

**Universidad Autónoma de Yucatán**

**Facultad de Matemáticas**

**Ingeniería de Software**



**MANTENIMIENTO DE SOFTWARE**

**Resumen Sesión 2**

**Integrantes:**

Jesús Fernando Méndez Avilés

Oscar Daniel Moreno Flores

David Alberto Pat Cituk

Becky Zhu Wu

Braulio Adrián Sarmiento Altonar

**Fecha:** lunes 15 de enero de 2024

<b>Introducción</b>	<b>3</b>
<b>Percepciones sobre problemas del mantenimiento del software</b>	<b>3</b>
<b>Base de Conocimiento sobre Mantenimiento de Software (SWEBOK)</b>	<b>4</b>
Tipos de Mantenimiento	5
Problemas clave del mantenimiento	5
Técnicos (T)	5
Gestión (M)	5
Costos y estimación	5
Medición	5
División general del mantenimiento del Software según el SWEBOK	5
Fundamentos del mantenimiento del Software	5
Proceso de mantenimiento	6
Técnicas para el mantenimiento	6
Herramientas	6
<b>Definición de Mantenimiento de Software</b>	<b>6</b>
Puntos Clave	7

## Introducción

A partir de que un software entre en operación pueden surgir defectos, nuevos requerimientos o cambios en el entorno operativo, por lo que el mantenimiento de software es una fase importante dentro del ciclo de vida del software. Entender las complejidades que tiene un software es crucial para poder garantizar que el software pueda seguir satisfaciendo las necesidades del cliente. Comúnmente este proceso inicia una vez de que se entregue al cliente el producto e inicie la operación.

El costo de mantenimiento suele ser elevado, además de que los usuarios pueden enfrentarse con problemas como la lentitud de atención a sus solicitudes. Mientras que los ingenieros deben de familiarizarse con el código para posteriormente hacer correcciones necesarias, estos se ven obligados a cumplir con los nuevos requerimientos. Por lo tanto, el SWEBOK reconoce que es importante que los ingenieros tengan conocimientos sólidos que los ayude a resolver las peticiones cambiantes y al mismo tiempo identificar los tipos de mantenimientos que existe para trabajar de manera eficiente.

Respecto a lo anterior podemos destacar que se debe de comprender los problemas internos y externos, ya que, el mantenimiento debe de tomar en cuenta una serie de consideraciones para poder realizar el funcionamiento óptimo, así como permitir que un software evolucione adaptandose a nuevas necesidades.

## Percepciones sobre problemas del mantenimiento del software

El mantenimiento del software **empieza cuando el producto es entregado al cliente**, una vez puesto en operación, donde pueden surgir fallas. Éstas son reportadas y se deben tomar acciones para garantizar que el desempeño del software sea el mejor posible. Principalmente, **los problemas que surgen pueden ser de dos tipos** dependiendo de la perspectiva del individuo, de modo que existen problemas de **tipo externos**, desde la perspectiva del usuario; o de **tipo interno**, bajo el criterio de los ingenieros de mantenimiento.

A partir de la distinción anterior, **el principal problema externo** reconocido es **el alto costo** que pagan los usuarios por el mantenimiento el cual suele surgir de una comunicación inadecuada por parte de los gerentes con respecto al tipo de mantenimiento que debe llevarse a cabo en un determinado momento , este tipo de situaciones comúnmente **se debe** a que los gerentes suelen **agrupar tanto las acciones de corrección como las actividades de mejora** dentro de un mismo presupuesto, y por tal motivo se tiene una percepción errónea sobre el verdadero costo. Adicionalmente, otro problema que experimentan los usuarios consiste en una **atención lenta a sus peticiones de corrección** y, a su vez, demandan que todas ellas se resuelvan al mismo tiempo por considerarlas igual de prioritarias, esto ocurre debido a que durante el mantenimiento de un software suelen ocurrir fallos en el sistema aleatoriamente al igual de que las peticiones de los usuarios suceden en momentos inesperados; Este tipo de irregularidades durante el mantenimiento causan que los clientes sientan que el servicio que están recibiendo no es adecuado causando sobre reacciones y demandas de que los problemas y peticiones sean atendidos al mismo tiempo.

También se tiene que tomar en cuenta que el proceso de mantenimiento de software es un proceso laborioso, por lo que una buena cantidad de los costos se deben a los salarios de los programadores, debido a que el hardware ya no representan una porción significativa de los costos, al contratar nuevo personal experto de software se convierte en la prioridad.

Es debido a estos posibles riesgos y problemas que es **sumamente necesario** que las personas encargadas del mantenimiento mantengan **buenas comunicaciones** con sus usuarios y que cuenten con un sistema de manejo de peticiones efectivo que le permita a los operadores reportar fallos en el sistema y que habilite vías de comunicación con los clientes.

Ahora, desde el punto de vista de los individuos que realizan el mantenimiento, los **problemas internos** empiezan por tener que **familiarizarse con código** escrito por personas distintas **con el fin de proveer soluciones** a las peticiones urgentes, todo ello sin interrumpir el servicio que brinda el software. Además, en repetidas ocasiones los ingenieros se enfrentan con software cuya calidad ha sido comprometida debido a **omisiones en las etapas de desarrollo**, de modo que no solo deben atender las peticiones sino que también tienen la responsabilidad de corregir el software por la carga de trabajo acumulada durante el desarrollo. Ante esta realidad, se ha reconocido que los problemas detectados por los ingenieros **se clasifican en alguna de estas categorías**:

**Problemas de gestión:** Entre los problemas más importantes de gestión se encuentran la administración de prioridades cambiantes, la dificultad para medir el desempeño y la ausencia de documentación.

