

TEMA 101. MANTENIMIENTO DE SISTEMAS. MANTENIMIENTO PREDICTIVO, ADAPTATIVO Y CORRECTIVO. PLANIFICACIÓN Y GESTIÓN DEL MANTENIMIENTO.

Actualizado a 05/01/2025

1. CONCEPTO DE MANTENIBILIDAD

Es la capacidad de un producto software de ser modificado. Estas modificaciones incluyen correcciones, mejoras o adaptaciones a cambios de entorno, requisitos o especificaciones funcionales.

Según la terminología ANSI-IEEE, el mantenimiento del software es: *“la modificación de un producto software después de su entrega al cliente o usuario para corregir defectos, para mejorar el rendimiento u otras propiedades deseables, o para adaptarlo a un cambio de entorno”*.

1.1. CARACTERÍSTICAS DE MANTENIMIENTO

- **Analizabilidad:** Capacidad del producto software de diagnosticar sus deficiencias o causas de fallos, o de identificar las partes que deben ser modificadas.
- **Cambiabilidad:** Capacidad del producto software de permitir implementar una modificación previamente especificada
- **Estabilidad:** Capacidad del producto software de minimizar los efectos inesperados de las modificaciones (resistencia al efecto dominó)
- **Comprobabilidad:** Capacidad del producto software de permitir evaluar las partes modificadas
- **Conformidad:** Capacidad del producto software de satisfacer los estándares o convenciones relativas a la mantenibilidad.

1.2. ACTIVIDADES DE MANTENIMIENTO

Se basa en la flexibilidad de reparación del código. Se agrupan en tres categorías funcionales:

- **Comprensión:** estudiando las peticiones, la funcionalidad, la documentación y el código
- **Corrección:** modificación del SW (incluye el diseño y codificación) y actualización de la documentación.
- **Comprobación:** mediante la realización de pruebas selectivas, inspección y certificación.

1.3. DIFICULTAD DEL MANTENIMIENTO

- Código heredado (*Legacy*), documentación desfasada o no existente
- Efecto dominó (*“ripple effect”*), a consecuencia del cambio se deben realizar otros adicionales.
- Impacto sobre datos y documentación.

2. MARCOS DE MANTENIMIENTO SW

Se establecen marcos de trabajo con procedimientos claramente definidos y estandarizados para el mantenimiento software. Mediante la asignación de recursos, se realiza una planificación y gestión del mantenimiento SW.

IEEE 1219- 1998

Este marco se basa en otros anteriores y recoge las definiciones del IEEE 610.12-1990. Define el mantenimiento como la modificación de un producto software después de su entrega al cliente o usuario para corregir defectos, mejorar el rendimiento u otras propiedades deseables, o para adaptarlo a un cambio de entorno.

Establece varias fases dentro del mantenimiento:

- Modificación, identificación, clasificación y priorización de problemas.
- Análisis –Diseño –Implementación –Regresión, y sistema de prueba –Prueba de aceptación – Entrega.

ISO/IEC 14764: 2006 (REVISIÓN DE LA IEEE 1219)

Define el mantenimiento como el conjunto de actividades destinadas a proporcionar soporte económicamente rentable para un determinado producto software.

La norma ISO/IEC 14764:2006 desarrolla con gran detalle la gestión del proceso de mantenimiento descrito en la ISO/IEC 12207, con correcciones. Establece las definiciones de los tipos de mantenimientos y guías de aplicación para la planificación, ejecución, control, revisión, evaluación y finalización del proceso de mantenimiento. Incluye la forma de mantener muchos productos con los mismos recursos.

