

Auditoría del Mantenimiento

Introducción

Debido a su elevado costo y a que se pensaba que era una etapa en la que sólo surgían una pequeña parte de los problemas, no se solía prestar demasiada atención a la auditoría del mantenimiento. Sin embargo sólo se estaba apreciando un fragmento de la problemática (efecto iceberg). Incluso, varias investigaciones revelaron que la etapa de mantenimiento consume la mayor parte de los recursos empleados en un proyecto.

La mantenibilidad es el factor de calidad que engloba todas las características del software destinadas a hacer que el producto sea más fácilmente mantenible y, en consecuencia, conseguir una mayor productividad durante la etapa de mantenimiento.

La productividad en la etapa de mantenimiento está directamente relacionada con la mantenibilidad del producto, por tanto va a ser un parámetro decisivo a la hora de auditar esta etapa. A menudo se olvida que los esfuerzos de auditoría en la etapa de mantenimiento se plasman en las primeras etapas de desarrollo, esto es en la ingeniería de requisitos.

Listas de comprobación

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

Modelización

Podríamos tomar como referente al modelo COCOMO (Constructive 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 y que ofrece fórmulas empíricas de estimación de costes o esfuerzos software.

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 - $MM(\text{desarrollo}) = 2,4 * KS^{1,06}$
- Modo semidetached - $MM(\text{desarrollo}) = 3,0 * KS^{1,12}$
- Modo embedded - $MM(\text{desarrollo}) = 3,6 * KS^{1,20}$

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.

$$TCA = (NLN + NLM) / NLI$$

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

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

$$MM(\text{mantenimiento}) = TCA * MM(\text{desarrollo})$$

Modelo de estimación

Para el cálculo del coste estimado de mantenimiento tenemos que considerar un factor que indique el grado de mantenibilidad o facilidad de mantenimiento del producto. Partiendo de la fórmula que se obtuvo en el apartado anterior, se incluirá en ella el índice de mantenibilidad, y que va a ser función de algunas medidas del software desarrollado:

$$MM(\text{mantenimiento}) = TCA * MM(\text{desarrollo}) * I(\text{mantenibilidad})$$

$$I(\text{mantenibilidad}) = f(X_1, X_2, \dots, X_n)$$

Este índice muestra el grado de facilidad de mantenimiento del producto de forma que valores grandes expresan baja mantenibilidad mientras que los valores bajos indican alta mantenibilidad.

Así pues, se intenta determinar la relación que existe entre el esfuerzo estimado de mantenimiento y aquellas características que hacen que el producto sea más o menos mantenible.

Ahora para normalizar el modelo se debe:

1. Establecer las métricas de mantenibilidad.
2. Obtener las funciones de mantenibilidad que relacionan las métricas establecidas con el índice de mantenibilidad.

Previo a estos pasos se descompone el índice de mantenibilidad en otros tres índices:

1. Índice de comprensibilidad: Comprensión del cambio a realizar.
2. Índice de modificabilidad: Modificación o realización del cambio.
3. Índice de testeabilidad: Prueba de colección del cambio realizado.

De esta manera se obtiene que $MM(\text{mantenimiento}) = MM(\text{comprensibilidad}) + MM(\text{modificabilidad}) + MM(\text{testeabilidad})$

Así tendremos tres índices que relacionan los parámetros del proyecto, TCA y MM (desarrollo) con los tres componentes del esfuerzo de mantenimiento:

• $MM(\text{comprensibilidad}) = TCA * MM(\text{desarrollo}) * I(\text{comprensibilidad})$

• $MM(\text{modificabilidad}) = TCA * MM(\text{desarrollo}) * I(\text{modificabilidad})$

• $MM(\text{testeabilidad}) = TCA * MM(\text{desarrollo}) * I(\text{testeabilidad})$

El esfuerzo total de mantenimiento estará dado por: $MM(\text{mantenimiento}) = TCA * MM(\text{desarrollo}) * (I(\text{comprensibilidad}) + I(\text{modificabilidad}) + I(\text{testeabilidad}))$

Métricas de mantenibilidad

El modelo aquí propuesto considera tres características, cada una de las cuales afecta de manera directa a un componente de la mantenibilidad:

• $X(\text{comprensibilidad})$: Número de líneas de comentario por cada 100 líneas de código. La estrecha relación entre la documentación interna del código y el esfuerzo de comprensión es evidente.

• $X(\text{modificabilidad})$: 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.

• $X(\text{testeabilidad})$: 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.

Funciones de mantenibilidad

Las funciones de mantenibilidad relacionan los índices de mantenibilidad con las métricas recién comentadas. Estas funciones son:

• $I(\text{comprensibilidad}) = Fc(X(\text{comprensibilidad}))$

• $I(\text{modificabilidad}) = Fc(X(\text{modificabilidad}))$

• $I(\text{testeabilidad}) = Fc(X(\text{testeabilidad}))$

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.

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

$I(\text{comprensibilidad}) = MM(\text{comprensión}) / (TCA * MM(\text{desarrollo}))$

Del mismo modo se obtienen los otros índices. Por otro lado, toda la información necesaria para la aplicación del modelo, ya comentada anteriormente, puede incluirse en una tabla histórica de proyectos.

Tomando las métricas y los índices históricos se define una nube de puntos representable en un plano de dos dimensiones. Haciendo un análisis de regresión lineal se puede obtener la curva que mejor se ajusta, que representa la función de mantenibilidad para dicho índice.

Bibliografía

[1] Piattini, M. G. y del Peso, E. (2001) Auditoría informática: Un enfoque práctico. México DF: Alfaomega grupo editor. 2ª edición ampliada y revisada.

[2] Piattini, M. G., del Peso, E. y del Peso, M. (2008) Auditoría de Tecnologías y Sistemas de Información. RA-MA S.A.

[3] https://es.wikipedia.org/wiki/Auditor%C3%ADa_inform%C3%A1tica. Accedido el 31/10/2015.