**Problemas técnicos:** Los problemas técnicos más reconocidos son referentes a técnicas de pruebas inadecuadas, ausencia de estándares y metodologías, y código complejo sin estructura.

Entre otros de los problemas que perciben los ingenieros al realizar labores de mantenimiento se encuentra el equipo de trabajo ineficiente para resolver las peticiones cambiantes de los usuarios. Usualmente, esto se acompaña de personal con experiencia insuficiente en las habilidades requeridas para realizar sus tareas, lo que podría vincularse con algún aspecto cultural que resta importancia al mantenimiento debido a que el esfuerzo de tales actividades no se considera significativamente visible para apreciar los beneficios que traería promover el mantenimiento con el fin de mitigar los problemas internos y externos descritos.

## Base de Conocimiento sobre Mantenimiento de Software (SWEBOK)

La mayoría de profesionales tienen un conjunto de conocimientos fundamentales que se adquieren durante una educación formal o experiencia práctica; por otra parte, para el mantenimiento del software el aprendizaje suele ser práctico y poco académico en donde la mayoría de los textos solo lo mencionan superficialmente.

Por esto se creó el **SWEBOK**: Primer consenso internacional sobre el conocimiento fundamental para los ingenieros de software.

## Tipos de Mantenimiento

1. **Adaptativo:** Modificación post-entrega ante cambios ambientales o actualización de sistemas operativos.
2. **Correctivo:** Reparación post-entrega para problemas descubiertos, incluyendo emergencias.
3. **Preventivo:** Post-entrega, detecta y corrige defectos latentes antes de que causen problemas.
4. **Perfectivo:** Modificación post-entrega para mejorar el desempeño del software.

## Problemas clave del mantenimiento

### ***Técnicos (T)***

- Conocimiento limitado
- Pruebas
- Impacto de los cambios
- Capacidad de modificación, (Mantenibilidad)

### ***Gestión (M)***

- Alineamiento con objetivos de la organización
- Mantener empleados
- Manejo de los procesos
- Organización
- Subcontratación

### ***Costos y estimación***

- Impacto de modelos paramétricos, (Basado en datos históricos)
- Experiencia

### ***Medición***

Impacto de la formas en que se especificó el modelo de mediciones del proyecto

## División general del mantenimiento del Software según el SWEBOK

### ***Fundamentos del mantenimiento del Software***

- Necesidad: Se requiere de mantenimiento que asegure que el software continúe satisfaciendo los requerimientos del usuario.
- Costos: Cerca del 80% del presupuesto se consume en actividades que no son correctivas.
- Evolución: Con el paso del tiempo, el software crece y se vuelve más complejo.

### **Proceso de mantenimiento**

- **Procesos de mantenimiento:** Esta sección puede incluir tareas como las presentadas en el proceso definido por el estándar IEEE 14764; la cual se compone de: implementación del proceso, análisis del problema y modificaciones a realizar, implementación de las modificaciones, revisión y aceptación del mantenimiento, migración y retiro del software. Por otro lado también se pueden optar por otros modelos de procesos de mantenimiento como el modelo espiral e incluso una adaptación de una metodología ágil.
- **Actividades de mantenimiento:** A pesar de que las actividades de mantenimiento en su mayoría sean similares a los del desarrollo de software (análisis, diseño, codificación, pruebas y documentación) existen ciertas actividades que pertenecen únicamente al mantenimiento del software, tales como **entender el funcionamiento** del producto software, la **transición** del software del desarrollador al individuo que realizará el mantenimiento, la **atención de solicitudes** de modificación, el **análisis de impactos** generados por cambios, y los **acuerdos o contratos** que definen los objetivos y la calidad deseada al desempeñar las actividades.

### **Técnicas para el mantenimiento**

- **Comprensión del programa:** usualmente implica la lectura de documentación, pero también se apoya de herramientas de organización de código.
- **Reingeniería:** implica la modificación del software heredado para organizar un programa y mejorar su estructura.
- **Ingeniería inversa:** no tiene como objetivo modificar el programa, sino que utiliza herramientas para generar su documentación faltante.
- **Migración:** implica la modificación y documentación del software para funcionar en un nuevo entorno.
- **Retiro:** se realiza un análisis para decidir si el software ha concluido su vida útil, con lo que se crea un plan de retiro y se archivan copias de datos.

### **Herramientas**

- **Slicers:** seleccionan solo partes del programa afectadas por los cambios.
- **Analizadores estáticos:** ofrecen una visión general del contenido del programa.
- **Analizadores dinámicos:** permiten trazar caminos de ejecución.
- **Analizadores de flujo de datos:** trazan todos los posibles flujos de datos del programa.
- **Analizadores de dependencias:** ayudan a comprender las interrelaciones entre componentes.

## **Definición de Mantenimiento de Software**

**La vida del software no termina con su entrega**, el mantenimiento del software toma el relevo, abarcando un amplio espectro de actividades; **como ilustran las siguientes definiciones:** En su esencia, se trata de **modificar el software después de su lanzamiento**, **esto incluye** corregir errores, mejorar funciones y **adaptarse a entornos** cambiantes. **Es la totalidad de los esfuerzos para mantener el software operativo y**

**valioso.** Si bien algunos mantenimientos comienzan durante el desarrollo, **son las actividades posteriores a la entrega** de lo que se trata el mantenimiento; por lo tanto, el mantenimiento no solo es la corrección de errores, **es la evolución continua que mantiene el software** relevante y próspero **a lo largo de su vida útil.**

### Puntos Clave

- La vida del software no termina con su entrega.
- Mantenimiento es modificar el software después del lanzamiento.
- Son todas las actividades luego del lanzamiento.
- Son los esfuerzos para mantener el software operativo, corregir fallas, mejorar su rendimiento y adaptarlo a nuevos entornos.
- Es la evolución continua del software.