidad,  $IMANT$  vendrá dado por la suma de los tres índices anteriores:

$$I_{MANT} = I_C + I_M + I_T$$

$I_{MANT}$  = Índice de mantenibilidad  
 $I_C$  = Índice de comprensibilidad  
 $I_M$  = Índice de modificabilidad  
 $I_T$  = Índice de testeabilidad

El esfuerzo total de mantenimiento:

$$MM_{MANT} = MM_C + MM_M + MM_T = TCA \cdot MM_{DES} (I_C + I_M + I_T)$$

### 13.4.2. Métricas de mantenibilidad

El modelo aquí propuesto, y que ha sido empleado en los casos de estudio, considera tres características, cada una de las cuales afecta de manera directa a un componente de la mantenibilidad:

$X_C$ : *Métrica de comprensibilidad*: Número de líneas de comentario por cada 100 líneas de código. La estrecha relación entre la documentación interna del código (o autodocumentación) y el esfuerzo de comprensión es evidente.

$X_M$ : *Métrica de modificabilidad*: Número de líneas sin datos constantes por cada 100 líneas de código. La existencia de un gran número de datos constantes en el código implica un mayor esfuerzo para la modificación.

$X_T$ : *Métrica de testeabilidad*: Número de líneas de tratamiento de errores por cada 100 líneas de código. La depuración o *testing* del código va a ser más fácil si existen procedimientos de detección y manejo de errores.

Las tres características se han elegido de manera que resulten fácilmente medibles y que tengan una gran influencia sobre la mantenibilidad. No obstante, el modelo puede aplicarse cualquiera que sea el conjunto de métricas escogido, siempre que quede demostrada la dependencia entre dichas métricas y el componente de mantenibilidad correspondiente.

### 13.4.3. Funciones de mantenibilidad

Las funciones así denominadas relacionan los índices de mantenibilidad ( $I_C$ ,  $I_M$  e  $I_T$ ) con las métricas recién comentadas ( $X_C$ ,  $X_M$  y  $X_T$ ).

$$I_C = F_C(X_C)$$

$$I_M = F_M(X_M)$$

$$I_T = F_T(X_T)$$

Para la obtención de estas funciones se hace necesario el empleo de un elemento que resulta fundamental en toda estimación: la información histórica. La experiencia adquirida en proyectos anteriores adquiere un gran valor al emprender nuevos proyectos. Por tanto, se ha de disponer de mecanismos que permitan tomar varias medidas:

a) Del producto desarrollado

$X_C$ : Métrica de comprensibilidad

$X_M$ : Métrica de modificabilidad

$X_T$ : Métrica de testeabilidad

b) Del proceso de mantenimiento

$MM_C$ : Esfuerzo de comprensión

$MM_M$ : Esfuerzo de modificación

$MM_T$ : Esfuerzo de prueba

Los índices de mantenibilidad se obtienen a partir de los valores de esfuerzo mediante la siguiente fórmula:

$$I_C = \frac{MM_C}{TCA * MM_{DES}}$$

$I_C$  = Índice de comprensibilidad

$MM_C$  = Esfuerzo de comprensión en mantenimiento

$TCA$  = Tráfico de cambio anual

$MM_{DES}$  = Esfuerzo de desarrollo

Del mismo modo se obtienen  $I_M$  e  $I_T$ , considerando el esfuerzo de modificación  $MM_M$  y el esfuerzo de prueba  $MM_T$  respectivamente.

Toda la información necesaria para la aplicación del modelo, ya comentada, puede incluirse en una tabla que denominamos Tabla Histórica (TH) con la siguiente estructura:

Proyecto	TCA	$MM_{DES}$	$X_C$	$MM_C$	$I_C$	$X_M$	$MM_M$	$I_M$	$X_T$	$MM_T$	$I_T$
$P_1$	$TCA_1$	$MM_{DES1}$	$X_{C1}$	$MM_{C1}$	$I_{C1}$	$X_{M1}$	$MM_{M1}$	$I_{M1}$	$X_{T1}$	$MM_{T1}$	$I_{T1}$
...	...	...	...	...	...	...	...	...	...	...	...
$P_n$	$TCA_n$	$MM_{DESn}$	$X_{Cn}$	$MM_{Cn}$	$I_{Cn}$	$X_{Mn}$	$MM_{Mn}$	$I_{Mn}$	$X_{Tn}$	$MM_{Tn}$	$I_{Tn}$

Tomando la subtabla formada por las columnas  $X_C$  e  $I_C$  tenemos una nube de puntos representable en un plano de dos dimensiones  $\{(X_{Ci}, I_{Ci}) / i=1..n\}$ . Haciendo un sencillo análisis de regresión sobre este conjunto de puntos se puede obtener la curva que mejor se ajusta, así como el coeficiente de determinación o grado en que dicha función es representativa de dicho conjunto de puntos. Así obtendríamos la función  $F_C$ .