3. TIPOS DE MANTENIMIENTO SW

AUTOR	TIPOS DE MANTENIMIENTO
MÉTRICA v.3	<ol style="list-style-type: none"> Correctivo: Abarca los cambios precisos para corregir errores del producto software. Adaptativo: Son las modificaciones que afectan a los entornos en los que el sistema opera. P. Ej., configuración del hardware, software de base, comunicaciones, etc. Evolutivo: Son las incorporaciones, modificaciones y eliminaciones necesarias en un producto software para cubrir la expansión o cambio en las necesidades del usuario. Perfectivo: Son las acciones llevadas a cabo para mejorar la calidad interna de los sistemas en cualquiera de sus aspectos: reestructuración del código, definición más clara del sistema y optimización del rendimiento y eficiencia.
BENNET	<ol style="list-style-type: none"> Correctivo: Abarca todas las modificaciones que impliquen la corrección de un error. Adaptativo: Son las modificaciones llevadas a cabo como resultado de cambios producidos en el entorno externo. Perfectivo: Es consecuencia de cambios en los requisitos de los usuarios. Preventivo: Es la realización del mantenimiento anticipándose a problemas futuros.
LIENTZ SWANSON	<ol style="list-style-type: none"> Correctivo: Equivalente al correctivo de Bennet. Adaptativo: Formado por el mantenimiento adaptativo y perfectivo de Bennet. Perfectivo: Equivalente al preventivo de Bennet.
PRESSMAN SOMMERVILLE	<ol style="list-style-type: none"> Correctivo: Abarca todas las modificaciones que impliquen la corrección de errores. Adaptativo: Son las modificaciones debidas a los cambios inherentes a cualquier aspecto de la informática, principalmente, las llevadas a cabo como resultado de cambios producidos en el entorno externo. Perfectivo: Es consecuencia de cambios en los requisitos de los usuarios, e incluye también el mantenimiento que se realiza para proporcionar una base mejor para futuras mejoras.

4. COSTES DE MANTENIMIENTO DEL SOFTWARE

Los costes de mantenimiento representan el porcentaje más alto del coste de todo el ciclo de vida de un sistema y son difíciles de estimar. Como consecuencia de estos grandes costes, es que el coste relativo de reparar un error aumenta considerablemente en las últimas fases del ciclo de vida del software, por lo que se busca detectarlos y repararlos lo antes posible. Estos costes se ven afectados por diversos factores:

FACTORES RELACIONES CON EL SOFTWARE

- Calidad de la documentación
- Cohesión y acoplamiento de los módulos
- Lenguaje de programación
- Configuración del producto
- Pruebas (lo más costoso)

FACTORES AJENOS AL SOFTWARE

- Rotación del personal (Gestión del conocimiento)
- Dependencia del programa al entorno de desarrollo y hardware

Para realizar una estimación de los costes de mantenimiento, Boehm definió una fórmula conocida como TCA – Tráfico de Cambio Anual, y que se define como la fracción de instrucciones de un producto software que sufren modificaciones durante un año, ya sea por adición o alteración. Para calcular el esfuerzo de mantenimiento anual (EMA o ACT) se utilizaría la siguiente expresión:

$$EMA = TCA \times E$$

TCA: Tráfico de cambio anual, por ejemplo, se prevén modificar 1000 líneas de código (LDC) de un total de 10000 líneas, el TCA es de un 10%

E: Esfuerzo de desarrollo del software, por ejemplo, el esfuerzo del desarrollo del software fue de 10 personas/mes.

$$EMA = TCA \times E = 0,10 \times 10 = 1 \text{ persona/mes esfuerzo de mantenimiento}$$

5. TÉCNICAS DE MANTENIMIENTO SW: INGENIERÍA INVERSA Y REINGENIERÍA

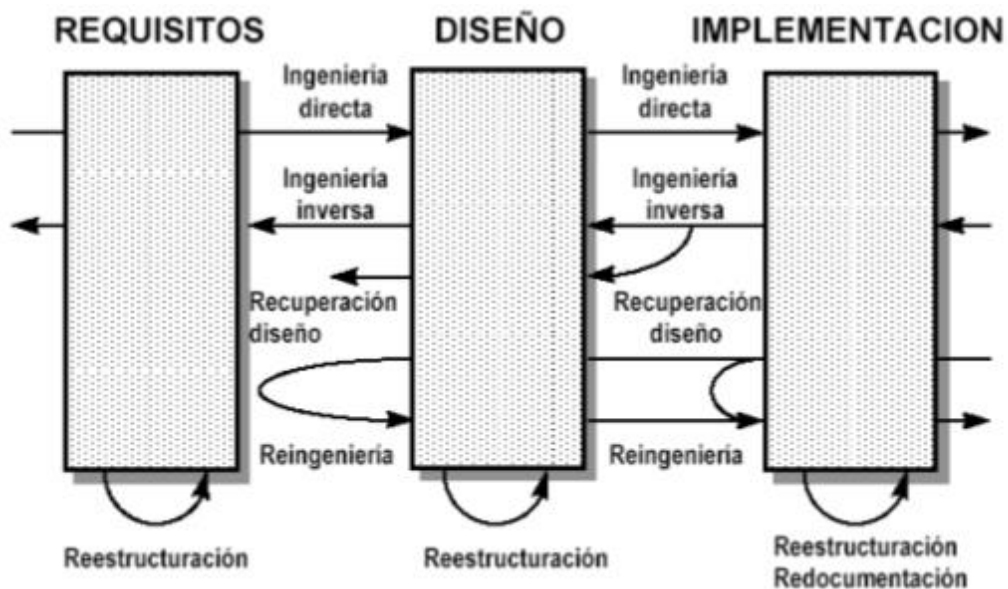
NOTA: El tema asociado a la ingeniería inversa y reingeniería fue eliminado en la convocatoria de 2018, no obstante, debido a su relación con el mantenimiento se incluye una breve reseña de los principales conceptos.