Del mismo modo llegaríamos a las funciones  $F_M$  y  $F_T$  a partir de los conjuntos de puntos  $\{(X_{Mi}, I_{Mi}) / i=1..n\}$  y  $\{(X_{Ti}, I_{Ti}) / i=1..n\}$ .

#### 13.4.4. Método de implementación

En este apartado se describe el método a seguir para implementar el modelo en un proyecto software. La figura 13.1 muestra los elementos y procesos que intervienen en el modelo.





### 13.4.4.2. Estimación del TCA

Éste es un proceso que ha de basarse en la experiencia. Dos son los elementos básicos en todo proceso experimental: la información histórica y el juicio de expertos.

El método aquí propuesto se sirve de estos dos elementos para obtener la estimación. Partimos de la existencia de un conjunto de **cualidades** atribuibles a un proyecto software. Este conjunto va a ser elaborado inicialmente y revisado de forma periódica por los expertos de forma que manifieste las características distintivas de los proyectos que componen la tabla de datos históricos. Cada cualidad  $j$  va a tener un **peso**  $p_j$  que permite valorar unas cualidades más que otras. Cada proyecto tendrá sólo dos posibilidades con respecto a cada cualidad: poseerla o no poseerla. Así, si la tabla histórica está compuesta por  $n$  proyectos, tendremos la siguiente información que representamos en forma matricial:

#### *Información histórica*

**A** Matriz de  $n \times m$  elementos que indica las cualidades de cada proyecto que compone la tabla histórica (TH)

$$A = \begin{pmatrix} C_{11} & C_{12} & \dots & C_{1m} \\ C_{21} & C_{22} & \dots & C_{2m} \\ \dots & \dots & & \dots \\ C_{n1} & C_{n2} & \dots & C_{nm} \end{pmatrix}$$

$C_{ij}$  = Cualidad  $j$  para el proyecto  $i$

Dos valores posibles:

- 1: El proyecto posee la cualidad
- 0: En otro caso

**T** Matriz de  $n \times 1$  elementos que indica el tráfico de cambio anual de cada proyecto de la TH

$$T = (TCA_1, TCA_2, \dots, TCA_n)^T \quad TCA_i = \text{Tráfico de cambio anual de proyecto } i$$

NOTA: El superíndice T indica "matriz transpuesta".

#### *Información del proyecto en estudio*

**C** Matriz de  $1 \times m$  elementos que indica las cualidades del proyecto en curso. La extracción de esta información requiere la intervención de personal experto. Es en este momento cuando se va a revisar el conjunto de cualidades. La modificación de este conjunto requiere la actualización de la tabla histórica revisando las cualidades de todos los proyectos que la componen.

$$C = (c_1, c_2, \dots, c_m)$$

$c_j$  = Calidad  $j$  para el proyecto en curso  
 Dos valores posibles:  
 $p_j$  : El proyecto posee la calidad  
 0 : En otro caso.

**B** Matriz de  $n \times l$  elementos que indica el número de coincidencias que tiene el proyecto en curso con respecto a cada proyecto de la **TH**, es decir, el número de cualidades que tienen en común.

$$B = A * C^T$$

NOTA: El símbolo  $*$  representa el producto de matrices.

El TCA estimado viene dado por la siguiente expresión:

$$TCA = \frac{B * T^T}{B^T * B}$$

De esta forma, cada proyecto interviene en el cálculo de la estimación en la medida en que sus cualidades coinciden con las del proyecto en estudio

### 13.4.4.3. Aplicabilidad del modelo

Una vez que se dispone de las funciones de mantenibilidad (FC, FM y FT) así como del TCA estimado, el coste estimado de mantenimiento se obtiene sólo con aplicar la fórmula ya conocida:

$$MM_{MANT} = MM_C + MM_M + MM_T = TCA \cdot MM_{DES} (I_C + I_M + I_T)$$

siendo

$$I_C = F_C(X_C)$$

$$I_M = F_M(X_M)$$

$$I_T = F_T(X_T)$$

Por tanto, el proceso de desarrollo ha de suministrar la siguiente información:

$MM_{DES}$  : Esfuerzo de desarrollo  
 $X_C$  : Métrica de comprensibilidad  
 $X_M$  : Métrica de modificabilidad  
 $X_T$  : Métrica de testeabilidad

Con toda esta información recogida se podrá aplicar la fórmula y obtener la estimación del esfuerzo o coste de mantenimiento.