Para cubrir la necesidad de realizar evoluciones más amplias de un sistema es necesario trabajar en los diferentes niveles de abstracción (requisitos, diseño, implementación) con el objetivo de realizar cambios más drásticos, a esta forma de modernización se le denomina “**reingeniería**”. Por otro lado, ante la necesidad de recuperar requisitos o el diseño de un sistema desde el código surge la “**ingeniería inversa**” que se ocupa de estudiar el sistema en orden inverso al establecido en el ciclo de vida habitual.

5.1. MODELO DEL IEEE

- Considera tres niveles de abstracción en un sistema: requisitos, diseño e implementación.
- El modelo define los siguientes conceptos:

- **Inversa (Reverse Engineering):** Es el proceso de analizar un sistema para identificar los componentes y las interrelaciones entre ellos, creando representaciones del sistema en otra forma distinta a la original o bien a un nivel superior de abstracción.
- **Reestructuración (Restructuring):** Es la transformación de una forma de representación del sistema en otra distinta, pero del mismo nivel de abstracción, sin modificar el comportamiento externo del sistema.
- **Ingeniería hacia delante (Forward Engineering):** Es el proceso que va desde un alto nivel de abstracción, que es independiente de la implementación concreta, hasta la propia implementación física del sistema.
- **Reingeniería (Reengineering):** Es el examen y modificación de un sistema para ser reconstruido de una forma nueva y además realizar la implantación derivada de esa nueva forma.



5.2. TIPOS DE REINGENIERIA

- **Modernización de caja blanca:** Reingeniería que requiere de conocimiento a bajos niveles de abstracción (código fuente).
- **Modernización de caja negra o Wrapping:** Solo requiere el conocimiento de las interfaces del sistema (entradas y salidas), ignorando los detalles internos.

6. HERRAMIENTAS DE MANTENIMIENTO

Hacen un uso mucho mayor de técnicas de inteligencia artificial, aplicando una base de conocimientos que sea específica del dominio de la aplicación

- Herramientas de ingeniería inversa a especificaciones. Toman el código fuente como entrada y generan modelos de diseño y análisis estructurado, listas de utilización y otra información con el diseño.
- Herramientas de reestructuración y análisis de código. Analizan la sintaxis del programa, generan un grafo de flujo de control y un programa estructurado.

- Herramientas interactivas de reingeniería de sistema. Se utilizan para modificar sistemas de base de datos.

7. REFERENCIAS

Este tema está relacionado con el **tema 95**, La metodología de planificación y desarrollo de sistemas de información Métrica, en concreto con la etapa del *Mantenimiento del Sistema de Información (MSI)*. Conviene recordar que la estructura de este proceso comprende las siguientes actividades:



ACTIVIDADES	TAREAS
MSI 1	<ul style="list-style-type: none"> • Registro de la petición • Asignación de la petición
MSI 2	<ul style="list-style-type: none"> • Verificación y Estudio de la Petición • Estudio de la Propuesta de Solución
MSI 3	<ul style="list-style-type: none"> • Identificación de Elementos Afectados • Establecimiento del Plan de Acción • Especificación del Plan de Pruebas de Regresión
MSI 4	<ul style="list-style-type: none"> • Seguimiento de los Cambios • Realización de las Pruebas de Regresión • Aprobación y Cierre de la Petición