### 13.5. CASO DE ESTUDIO

Se han estudiado tres proyectos con el fin de aplicar el modelo recién expuesto. Se trata de un proyecto para el desarrollo de un paquete de gestión contable ( $P_1$ ) y dos de gestión comercial ( $P_2$  y  $P_3$ ). El estudio se ha simplificado considerando solamente uno de los componentes de la mantenibilidad, a saber, la comprensibilidad. El estudio de la modificabilidad y de la testeabilidad se haría de manera idéntica.

Seguidamente se muestra la tabla histórica en la que intervienen los tres proyectos citados. En ella, todos los datos han sido medidos excepto el índice de comprensibilidad,  $I_C$ , que se obtiene mediante la fórmula:

$$I_C = \frac{MM_C}{TCA \cdot MM_{DES}}$$

Hemos de mencionar también que  $C_1$  y  $C_2$  son las cualidades escogidas para diferenciar los proyectos en base a su incidencia en el tráfico de cambio anual:

- $C_1$  :- Proyecto de gestión contable  
 $C_2$  :- Proyecto de gestión comercial

Asignamos igual peso a ambas cualidades e igual a la unidad ( $p_1 = 1$ ,  $p_2 = 1$ )

Proyecto	$C_1$	$C_2$	TCA	$MM_{DES}$	$X_C$	$MM_C$	$I_C$
$P_1$	1	0	0,23	48	14	6,6	0,60
$P_2$	0	1	0,29	72	11	15,7	0,75
$P_3$	0	1	0,30	24	115	3,8	0,53

El proyecto en estudio,  $P_4$ , tiene como finalidad el desarrollo de un paquete de gestión comercial. Su etapa de desarrollo ha concluido. El coste del desarrollo ha sido de 57 Hombres x Mes y la métrica de comprensibilidad,  $X_C$ , tiene un valor de 17.

#### a) Obtención de la función de comprensibilidad ( $F_C$ )

Como cabe esperar, según aumenta el valor de la métrica de comprensibilidad (número de líneas de comentario), el índice de comprensibilidad (directamente proporcional al esfuerzo de comprensión) va a disminuir. Por tanto, para el análisis de regresión del conjunto de puntos  $\{(X_C, I_C)\}$  hay dos modelos bastante evidentes con

los cuales ensayar: el modelo lineal de pendiente negativa y el modelo exponencial negativo.

De los dos modelos, el exponencial negativo es el más adecuado ya que, normalmente, la mejora de comprensión que supone una nueva línea de comentario va a ser mayor cuanto menor sea la concentración de líneas de comentario en el programa. Esto se comprende perfectamente yéndonos a los límites, es decir, viendo lo que sucede si se añade una línea de comentario en:

- a) Un programa sin ninguna documentación interna
- b) Un programa con una documentación interna perfecta.

Es evidente que la comprensión en el caso (a) va a verse mejorada en una cuantía mucho mayor que en el caso (b).

Empleando el método de ajuste por mínimos cuadrados, tenemos que se trata de:

$$\text{minimizar } \sum_{i=1}^n (I_{ci} - (X_{ci}))$$

Por su sencillez, vamos a considerar en primer lugar el caso de ajuste a una función lineal:

$$\text{minimizar } \sum_{i=1}^n (I_{ci} - (a + bX_{ci}))$$

Derivando parcialmente esta expresión respecto de  $a$  e igualando a 0, y por otro lado, derivando parcialmente respecto de  $b$  e igualando a 0, se obtiene el siguiente sistema de ecuaciones (en el que simplificamos la notación no incluyendo los límites de los sumatorios que siempre son  $i=1$  hasta  $n$ ):

$$\left. \begin{aligned} \sum I_{ci} &= aN + b \sum X_{ci} \\ \sum X_{ci} I_{ci} &= a \sum X_{ci} + b \sum X_{ci}^2 \end{aligned} \right\}$$

Consideremos ahora la función exponencial:

$$I_{ci} = ae^{-bX_{ci}}$$

Aplicando logaritmos obtenemos:

$$\ln I_{C_i} = \ln a + b X_{C_i}$$

Por tanto, volvemos a tener una función lineal donde la variable independiente es  $X_{C_i}$ , y la variable dependiente es  $\ln I_{C_i}$ . Haciendo el cambio de variable  $I'_{C_i} = \ln I_{C_i}$  así como  $a' = \ln a$  obtenemos el siguiente sistema a resolver:

$$\left. \begin{aligned} \sum I'_{C_i} &= a'N + b \sum X_{C_i} \\ \sum X_{C_i} I'_{C_i} &= a' \sum X_{C_i} + b \sum X_{C_i}^2 \end{aligned} \right\}$$

Los datos requeridos se muestran en la siguiente tabla:

Proyecto	$X_C$	$I_C$	$I'_C = \ln I_C$	$X_C^2$	$X_C I'_C$
P <sub>1</sub>	14	0,60	-0,51	196	-7,14
P <sub>2</sub>	11	0,75	-0,29	121	-3,19
P <sub>3</sub>	15	0,53	-0,63	225	-9,45
Sumas	40	1,88	1,43	542	-19,78

Sustituyendo y resolviendo el sistema de ecuaciones resultante, obtenemos los valores:

$$a = 1,86$$

$$b = -0,08$$

Por tanto, la función de comprensibilidad ( $F_C$ ) obtenida tiene la forma:

$$I_C = F_C(X_C) = 1,86 e^{-0,08 X_C}$$

## b) Obtención del tráfico de cambio anual estimado (TCA)

Vamos a construir las matrices A, T, C y B de acuerdo al procedimiento ya expuesto:

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} T = \begin{pmatrix} 0,23 \\ 0,29 \\ 0,30 \end{pmatrix} C = (0 \quad 1) B = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

Aplicando la fórmula de cálculo de TCA tenemos:

$$TCA = \frac{B \cdot T^T}{B^T \cdot B} = \frac{0,29 + 0,30}{2} = 0,30$$

c) **Aplicación del modelo al proyecto P<sub>4</sub> para el cálculo del coste de comprensión estimado en la etapa de mantenimiento (MM<sub>C</sub>)**

$$MM_C = TCA \cdot MM_{DES} \cdot I_C$$

$$MM_C = 0,30 \cdot 57 \cdot 1,86 \cdot e^{-0,08 \cdot 17} = 8,16 \text{ Hombres x Mes}$$

Como ya se ha comentado, en el caso de estudio se ha considerado sólo un componente de la mantenibilidad. Para los otros dos componentes, el procedimiento sería idéntico. El coste o esfuerzo estimado de mantenimiento se obtendría como suma de los tres costes, comprensión (MM<sub>C</sub>), modificación (MM<sub>M</sub>) y prueba (MM<sub>T</sub>).

## 13.6. CONCLUSIONES

Como hemos podido comprobar, mediante el estudio de la mantenibilidad y auditando la utilización en el proyecto software de las técnicas que permitan asegurar unos niveles de mantenibilidad, podremos fijar el campo de actuación de la Auditoría Informática en el Mantenimiento.

## 13.7. LECTURAS RECOMENDADAS

La publicación periódica *The Journal of Information Systems Audit and Control Association* de la ISACA.

La obra de Weber R. *EDP Auditing. Conceptual Foundations and Practice*, 2ª ed., editada por McGraw-Hill, Sydney, 1988.

Por último podríamos mencionar por el campo de su utilización la obra de Buttery R. Hurford C. y Simpson R.K. de la *Internal Audit in the Public Sector*, publicada por ICSA cop en 1993.

## 13.8. CUESTIONES DE REPASO

1. Exponga las razones que hacen de la auditoría del mantenimiento un área especialmente crítica.
2. Desarrolle una lista de comprobación que recija los aspectos más importantes a la hora de evaluar la gestión de cambios.
3. ¿Qué ventajas aporta una herramienta de gestión de configuración a la hora de auditar el mantenimiento de sistemas informáticos?
4. Aplique las métricas propuestas en este capítulo a algún sistema real, calibrándolas si fuera necesario a su entorno específico.
5. ¿Qué factores pueden influir en la modificabilidad de los programas?
6. Analice en la literatura existente diversas métricas de complejidad y describa su influencia en la mantenibilidad.
7. Existen herramientas específicas para la gestión de pruebas de software, analice su impacto en la testeabilidad.
8. ¿Cómo debería organizarse la gestión de incidencias de mantenimiento en un departamento de informática desde el punto de vista de la auditoría?
9. La influencia de la documentación en el mantenimiento de los sistemas parece obvia, pero ¿cómo mediría la documentación existente sobre un sistema?
10. Compare otros modelos de estimación que conozca con el propuesto en este capítulo.